Proceedings of the

Nineteenth Annual Symposium of the Pattern Recognition Association of South Africa

27-28 November 2008
Cape Town, South Africa

Proceedings of the

Nineteenth Annual Symposium of the Pattern Recognition Association of South Africa

27-28 November 2008
Cape Town, South Africa

Hosted by:
Department of Electrical Engineering
University of Cape Town
http://www.prasa.org

Edited by: F. Nicolls
ISBN 978-0-7992-2350-7

Member of the International Association of Pattern Recognition (IAPR)

Organisation

PRASA 2008 was organised by the University of Cape Town, Department of Electrical Engineering.

Organising Committee

Fred Nicolls (chair)
Jules-Raymond Tapamo
Febe de Wet

Programme Committee

Fred Nicolls (chair)
Marelie Davel
Febe de Wet
Etienne Barnard
Tshilidzi Marwala
Neil Muller
Jules-Raymond Tapamo

Review process

Full-length papers accepted to PRASA have passed a strict single blind peer review process. The submission and review procedure is as follows:

- Authors submit full camera-ready papers to the conference
- Each paper is evaluated by at least two reviewers
- The programme committee members decide whether to accept or reject the paper based on their comments
- Reviews are returned to the authors. For papers that were conditionally accepted, the authors are permitted to make changes as required
- The authors resubmit the papers for final publication

List of reviewers:

Asheer Bachoo
Etienne Barnard
Niko Brummer
Hanno Coetzer
Marelie Davel
Gerhard de Jager
Alta de Waal
Herman Engelbrecht
Gordon Forbes
Keith Forbes
Kenneth Halland
Ben Herbst
Karin Hunter
Brain Leke
Aby Louw
Vukosi Marivate
Olga Martirosian
Jonas Manamela
Dan Mashao
Shakir Mohamed
Linda Mthembu
Hugh Murrell
Fred Nicolls
Thomas Niesler
Henry Nyongesa
Richard Otukei
Meir Perez
Pieter Scholtz
Helmer Strik
Jules-Raymond Tapamo
Frans van den Bergh
Christiaan van der Walt
Stefan van der Walt
Ewald van Dyk
Charl van Heerden
Gerhard van Huyssteen
Daniel van Niekerk
Anton van Wyk
Barry-Michael van Wyk
Busisiwe Vilakazi
Nicholas Zulu

Table of Contents

Keynote and plenary talks

Full papers

Poster abstracts

Keynote addresses

The difference that South Africa has made to Speaker Recognition
David van Leeuwen

Automatic speaker recognition is an area of speech technology that has received much attention from speech researchers in recent years. Some believe that it is the cleanest of all speech related recognition problems. Although simple in its formulation, the speaker recognition problem appears to have an intricate relation with its application. Text independent Speaker Recognition can be seen as a pattern recognition problem, where features are highly variable sequences related to a single source. The task is to detect whether the source is of known identity.

The engineering of speaker recognition systems depends largely on the availability of example material, in this case speech recordings of thousands of different speakers. Performance is driven by international benchmark evaluations which have been carried out almost every year since 1996 by the National Institute of Standards and Technology in the United States. These competitive evaluations donate new evaluation data to the research community, which guides research directions. In recent years, the challenge of channel and session variability has been the focus of these evaluations.

In this presentation, the typical characteristics of the speaker recognition approach are reviewed, and an overview of the machine learning techniques employed is given.

In recent years, new approaches to the presentation of the speaker recognition output have been developed. This way of presentation makes the technology applicable in a wider range of applications without the need of recalibration. Both in the attempts to overcome channel variability and the application-independent presentation of speaker recognition output researchers from South Africa have played an important role.

The Careful Listener: Speech Processing in Meetings
Thomas Hain

Meetings form an essential part of life for many people and the time spent in face to face meetings is ever increasing while more and more people complain about inefficiency, lack of planning and loss of information. Meetings have to be postponed due to lack of information at the time, essential participants that could not attend or deviation from the real topics at hand. While we are normally very eager to use tools that help to increase productivity in many areas, meetings seem to have been mostly excluded in this quest.

Under the AMI and AMIDA projects observant technologies are developed that aim to assist humans in their tasks rather than replacing their functions. While many of these technologies use several modalities (such as video, speech, handwriting, etc) at the same time, the most important information to date can be derived from speech signals alone. However, most known algorithms have to be altered to cope with the complex acoustic situation and special information not relevant in other domains can be derived.

In this presentation a brief overview of the AMIDA project is given, followed by a discussion of required information for several applications. The information related to speech signals are the speakers identity and location, the timing, the content, the presentation style. Hence speaker diarisation, speaker tracking, and speech recognition are at the core of speech technologies used. The presentation will give an overview of state of the art systems for meetings and their performance. Since processing should be minimally invasive microphone array processing is fundamental to all systems presented. Examples of systems for higher level information extraction using the output of these speech processing algorithms are given.

Heuristics for State Splitting in Hidden Markov Models

Benjamin Murrell, Jules Raymond Tapamo

School of Computer Science
University of KwaZulu-Natal
murrellb@gmail.com, tapamoj@ukzn.ac.za

Abstract

The Baum-Welch algorithm for training Hidden Markov Models requires model topology and initial parameters to be specified, and iteratively improves model parameters. Sometimes prior knowledge of the process being modelled allows such specification, but often such knowledge is unavailable. Experimentation and guessing are resorted to. Techniques for discovering the model topology from the observation data exist, but their use is not commonplace. We propose a state splitting approach to structure discovery, where states are split based on two heuristics: 1) Within-state autocorrelation and 2) transition dependence. Statistical hypothesis testing provides a natural termination criterion, and takes into account the number of observations assigned to each state, splitting states only when the data demands it. With synthetic data, we demonstrate the algorithm's ability to recover the structure of Hidden Markov Models from their observation samples. We also show how it outperforms regular Baum-Welch training in both achieving lower training set AIC and BIC scores, and in a classification task. This superior performance is despite the fact that in both tasks, Baum-Welch training had the advantage of being initialized with the number of states of the HMM that actually generated the data.

1. Introduction

Hidden Markov Models (HMMs) are efficient tools for modeling time varying processes. They are used for classification, prediction, and clustering [10] in fields diverse as speech recognition, bioinformatics, finance and more. Rabiner [1] gives a good introduction to the theory and the details of application. Very briefly, an HMM is a Markov process in which the states cannot be directly observed, but each state has a probability distribution over possible outputs, which can be observed. The parameters of an HMM describe the initial state probability distribution, the state transition probability matrix, and the output probability distribution per state. Following Rabiner [1] in exposition, let λ denote the complete set of model parameters, and O a sequence of observations. There are three canonical problems associated with HMMs. The first is to calculate $P(O|\lambda)$, the probability of a particular observation sequence given a model. This is achieved with the Forward algorithm. The second, given a model λ and an observation sequence O, is to calculate the optimal sequence of hidden states Q. This is efficiently solved with the Viterbi algorithm. The third, when given O and λ, is to adjust the model parameters to maximize $P(O|\lambda)$. The Baum-Welch algorithm is typically used for this, but other faster approximate techniques such as segmental k-means training [1] are also used.

When using HMMs for classification, one typically trains one HMM, λ_w, per category w, by selecting a model architecture and initial parameters, and improving the parameters through Baum-Welch reestimation until a (local) maximum for $P(O_w|\lambda_w)$ is reached, where O_w is the set of observation sequences known to be from category w. To categorize a novel observation sequence O, $P(O|\lambda_w)$ is computed for each λ_w using the Forward algorithm and the novel sequence is assigned to the category of the model with the highest such value.

It should be noted that the initialization of the models is left to art, from selecting the appropriate number of states through to setting the initial parameters of the output distributions. This, combined with the fact that Baum-Welch reestimation gets stuck at local maxima, often leads to sub-optimal performance in domains where there are few clues for model initialization.

The rest of this paper will introduce structure discovery algorithms in general and state splitting algorithms in particular. Focusing on heuristic-based state splitting, we say why previously proposed heuristics don't seem well justified, and propose and motivate two new heuristics. We will evaluate their performance on synthetic data, and describe how they can be combined with an exhaustive approach to improve its efficiency.

2. Structure Discovery

Structure discovery algorithms attempt to circumvent problems of architecture selection and initialization by searching for the appropriate architecture whilst learning the parameters. It is useful to distinguish between top down state splitting approaches where the starting point is a single state, and bottom up state merging approaches where the starting point is a complex model with many states[4]. Some approaches use a combination of merging and splitting. Our approach falls into the top down, state splitting family.

2.1. State Splitting

State splitting involves iteratively creating a new model with one state more than the old model. A state in the old model is duplicated, and the model parameters are reestimated. Two important questions that characterize state splitting approaches are 1) how they decide which state to split, and 2) how they decide when to stop splitting. The answer to the first question suggests a further division of state splitting algorithms. There are algorithms that split every state in turn, and select the split (after parameter reestimation) that produces the best improvement in some model selection criterion (eg. Bayesian Information Criterion [9]). For convenience, we will refer to these algorithms as 'exhaustive', because at each step they try all possible splits. This doesn't mean that they try all possible model architectures. Such techniques usually terminate when no split gives any further improvement in the model selection criterion. Examples

are Ostendorf and Singer [6] and Siddiqi *et al.* [4]. A different class of algorithms use one or more heuristics for deciding which state to split. This circumvents the need to retrain the model once for every state when deciding which state to split. We will focus on such approaches.

2.2. Previous Approaches

Li and Biswas [10] model the state outputs with a single Gaussian. They propose an algorithm that selects the state with the largest variance as a candidate for splitting. They also merge the states with the closest means. They explore both the Bayesian Information Criterion and the Cheeseman-Stutz Approximation to decide when to stop splitting. Splitting the state with the highest variance is a crude heuristic. If the process being modelled really does have some high variance states, these will be needlessly split. It does have the advantage of being fast, as the states variance is one of the parameters in state output distribution.

Takami and Sagayama [2] describe an algorithm that models each state's output with a mixture of two Gaussians. They select as a split candidate the state with the largest divergence between its two Gaussians. This is slightly more sophisticated than the Li and Biswas approach, and will not split states simply because they have large variance in their outputs. There are other objections to this as a splitting heuristic though. Firstly, a state's output might be genuinely bi-modal, and then it would be unnecessarily split. Secondly, this restricts the state output model to two Gaussians, whereas more are often required. In [1] for instance, up to 9 mixture components are used for each state output distribution.

Stenger *et al.* [12] consider a goodness-of-fit test to determine which state to split. After the model parameters have converged through Baum-Welch reestimation, they use the Viterbi algorithm to assign state labels to each observation. They then build up a histogram of the outputs assigned to each state, and use a chi squared goodness-of-fit test to compare that histogram to the corresponding state's output distribution. A bad fit implies a bad approximation, so they consider the states with bad fits as candidates for splitting.[1]

3. Rethinking the reasons for state splitting

The problem with all of the heuristics discussed in the previous section is that they pay no attention to the temporal structure of observations belonging to a state. They depend on assumptions about the distribution of outputs, often advocating state splits in situations that would be more profitably remedied by changing the state observation model.

So when should a state be split? One answer is when we can better approximate the observation sequence through that split. If we do not want to make any assumptions about the output distributions for a given state, then the above heuristics lose their motivation. So what reasons are left for splitting states? We suggest that a good reason for splitting a state is if the observations produced by that state have temporal structure. Within all runs[2] of a particular state, if the output at a particular time is not independent of previous outputs in that run, then a better approximation can often be found with more states. In the case of HMMs with continuous valued outputs, one test for such temporal structure is based on the autocorrelation of the outputs for each state. This is analogous to checking the residuals of a regression for significant autocorrelation, and increasing the model complexity in an attempt to remove it. This introduces our first heuristic for state splitting: Significant autocorrelation.

The autocorrelation of a process is the correlation of that process with a time shifted version of itself (Figure 1). The time shift is typically called the lag, and the autocorrelation at lag k is defined as:

$$R(k) = \frac{E[(O_i - \mu)(O_{i+k} - \mu)]}{\sigma^2} \tag{1}$$

where σ^2 is the variance and μ is the mean. For simplicity, we only consider the autocorrelation at lag 1, although the technique could easily be extended to a range of lags if such a thing proves useful. Restricting to lag 1 the autocorrelation can be rewritten as:

$$R = \frac{E[(O_i - \mu)(O_{i+1} - \mu)]}{\sigma^2} \tag{2}$$

A number of runs of state q occur within our observation sequence O. We want to compute the autocorrelation of observations assigned to state q in such a manner that the end of the j^{th} run of state q does not overlap with the start of the $(j+1)^{th}$ run of state q. Thus we select the subsequence O^q of O where the state of O_i and the state of O_{i+1} are both q and then compute a state q autocorrelation, R^q, as follows:

$$R^q = \frac{E[(O_i^q - \mu_q)(O_{i+1}^q - \mu_q)]}{\sigma_q^2} \tag{3}$$

where μ_q and σ_q^2 are the mean and variance of O^q.

Outputs are assigned to states using the Viterbi algorithm, and all runs of a state are considered. To evaluate whether or not the observations for a particular state exhibit significant autocorrelation, we use a statistical hypothesis test under the null hypothesis that there is no autocorrelation, and reject it if the p-value for the test is below a chosen alpha. We employ the standard test for significant correlation using the Student's T distribution.[3] If there is more than one state that exhibits significant autocorrelation, we split the state with the lowest p-value.

Splitting states based only on autocorrelation proves remarkably successful at recovering the structure of models from synthetically generated data. States are seldom split when they shouldn't be, except when chance dictates that autocorrelation appears in the data when there is none in the generating process, but such is the nature of hypothesis testing. It is possible to construct cases where autocorrelation does not split states that should be. If two states have very similar output distributions, splitting based on autocorrelation cannot separate them. The example in Figure 2 (due to Siddiqi *et al.* [4]) illustrates this. The output distributions for states 2 and 4 are identical, but state 2 only transits to state 3, and state 4 only transits to state 1. No 3 state model can capture the dynamics of such a process, but autocorrelation alone as a splitting criterion never discovers the difference between states 2 and 4.

For this reason, we propose a secondary heuristic, to be employed when no significant autocorrelation remains in the

[1] Confusingly, they cite Montacie *et al.* [8], but Montacie *et al.* were more sensibly using this technique to split mixtures within each state, rather than add states when the histograms don't match the distributions.

[2] A "run" of a sequence is a maximal non-empty segment of the sequence consisting of adjacent equal elements.

[3] The Ljung-Box test for autocorrelation would probably be more appropriate here, and would certainly be essential when testing a range of lags. We chose the simple test as it yields good results. It also has a rank based non-parametric version which can be used if odd distributions are inflating the significance

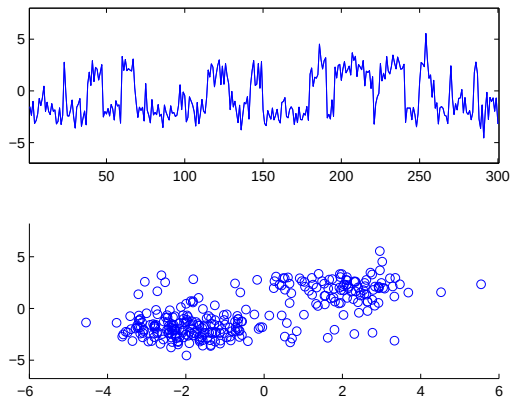


Figure 1: *Visualizing autocorrelation.* Top is an observation sequence O from a 2 state HMM with one observation distribution mean at -2 and the other at 2. Bottom is a scatter plot of points whose x and y values are O_t and O_{t+1} respectively. The positive relationship is evident. If these observations were being credited to a single state, significant autocorrelation would suggest that state be split.

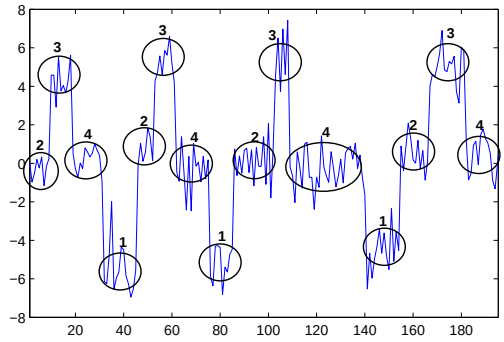


Figure 2: *Overlapping observation densities.* A 4 state HMM, with a single Gaussian output per state. Labeled ellipses denote states that produced corresponding observations. Outputs for states 1 and 3 have means -5 and 5 respectively, but outputs for both states 2 and 4 have mean 0. We can tell states 2 and 4 apart, however, because 2 always transits to itself or 3, and 4 always transits to itself or 1.

outputs of any states. For convenience, we refer to it as the 'transition dependence' heuristic. It attempts to deal with situations such as Figure 2, where the observation distributions for two states cannot be told apart, but their transition vectors are different. This heuristic tests for dependence between where a state transits to, and where it came from. The most likely state path through the observation sequences is estimated using the Viterbi algorithm. Let q denote the state under consideration, and N the total number of states. A $N \times N$ matrix T is constructed where each entry $T(r, c)$ represents the number of times a run of state q transitioned to state c, after being preceded by state r. Note that the q^{th} row and column are empty (as we are considering runs of q), and are removed from the matrix leaving a $(N-1) \times (N-1)$ matrix. The differences in the relative frequencies between these rows is a measure of how much a state's successor depends on its predecessor. If there is only one state producing the observations we have attributed to q, then we should expect these frequencies to differ only by chance. We use a 2 sample chi-square test and compare each row of frequencies to the frequencies summed over all the other rows, under the null hypothesis that they came from the same distribution. This is $N-1$ different tests, and in order to account for this we use Bonferroni correction for multiple tests and scale our alpha accordingly. If any of these $N-1$ tests reject the null hypothesis, then state q is a candidate for splitting. This procedure is repeated for each state, and if more than one states show significant transition dependence, then the one with the lowest p-value is selected for splitting. [4]

Using hypothesis tests means that states are split only when the data demands it, which provides a natural stopping point. The framework is also useful when using many different heuristics, as a single parameter can be set for all of them. It should be noted that, of two states, if the outputs are the same and the transitions are the same, then the states may be considered equivalent, as which of the two states the system is in makes no difference to the present outputs or future states.

[4] Another way of viewing this heuristic is as a test for the state sequence violating the Markov property in a particular way.

Algorithm 1 :Discover

1: Initialize: Create a single state model. Initialize output model using k-means to identify mixture components (see [1]).
2: **repeat**
3: Run Baum-Welch reestimation until convergence. Compute p-values for autocorrelation in each state(p-valAC)
4: **if** min(p-valAC) $<$ alpha **then**
5: Split state argmin(p-valAC)
6: **else**
7: Compute p-values for transition dependence in each state (p-valTD)
8: **if** min(p-valTD) $<$ alpha **then**
9: Split state argmin(p-valTD)
10: **end if**
11: **end if**
12: **until** (min(p-valAC) $\geq$ alpha)) && (min(p-valTD)) $\geq$ alpha)

4. Evaluating Discover

We call our state splitting algorithm 'Discover' (see pseudocode above, and Figure 3 for an example). Discover was evaluated on synthetic data. Randomly generated HMMs were used, with many different state sizes. A single Gaussian output was assumed, and all the parameters were chosen randomly from specified ranges. Each HMM was used to generate sample sequences. Discover was compared to standard Baum-Welch training, where an initial model is selected, and Baum-Welch reestimation is iterated until convergence. Baum-Welch training had the advantage of being initialized with the same number of states as the model that generated the state sequence whilst Discover had no such clues and had to terminate naturally. A single Gaussian output per state was assumed for both Discover, and Baum-Welch training. Baum-Welch is known to be particularly

sensitive to the initial parameters of the state output distributions [1] so we experimented with different state output distribution initialization techniques. A k-means approach made for the strongest opponent, and was used when initializing Baum-Welch training in the following comparisons.

4.1. Performance on model selection criteria

Discover was evaluated using model selection criteria. We chose the unpenalised log-likelihood, the Akaike's Information Criterion with a correction for small samples (AICc), and the Bayesian Information Criterion (BIC). These measures are defined by:

$$AICc = -2\ln L + 2k + \frac{2k(k+1)}{n-k-1}$$

and

$$BIC = -2\ln L + k\ln n$$

where L is the maximized value of the likelihood function for the estimated model, k is the number of free parameters, and n is the number of observations [9]. In both AICc and BIC, the terms after the $-2\ln L$ term are penalties in terms of the number of parameters and the number of observations. BIC penalizes an increase in parameters more heavily than AICc. Note that unlike log-likelihood values, lower AICc and BIC scores correspond to better models.

We compared the model selection criterion scores for Discover against those of Baum-Welch training. For every experiment, the means for all 3 criteria showed Discover outperforming Baum-Welch training. The distribution of the differences between the two methods for each set of sequences generated by a particular model has a curious shape, and as such the full histograms in Figure 4 tells a more detailed story than any comparison of measures of central tendency. Note that values greater than 0 indicate Baum-Welch training being outperformed by Discover, and values less than 0 indicate the converse. The number of states in the generating process did not qualitatively affect the shape of the distributions. The histograms show that very often the values achieved are similar. The tails closer to 0 correspond to smaller, but non-negligible differences (see [9] for a discussion), and the tails spread further out correspond to vast differences and severely deleterious local maxima (relative to the other method). While the peak stays close to 0, the tails are heavier to the positive side, indicating similar performance on many sequence sets, but where differences occur, they more often and more strongly favor Discover.

4.2. Classification tests

A more objective test of performance is a classification task. As above, synthetic data was used from randomly generated HMMs. A 20 'word' vocabulary was created, with each word corresponding to a randomly generated HMM. 30 observation sequences, each with 400 observations were generated by each word, 15 for its training set and 15 for its test set. For each method, one model was trained on the training sequences of each word, with Baum-Welch training once again being initialized with the number of states as the model that generated the sequences. This yields 3 models per word, one from Discover, one from Baum-Welch training, and the original model that generated the sequences. The original model is included as an upper-bound for the classification accuracy, which will change from one iteration of the experiment to the next, as the

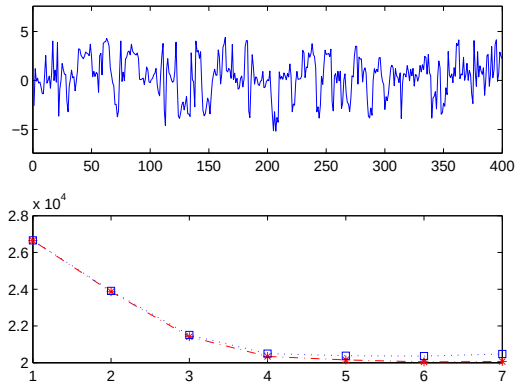


Figure 3: *Discovering a 7 state model.* Top is one of 15 observation sequences from a randomly designed 7 state model, and bottom are the decreasing AICc (red) and BIC (blue) values as Discover splits states to infer the model structure.

similarity between the randomly created models will determine the similarity of the sequences generated and thus the difficulty of the classification task. Basically, some experiment iterations will result in more similar words than others, and using synthetic data generated from known models allows an estimate of the difficulty of the classification task.

Classification accuracies using Discover, Baum-Welch training, and the original models are reported in Tables 1, 2, 3 and 4 for words with 4, 7, 10 and 20 states, with each classification experiment being performed 10 times. Remarkably, the performance of Discover was extremely close to that of the upper bound set by the original model for each iteration. Baum-Welch training performed much worse, even though it was initialized with the correct number of states. The discrepancy between the performance difference in the classification task and that in the model selection criteria evaluation seems large. Examining the confusion matrices is illuminating. Whilst with the original models and Discover, a few errors occur for some words, classification using the Baum-Welch models gets entire words incorrect, with 0s at their corresponding entries along the diagonal of the confusion matrix. We hypothesize that such errors correspond to severe local minima as seen in the model selection criteria evaluation, and it only takes a few to drastically degrade classification accuracy.

5. Future research

The exhaustive state splitting algorithms described earlier have shown to be very successful on a variety of datasets, especially the STACS algorithm due to Siddiqi *et al* [4]. The efficiency of such techniques is increased orders of magnitude by assuming the state paths through the observation sequence remain fixed for all states except the split one, so far fewer parameters need reestimation after each split. Recall that such reestimation has to happen in order to decide which state to split, so such efficiency gains are crucial.

There are still situations in which useful splits will not be discovered by either of our heuristics. For example, if two states have the same output mean and state transition vectors, but the output variance is different, neither autocorrelation nor transition dependence will tell them apart. For this reason, future

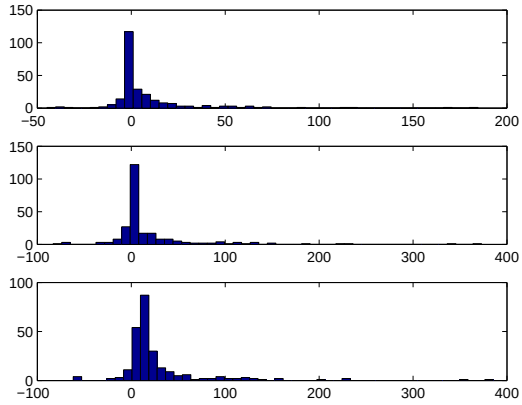


Figure 4: *Pairwise performance differences on model selection criteria.* Top, middle and bottom correspond to log-likelihoods, AICc, and BIC respectively. Each histogram is the distribution of differences between regular Baum-Welch training and Discover, for each set of observations generated from a randomly designed model. The greater density to the right of 0 indicates Discover outperforming regular Baum-Welch training. In the above, 250 models were used, each with 7 states.

Table 1: *Classification accuracies on 4 state HMMs*

iteration	1	2	3	4	5	6	7	8	9	10	mean
Generating Model	0.86	0.93	0.93	0.81	0.82	0.82	0.84	0.87	0.93	0.94	0.87
Structure Discovery	0.86	0.92	0.93	0.81	0.82	0.82	0.82	0.87	0.92	0.94	0.87
Baum-Welch Training	0.70	0.68	0.71	0.68	0.53	0.69	0.75	0.69	0.80	0.66	0.69

Table 2: *Classification accuracies on 7 state HMMs*

iteration	1	2	3	4	5	6	7	8	9	10	mean
Generating Model	0.78	0.92	0.89	0.87	0.95	0.90	0.81	0.91	0.86	0.80	0.87
Structure Discovery	0.78	0.90	0.90	0.87	0.96	0.89	0.82	0.90	0.85	0.79	0.87
Baum-Welch Training	0.71	0.61	0.61	0.66	0.82	0.65	0.65	0.64	0.63	0.70	0.67

Table 3: *Classification accuracies on 10 state HMMs*

iteration	1	2	3	4	5	6	7	8	9	10	mean
Generating Model	0.92	0.88	0.87	0.75	0.94	0.73	0.93	0.95	0.80	0.85	0.86
Structure Discovery	0.91	0.84	0.85	0.75	0.93	0.73	0.92	0.95	0.79	0.84	0.85
Baum-Welch Training	0.75	0.64	0.69	0.66	0.68	0.62	0.75	0.80	0.59	0.65	0.68

Table 4: *Classification accuracies on 20 state HMMs*

iteration	1	2	3	4	5	6	7	8	9	10	mean
Generating Model	0.85	0.84	0.90	0.87	0.85	0.95	0.84	0.76	0.84	0.89	0.86
Structure Discovery	0.77	0.80	0.87	0.82	0.80	0.85	0.79	0.71	0.82	0.88	0.81
Baum-Welch Training	0.58	0.52	0.61	0.59	0.53	0.77	0.59	0.59	0.51	0.71	0.60

work will seek to produce a hybrid structure discovery algorithm, combining the strengths of heuristic exhaustive methods. Firstly, we will adopt the reestimation optimization described above. States will be split by the heuristics described in the present paper, and when no significant autocorrelation or transition dependence remains, we will switch to the exhaustive technique to exploit any undiscovered splits. We don't expect this to discover better models than the exhaustive techniques alone, but we do expect it to be faster. Also, our current implementation deals only with univariate time series data. When generalizing the test for significant autocorrelation to the multivariate case, correction for multiple hypothesis testing will once again be necessary. With very many observations, it might be necessary to adopt a correction less severe than Bonferroni.We also plan to generalize our heuristics to the discrete observation case. This will have to take a form similar to our conditional dependence, but between successive observations rather than states.

6. Conclusion

Two novel heuristics for HMM structure discovery through state splitting, and their superior performance relative to the canonical Baum-Welch technique used in most of the literature was demonstrated. To our knowledge, these heuristics are better motivated than previous ones. We also suggest how they can be incorporated into existing exhaustive techniques, producing a hybrid structure discovery algorithm.

7. Acknowledgements

The authors would like to thank Kevin Murphy for making his "Bayes Net Toolbox for Matlab" freely available, upon which this algorithm was constructed. We would also like to thank Professor Hugh Murrell for formatting assistance and suggestions.

8. References

[1] Rabiner L.A., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol. 77:2, pp. 257-286, 1989.

[2] Takami, J. and Sagayama, S., A successive state splitting algorithm for efficient allophonemodeling, IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP-92), Vol. 1, pp. 573–576, 1992.

[3] Li C., Biswas G., Dale M., and Dale P., Building Models of Ecological Dynamics Using HMM Based Temporal Data Clustering A Preliminary Study, LNCS 2189 F. Hoffmann *et al*. (eds.), Springer-Verlag, pp. 5362, 2001.

[4] Siddiqi S.M., Gordon G.J. and Moore A.W., Fast State Discovery for HMM Model Selection and Learning, Proc. of the Int. Conf. on Artificial Intelligence and Statistics, March 2007.

[5] Stolcke A. and Omohundro S., Hidden Markov Model Induction by Bayesian Model Merging, Advances in Neural Information Processing Systems 5, Giles, Hanson and Cowan, (eds.), Morgan Kaufman, 1993.

[6] Ostendorf M. and Singer H., HMM topology design using maximum likelihood successive state splitting, Computer Speech and Language, vol. 11, pp. 1741, 1997.

[7] Ljung G.M. and Box G.E.P., On a measure of lack of fit in time series models., Biometrica, vol. 65:2, pp. 297-303, 1978.

[8] Montacie C., Caraty M.J. and Barras C., Mixture Splitting Technic and Temporal Control in a HMM-based Recognition System , ICSLP-1996, pp. 977-980, 1996.

[9] Burnham K.P. and Anderson D.R., Multimodel Inference: Understanding AIC and BIC in Model Selection, Sociological Methods and Research, Vol. 33:2, pp. 261-304, 2004.

[10] Li C. and Biswas G., Temporal Pattern Generation Using Hidden Markov Model Based Unsupervised Classification, Hand, Kok and Berthold (eds.): IDA'99, LNCS 1642, Springer-Verlag, pp. 245–256, 1999.

[11] Juang B.H. and Rabiner L.R., The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38:9, pp. 1639–1641, 1990.

[12] Stenger B., Ramesh V., Paragios N., Coetzee F., Buhmann J.M., Topology Free Hidden Markov Models: Application to Background Modeling, Eighth International Conference on Computer Vision (ICCV'01), Vol. 1, pp.294, 2001.

[13] Murphy K., The Bayes Net Toolbox for Matlab, Computing Science and Statistics: Proceedings of Interface, 33, 2001.

Binary Naive Bayesian classifiers for correlated Gaussian features: A theoretical analysis

Ewald van Dyk[1], *Etienne Barnard*[2]

[1,2]School of Electrical, Electronic and Computer Engineering, University of North-West, South Africa
[1,2]Human Language Technologies Research Group, Meraka Institute, Pretoria, South Africa
evdyk@csir.co.za, ebarnard@csir.co.za

Abstract

We investigate the use of Naive Bayesian classifiers for correlated Gaussian feature spaces and derive error estimates for these classifiers. The error analysis is done by developing an exact expression for the error performance of a binary classifier with Gaussian features while using any quadratic decision boundary. Therefore, the analysis is not restricted to Naive Bayesian classifiers alone and can, for instance, be used to calculate the Bayes error performance. We compare the analytical error rate to that obtained when Monte-Carlo simulations are performed for a 2 and 12 dimensional binary classification problem. Finally, we illustrate the robust performances obtained with Naive Bayesian classifiers (as apposed to a maximum likelihood classifier) for high dimensional problems when data sparsity becomes an issue.

1. Introduction

The popularity of Naive Bayesian (NB) classifiers has increased in recent years [1, 2], among others due to exceptional classification performance in high dimensional feature spaces. NB classifiers ignore all correlation between features and are inexpensive to use in high dimensional spaces where it becomes practically infeasible to estimate accurate correlation parameters. An attempt to estimate correlations can often lead to over fitting and decrease the performance (both efficiency and accuracy) of the classifier. Empirical evidence and an intuitive explanation on why NB classifiers perform so well in high dimensional feature spaces (in terms of the bias-variance problem) can be found in [3].

The increase in popularity of NB classifiers has not been matched by a similar growth in theoretical understanding (such as proper error analysis and feature selection). In one of our previous papers [2], we developed analytical tools for estimating error rates and used them as similarity measures for feature selection in discrete environments (all features were assumed to be multinomial).

In this paper, we focus on developing an exact expression for the error rates of binary (two-class) NB classifiers where all features are continuous, correlated multivariate Gaussian distributions.

There have been a few misunderstandings in the past regarding NB classifiers. One good example as pointed out by [3] is the confusion between NB classifiers and linear classifiers in [4]. Consider, for example, a parametric classifier where all features are assumed to be Gaussian. The only way that one can obtain a piecewise linear boundary, is if all classes have identical covariance matrices, which is clearly not the case for general NB classifiers. Therefore, later on in this paper, we discuss the different decision boundaries that can be obtained in a binary NB classification problem with Gaussian features and discuss their intuitive meaning.

In order to calculate the error performance of a binary NB classifier we turn to basic decision theory were we calculate an NB decision boundary that separates two hyperspace partitions Ω_1 and Ω_2. Whenever an observed feature vector falls within region Ω_1 or Ω_2, we classify the pattern to come from class ω_1 or ω_2 respectively. Therefore we can calculate the classification error rate by computing eq. 1[5]

$$\epsilon = p(\omega_1) \int_{\Omega_2} p(\mathbf{x}|\omega_1) d\mathbf{x} + p(\omega_2) \int_{\Omega_1} p(\mathbf{x}|\omega_2) d\mathbf{x}, \quad (1)$$

where ϵ is the classification error rate, $\mathbf{x}$ is the input vector and $p(\omega_1)$ and $p(\omega_2)$ are the prior probabilities for classes ω_1 and ω_2 respectively. Therefore, the very specific challenge addressed in this paper, is to calculate the integral parts in eq. 1, where $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$ are correlated Gaussian distributions of arbitrary dimensionality. Since we are working with NB classifiers, the decision boundary will generally be a quadratic surface.

There exist many upper bounds on the Bayes error rate for Gaussian classification problems. Some popular loose bounds that can be calculated efficiently include the Chernoff bound [6] and the Bhattacharyya bound [7]. Some tighter upper bounds include the equivocation bound [8], Bayesian distance bound [9], sinusoidal bound [10] and exponential bound [11]. Unfortunately, none of these bounds are useful for the analysis of NB classifiers, since they obtain bounds for the Bayes error rate which do not allow us to investigate the effects of the assumption of uncorrelatedness. In order to investigate these effects, we choose to calculate an asymptotically exact error rate. The easiest way to do this, is to do Monte-Carlo simulations where we generate samples from the class distributions and simply count the errors; this is a time-consuming exercise, but does asymptotically converge to the true error rate. Instead, we derive an exact analytical expression similar to work done in [12, 13]. In our derivation, we first transform the integral problems in eq. 1 into a problem of finding the cumulative distribution (cdf) of a linear combination of chi-square variates.

The main contribution of the current paper is that we are able to derive exact analytic expressions for the Naive Bayesian error rate in the general case, whereas previous authors were able to do so only in terms of computationally expensive series expansions [14] or imprecise approximations [13].

The rest of this paper is organized as follows. In section 2, we derive the equations needed to transform the classification problem into one represented as a linear combination of chi-square variates. In section 3, we discuss all possible quadratic

decision boundaries obtained in the context of the work done in section 2 and we show the exact solution to the cdf for most of these boundaries. In section 4, we run simulations to compare NB error rates obtained from both Monte-Carlo simulations and the analytical expressions found.

None of the theory developed in sections 2 and 3 is limited to NB classifiers and applies to quadratic discriminant analysis (QDA) in general. To be more specific, Sections 2 and 3 focus on methods for calculating $\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x}$. It is easy to calculate $\int_{\Omega_1} p(\mathbf{x}|\omega_2)d\mathbf{x}$ by simply reversing the roles of ω_1 and ω_2.

2. Linear combinations of non-central chi-square variates

Let us assume that $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$ are both Gaussian distributions with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2 respectively. Therefore

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma_i}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{i})^T\mathbf{\Sigma_i^{-1}}(\mathbf{x}-\mu_\mathbf{i})), \tag{2}$$

where D is the dimensionality of the problem. Unfortunately, the exact values for μ_i and Σ_i are almost never known and need to be estimated, with say $\hat{\mu}_\mathbf{i}$ and $\hat{\mathbf{\Sigma}}_\mathbf{i}$. For NB classifiers, $\hat{\mathbf{\Sigma}}_\mathbf{i}$ is a diagonal matrix. For simplicity we assume that $\hat{\mu}_\mathbf{1} = \mu_\mathbf{1}$ and $\hat{\mu}_\mathbf{2} = \mu_\mathbf{2}$ – inaccuracy in estimating the sample means is best treated as a separate issue.

We can use these estimates to calculate the decision boundary for a binary classification problem. Eq.3 is the simplest way to describe the decision boundary hyperplane in terms of the estimated parameters.

$$p(\omega_1)p(\mathbf{x}|\hat{\mu}_\mathbf{1}, \hat{\mathbf{\Sigma}}_\mathbf{1}) = p(\omega_2)p(\mathbf{x}|\hat{\mu}_\mathbf{2}, \hat{\mathbf{\Sigma}}_\mathbf{2}) \tag{3}$$

When we take the logarithm on both sides of eq. 3 and use eq. 2, we get the following representation for the decision boundary:

$$\beta_1(\mathbf{x}) = (\mathbf{x}-\hat{\mu}_\mathbf{1})^T\hat{\mathbf{\Sigma}}_\mathbf{1}^{-1}(\mathbf{x}-\hat{\mu}_\mathbf{1}) - (\mathbf{x}-\hat{\mu}_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}(\mathbf{x}-\hat{\mu}_\mathbf{2}) = t_1, \tag{4}$$

where

$t_1 = \log(\frac{|\hat{\mathbf{\Sigma}}_\mathbf{2}|}{|\hat{\mathbf{\Sigma}}_\mathbf{1}|}) + 2\log(\frac{p(\omega_1)}{p(\omega_1)})$.

In the context of eq. 1, it is easy to see that

$$\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} = p(\beta_1(\mathbf{x}) \geq t_1), \tag{5}$$

where $\mathbf{x} \sim N(\mu_\mathbf{1}, \mathbf{\Sigma_1})$.

In the rest of this section, we focus our efforts on transforming eq. 4 into a much more usable form,

$$F(\Phi, \mathbf{m}, t) = p(\sum_{i=1}^{D} \phi_i(y_i - m_i)^2 \leq t), \tag{6}$$

where $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$, $F(\Phi, \mathbf{m}, t)$ is a function that we can relate to the error (see section 3), ϕ_i and m_i are variance and bias constants. We do the transformation in four steps as follows.

2.1. Shift means by μ_1

We define $\mathbf{z} = \mathbf{x} - \mu_\mathbf{1}$ and with a little manipulation (and assuming $\hat{\mu}_\mathbf{1} = \mu_\mathbf{1}$ and $\hat{\mu}_\mathbf{2} = \mu_\mathbf{2}$) we can rewrite eq. 4 as follow.

$$\begin{aligned}
\beta_2(\mathbf{z}) &= \mathbf{z}^T\mathbf{B_1}\mathbf{z} - 2\mathbf{b_1^T}\mathbf{z} = t_2 \\
\mathbf{B_1} &= \hat{\mathbf{\Sigma}}_\mathbf{1}^{-1} - \hat{\mathbf{\Sigma}}_\mathbf{2}^{-1} \\
\mathbf{b_1^T} &= (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1} \\
t_2 &= t_1 + (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}(\mu_\mathbf{1} - \mu_\mathbf{2}) \\
\mathbf{z} &\sim N(\mathbf{0}, \mathbf{\Sigma_1})
\end{aligned} \tag{7}$$

Note that $\mathbf{B_1}$ is in general not a positive-definite matrix, but is symmetric and can be rotated.

2.2. Rotate matrices to Diagonalize Σ_1

Since $\mathbf{z}$ is centered at the origin, we can rotate $\mathbf{\Sigma_1}$ to be diagonal, as long as we rotate the decision boundary too. We define $\mathbf{v} = \mathbf{U_{\omega_1}^T}\mathbf{z}$, where $\mathbf{U}_{\omega_\mathbf{1}}$ is the eigenvector matrix of $\mathbf{\Sigma_1}$ satisfying
$\mathbf{U_{\omega_1}^T}\mathbf{\Sigma_1}\mathbf{U}_{\omega_\mathbf{1}} = \mathbf{\Lambda}_{\omega_\mathbf{1}}$,
$\mathbf{\Lambda}_{\omega_\mathbf{1}} = \mathrm{diag}(\lambda_{\omega_1,1}, ...\lambda_{\omega_1,D})$,
where $\lambda_{\omega_1,1}, ..., \lambda_{\omega_1,D}$ are the eigenvalues of $\mathbf{\Sigma_1}$. From this we can derive eq. 8.

$$\begin{aligned}
\beta_3(\mathbf{v}) &= \mathbf{v}^T\mathbf{B_2}\mathbf{v} - 2\mathbf{b_2^T}\mathbf{v} = t_2 \\
\mathbf{B_2} &= \mathbf{U_{\omega_1}^T}(\hat{\mathbf{\Sigma}}_\mathbf{1}^{-1} - \hat{\mathbf{\Sigma}}_\mathbf{2}^{-1})\mathbf{U}_{\omega_\mathbf{1}} \\
\mathbf{b_2^T} &= (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}\mathbf{U}_{\omega_\mathbf{1}} \\
\mathbf{v} &\sim N(\mathbf{0}, \mathbf{\Lambda}_{\omega_\mathbf{1}})
\end{aligned} \tag{8}$$

2.3. Scale dimensions to normalize all variances in Σ_1

We assume that $\mathbf{\Lambda}_{\omega_\mathbf{1}}$ is positive-definite and therefore none of the eigenvalues are zero. If some of the eigenvalues are zero, the dimensionality of the problem can either be reduced or the classification problem is trivial (if ω_2 has a variance in this dimension or a different mean). (Of course, an NB classifier may not be responsive to this state of affairs, and therefore perform sub-optimally. However, we do not consider this degenerate special case below.)

We define $\mathbf{u} = \mathbf{\Lambda}_{\omega_\mathbf{1}}^{-\mathbf{1/2}}\mathbf{v}$ and derive eq. 9.

$$\begin{aligned}
\beta_4(\mathbf{u}) &= \mathbf{u}^T\mathbf{B}\mathbf{u} - 2\mathbf{b_3^T}\mathbf{v} = t_2 \\
\mathbf{B} &= \mathbf{\Lambda}_{\omega_\mathbf{1}}^{\mathbf{1/2}}\mathbf{U_{\omega_1}^T}(\hat{\mathbf{\Sigma}}_\mathbf{1}^{-1} - \hat{\mathbf{\Sigma}}_\mathbf{2}^{-1})\mathbf{U}_{\omega_\mathbf{1}}\mathbf{\Lambda}_{\omega_\mathbf{1}}^{\mathbf{1/2}} \\
\mathbf{b_3^T} &= (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}\mathbf{U}_{\omega_\mathbf{1}}\mathbf{\Lambda}_{\omega_\mathbf{1}}^{\mathbf{1/2}} \\
\mathbf{u} &\sim N(\mathbf{0}, \mathbf{I})
\end{aligned} \tag{9}$$

2.4. Rotate matrices to diagonalize the quadratic boundary

Now that $\mathbf{u}$ is normally distributed with mean $\mathbf{0}$ and covariance $\mathbf{I}$, it is possible to rotate $\mathbf{B}$ until it is diagonal without inducing any correlation between random variates. Therefore, we define $\mathbf{U_B}$ and $\mathbf{\Lambda_B}$ to be the eigenvector matrix and diagonal eigenvalue matrix of $\mathbf{B}$ respectively.

We finally define $\mathbf{y} = \mathbf{U_B^T}\mathbf{u}$ and derive eq. 10.

$$\begin{aligned}
\beta(\mathbf{y}) &= \mathbf{y}^T\mathbf{\Lambda_B}\mathbf{y} - 2\mathbf{b^T}\mathbf{y} = t_2 \\
\mathbf{b^T} &= (\mu_\mathbf{1} - \mu_\mathbf{2})^T\hat{\mathbf{\Sigma}}_\mathbf{2}^{-1}\mathbf{U}_{\omega_\mathbf{1}}\mathbf{\Lambda}_{\omega_\mathbf{1}}^{\mathbf{1/2}}\mathbf{U_B} \\
\mathbf{y} &\sim N(\mathbf{0}, \mathbf{I})
\end{aligned} \tag{10}$$

It is easy to derive the values for $\mathbf{\Phi}$, $\mathbf{m}$ and t in eq. 6 using eq. 10. These values are given in equation 11.

$$\begin{aligned}\phi_i &= \lambda_{B,i} \quad \forall i \in \{1,...,D\} \\ m_i &= \frac{b_i}{\lambda_{B,i}} \quad \forall i \in \{1,...,D\} \\ t &= t_2 + \sum_{i=1}^{D} \frac{b_i^2}{\lambda_{B,i}}. \end{aligned} \tag{11}$$

It is possible for some of the $\lambda_{B,i}$ values to be zero in which case some of the m_i coefficients become infinite or undefined (this is also the case for t). This happens when some of the random variates only have a linear component in eq. 10 or if the variates make no discriminative difference (in which case b_i is also zero). These cases are discussed in the next section.

3. Decision boundaries and their solutions

In this section we discuss all possible quadratic boundaries derivable from the theory developed in section 2. We also give analytical solutions to the error rate performances associated with each decision boundary (except for paraboloidal decision boundaries discussed later).

3.1. Linear decision boundaries

Linear decision boundaries are the simplest case to solve and occur when $\mathbf{\Lambda_B} = \mathbf{B} = \mathbf{0}$. From eq. 9 it is easy to see that $\hat{\mathbf{\Sigma}}_\mathbf{1} = \hat{\mathbf{\Sigma}}_\mathbf{2}$ for this to be true and it follows that

$$\begin{aligned}\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} &= p(-2\mathbf{b^T y} > t_2) \\ -2\mathbf{b^T y} &\sim N(0, 4\mathbf{b^T b}) \end{aligned} \tag{12}$$

From eq. 12 it is easy to prove that

$$\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} = \frac{1}{2}\,\mathrm{erfc}(\frac{t_2}{\sqrt{8\mathbf{b^T b}}}) \tag{13}$$

3.2. Ellipsoidal decision boundaries

Ellipsoidal decision boundaries occur when either $\mathbf{B}$ or $-\mathbf{B}$ is positive-definite. In other words the eigenvalues $\lambda_{B,1},...,\lambda_{B,D}$ are either all negative or all positive. This is a special case that occurs in NB classifiers when one class consistently has a larger variance than the other class for all dimensions. Since $\mathbf{m}$ (see eq. 11) is defined (none of the eigenvalues are zero), we can attempt to solve eq. 6. Many solutions have been proposed for this problem (see, for example [14]), but the one that we find most efficient is proposed in [13, 15] and is restated here.

Theorem 1. For $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$ and $F(\Phi, \mathbf{m}, t)$ as defined in eq. 6, we have

$$F(\Phi, \mathbf{m}, t) = \sum_{i=0}^{\infty} \alpha_i F_{D+2i}(\frac{t}{p}), \quad \phi_i > 0 \;\; \forall i \in \{1,...,D\},$$

where $F_n(x)$ is defined to be the cdf of a central chi-square distribution with n degrees of freedom, p is any constant satisfying

$$0 < p \leq \phi_i \quad \forall i \in \{1,...,D\},$$

and α_i can be calculated with the recurrence relations

$$\begin{aligned}\alpha_0 &= \exp(-\frac{1}{2}\sum_{j=1}^{D} m_j^2)\sqrt{\prod_{j=1}^{D} p/\phi_j} \\ \alpha_i &= \frac{1}{2i}\sum_{j=0}^{i-1} \alpha_j g_{i-j} \\ g_r &= \sum_{i=1}^{D}(1-p/\phi_i)^r + rp\sum_{i=1}^{D}\frac{m_i^2}{\phi_i}(1-p/\phi_i)^{r-1} \end{aligned}$$

Also, the α coefficients above will always converge and

$$\sum_{i=0}^{\infty} \alpha_i = 1$$

Finally, a bound can be placed on the error from summing only k terms as follows

$$\begin{aligned} 0 &\leq F(\Phi, \mathbf{m}, t) - \sum_{i=0}^{k-1} \alpha_i F_{D+2i}(\frac{t}{p}) \\ &\leq (1 - \sum_{i=1}^{k-1} \alpha_i) F_{D+2k}(t/p) \end{aligned}$$

Proof. The proof can be found in [15].

For optimal convergence in the above series we select $p = \inf\{\phi_1,...,\phi_D\}$, the largest possible value for p.

A useful recurrence relation for calculating $F_n(x)$ is as follows

$$\begin{aligned} F_1(x) &= \mathrm{erf}(\sqrt{\frac{x}{2}}) \\ F_2(x) &= 1 - \exp(-\frac{x}{2}) \\ F_{n+2}(x) &= F_n(x) - \frac{(x/2)^{n/2}e^{-x/2}}{\Gamma(n/2+1)} \end{aligned} \tag{14}$$

We discussed analytical solutions for the case where all α_i's are greater than zero. A symmetric statement can be made for all α_i's less than zero. Therefore, we conclude that

$$\begin{aligned} &\int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x} \\ &= \begin{cases} F(-\Phi, \mathbf{m}, -t) & \sup\{\phi_1,...,\phi_D\} < 0 \\ 1 - F(\Phi, \mathbf{m}, t) & \inf\{\phi_1,...,\phi_D\} > 0 \end{cases} \end{aligned} \tag{15}$$

3.3. Hyperboloidal decision boundaries

Hyperboloidal decision boundaries occur when $\mathbf{B}$ is indefinite and invertible. Therefore, some of the eigenvalues of $\mathbf{B}$ will be positive and others negative, but none of them zero. This is the most frequently occurring case and also the most difficult to solve. Although much research has been done on solving the definite quadratic form (as for the elliptic boundary discussed above), finding an exact analytical expression for the indefinite quadratic form has been unsuccessful (see [12, 13, 14, 16]). The existing solutions all lead to estimates, bounds or unwieldy solutions (and unusable for NB error analysis). In contrast, we propose a solution that is exact and efficient.

Theorem 2. For $\mathbf{y} \sim N(\mathbf{0}, \mathbf{I})$ and $F(\Phi, \mathbf{m}, t)$ as defined in eq. 6, we can rewrite $F(\Phi, \mathbf{m}, t)$ as follows.

$$\begin{aligned} F(\Phi, \mathbf{m}, t) &= p(\sum_{i=1}^{d_1} \phi_i'(y_i - m_i')^2 - \\ &\quad \sum_{j=1}^{d_2} \phi_j^*(y_{d_1+j} - m_j^*)^2 \leq t), \\ \phi_i', \phi_j^* > 0 &\quad \forall i \in \{1, ..., d_1\}, \forall j \in \{1, ..., d_2\}, \end{aligned}$$

where $d_1 + d_2 = D$. From this, we can show that

$$F(\Phi, \mathbf{m}, t) = 1 - \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \alpha_i' \alpha_j^* \Upsilon_{d_1+2i, d_2+2j}(t/p), \quad t \geq 0$$

where we calculate the α_i' and α_j^* coefficients by applying theorem 1 (with common value p) to $F(\Phi', \mathbf{m}', t)$ and $F(\Phi^*, \mathbf{m}^*, t)$ respectively. Note that the α_i' and α_j^* coefficients are independent of t. p can be any arbitrary constant satisfying

$$0 < p \leq \phi_i', \phi_j^* \ \ \forall i \in \{1, ..., d_1\}, \forall j \in \{1, ..., d_2\}$$

$\Upsilon_{k_1,k_2}(z)$ can be calculated using the following recurrence ralations.

$$\begin{aligned} \Upsilon_{1,0}(z) &= \frac{1}{\sqrt{\pi}} \Gamma(1/2, z/2) \\ \Upsilon_{1,1}(z) &= \frac{1}{2}\left[1 - \frac{z}{2}\left(K_0(\frac{z}{2})\mathbf{L}_{-1}(\frac{z}{2}) + K_1(\frac{z}{2})\mathbf{L}_0(\frac{z}{2})\right)\right] \\ \Upsilon_{2,k_2}(z) &= 2^{-k_2/2} e^{-z/2} \\ \Upsilon_{k_1,k_2}(z) &= \Upsilon_{k_1-2,k_2}(z) + D_{k_1,k_2}(z) \\ \Upsilon_{k_1,k_2}(z) &= \Upsilon_{k_1,k_2-2}(z) - D_{k_1,k_2}(z), \end{aligned}$$

where

$$D_{k_1,k_2}(z) = \frac{e^{-z/2}}{2^{(k_1+k_2)/2-1}\Gamma(k_1/2)} \psi(1 - \frac{k_1}{2}, 2 - \frac{k_1+k_2}{2}; z)$$

$\Gamma(a)$ is the gamma function and $\Gamma(a, x)$ is the upper incomplete gamma function. $K_n(x)$ is the modified Bessel function of the second kind and $\mathbf{L}_n(x)$ is the modified Struve function. $\psi(a, b; z)$ is the Tricomi confluent hypergeometric function (also known as the $U(a, b; z)$ function discussed in [17]).

Finally, a bound can be placed on the error from summing only K and L terms.

$$\begin{aligned} 0 &\leq 1 - \sum_{i=0}^{K} \sum_{j=0}^{L} \alpha_i' \alpha_j^* \Upsilon_{d_1+2i, d_2+2j}(t/p) - F(\Phi, \mathbf{m}, t) \\ &\leq (1 - \sum_{i=0}^{K-1} \alpha_i')(\sum_{j=0}^{L-1} \alpha_j^*) \Upsilon_{d_1+2K, d_2+2L}(t/p) \\ &\quad + 1 - \sum_{j=0}^{L-1} \alpha_j^* \end{aligned}$$

Proof. Partial proofs can be found in [12, 13]. Unfortunately, the full proof of this theorem is fairly involved and will be provided in a future paper.

It becomes impractical to calculate $D_{k_1,k_2}(z)$ for large values of k_1 and k_2 and therefore the following recurrence relations become useful

$$\begin{aligned} D_{k_1,k_2}(z) &= \frac{1}{4 - 2k_1}[(4 - k_1 - k_2 - 2z) D_{k_1-2,k_2}(z) \\ &\quad + z D_{k_1-4,k_2}(z)] \\ D_{k_1,k_2}(z) &= \frac{1}{4 - 2k_2}[(4 - k_1 - k_2 + 2z) D_{k_1,k_2-2}(z) \\ &\quad - z D_{k_1,k_2-4}(z)] \\ D_{k_1,k_2}(z) &= \frac{1}{2}(D_{k_1-2,k_2}(z) + D_{k_1,k_2-2}(z)) \end{aligned} \tag{16}$$

Although it is theoretically possible to use only the first two recurrence relations in eq. 16, numerical experiments show that when combined, quantization noise will increase rapidly with each iteration. Therefore we use the first two recurrence relations independently and fill all the remaining gaps with recurrence relation three in eq. 16. Notice that theorem 2 only applies for cases where $t \geq 0$. A symmetric argument can be expressed for cases where $t < 0$. Finally, we conclude that

$$\begin{aligned} &\int_{\Omega_2} p(\mathbf{x}|\omega_1) d\mathbf{x} \\ &= \begin{cases} F(-\Phi, \mathbf{m}, -t) & t < 0 \\ 1 - F(\Phi, \mathbf{m}, t) & t \geq 0 \end{cases} \end{aligned} \tag{17}$$

3.4. Cylindrical decision boundaries

Cylindrical decision boundaries occur when some of the eigenvalues $\lambda_{B,i}$ and their corresponding linear parts b_i are zero. It is fairly easy to see from eq. 10 that these features can simply be dropped and the dimensionality decreased.

3.5. Paraboloidal decision boundaries

Paraboloidal decision boundaries occur when some of the eigenvalues $\lambda_{B,i}$ are zero, but their corresponding linear parts b_i are none-zero. In the context of NB classifiers, this only happens when some of the estimated variances (in a given dimension) are identical for ω_1 and ω_2, but their means differ. Unfortunately, an exact solution for this problem does not yet exist. Therefore, as a temporary solution, we simply add a small disturbance $\delta\lambda_i$ to eq. 10 to get an approximate hyperboloidal or ellipsoidal decision boundary.

4. Results

In this section, we compare the error performance of simple binary classifiers of different dimensionalities for both the Bayes error rate and that obtained using NB classifiers. These error rates will be obtained using two methods: Monte-Carlo simulations and the analytical methods proposed above. Our experimental configurations are similar to those proposed in [13].

4.1. Example 1: A two dimensional classification problem

For this example we will explore the error rates of a two dimensional Gaussian binary classification problem with parameters

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \mathbf{\Sigma_1} = \alpha \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix},$$
$$\mu_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad \mathbf{\Sigma_2} = \alpha \begin{bmatrix} 5 & -2 \\ -2 & 1 \end{bmatrix},$$

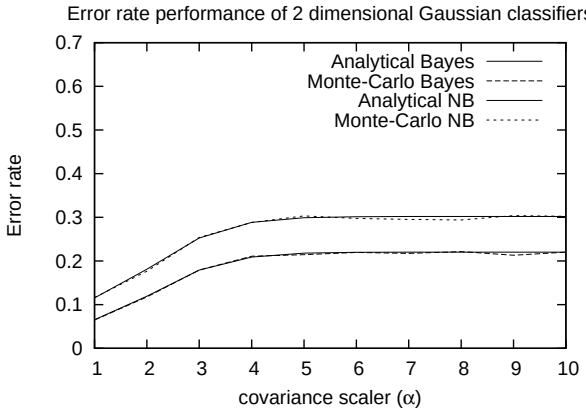


Figure 1: *Naive and Bayes error rates for two dimensional problem in example 1 with increasing class covariances.*

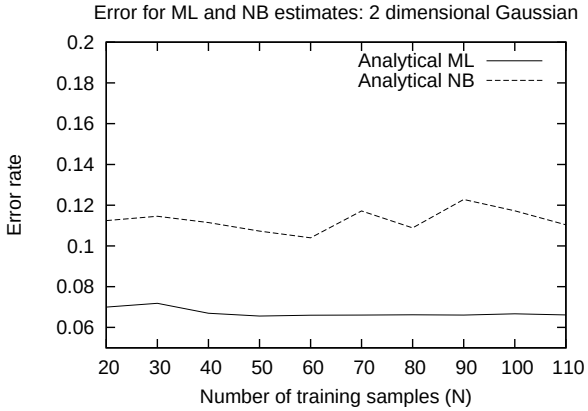


Figure 2: *Naive and maximum likelihood estimate error rates for two dimensional problem in example 2 while increasing the number of training samples.*

where α is a covariance scalar. Figure 1 shows the Bayes and NB (perfect estimate) error rates obtained with the analytical model developed and Monte-Carlo simulations. For this experiment $p(\omega_1) = p(\omega_2) = 0.5$ and 10000 samples in total were generated for the simulations.

Figure 2 shows the analytical results obtained for $\alpha = 1$ where we estimate both the Maximum likelihood (ML) and NB parameters using a varying number of training samples.

It is clear from this experiment that the low dimensional ML classifier provides superior performance to the NB classifier, and that our analytic estimates agree with those obtained by Monte-Carlo simulation.

4.2. Example 2: A 12 dimensional classification problem

Now we explore a high dimensional problem (12 dimensional) to illustrate the power of NB classifiers. For this example we

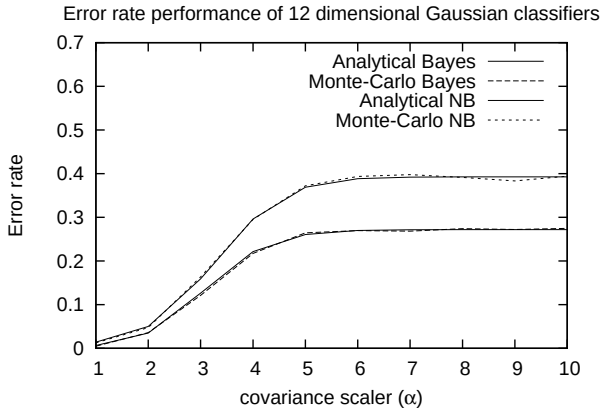


Figure 3: *Naive and Bayes error rates for 12 dimensional problem in example 2 with increasing class covariances.*

define

$$\mu_1 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{\Sigma_1} = \alpha \begin{bmatrix} 5 & -1 & 0 & \dots & 0 \\ -1 & 5 & -1 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & -1 & 5 & -1 \\ 0 & \dots & 0 & -1 & 5 \end{bmatrix},$$

$$\mu_2 = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ -1 \end{bmatrix} \quad \boldsymbol{\Sigma_2} = \alpha \begin{bmatrix} 6 & -2 & 0 & \dots & 0 \\ -2 & 4 & -2 & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & -2 & 6 & -2 \\ 0 & \dots & 0 & -2 & 4 \end{bmatrix},$$

where α is a covariance scalar. Figure 3 shows the Bayes and NB (perfect estimate) error rates obtained with the analytical model developed and Monte-Carlo simulations. For this experiment $p(\omega_1) = p(\omega_2) = 0.5$ and 10000 samples in total were generated for the simulations.

Figure 4 shows the analytical results obtained for $\alpha = 1$ where we estimate both the Maximum likelihood (ML) and NB parameters using a varying number of training samples.

It is clear from figure 4 that for high dimensional problems, NB classifiers perform better when data sparsity is an issue. This is due to the high variance in the ML estimate. NB classifiers are robust for sparse problems and for this specific problem, NB performs relatively well even when more than a hundred training samples are provided.

5. Conclusion

In this paper, we derived analytical solutions for calculating error probabilities in correlated Gaussian feature spaces for arbitrary quadratic decision boundaries. We applied the theory in the context of NB classifiers and showed the validity for both a 2 and 12 dimensional problem by comparing the analytical solutions to those obtained with Monte-Carlo simulations. Both of these case-studies had hyperboloidal Bayes and NB decision

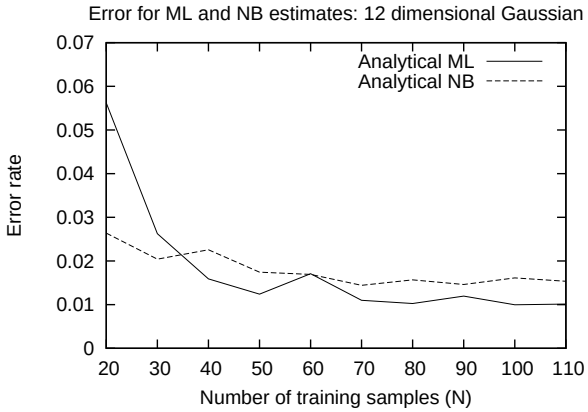


Figure 4: *Naive and maximum likelihood estimate error rates for 12 dimensional problem in example 2 while increasing the number of training samples.*

boundaries, a problem that had not been solved analytically previously.

We also demonstrated the robust behavior of NB classifiers in data sparse and high dimensional environments (see figure 4).

Unfortunately, we still don't have a proper solution for the paraboloidal decision boundaries and we suggested a method for approximating the boundary with a hyperboloidal or ellipsoidal boundary; this method has also been proposed in [13]. It should be noted that this method is not without problems, since the α_i terms in theorem 1 take longer to converge when an exceptionally small ϕ_i value or large m_i value is present. From eq. (11) it is clear that a small value for $\lambda_{B,i}$ will produce a small value for ϕ_i and a large value for m_i.

For future work, we propose to find an exact analytical solution for the error rates obtained when paraboloidal decision boundaries occur. Although these boundaries are themselves degenerate (requiring exactly equal class covariances), the same computational issues arise when the hyperboloidal boundaries are almost paraboloidal (i.e. when the relevant class covariances are close).

6. References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: a Modern Approach*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[2] E. van Dyk and E. Barnard, "Naive bayesian classifiers for multinomial features: a theoretical analysis," in *Proceedings of the 18th Annual Symposium of the Pattern Recognition Association of South Africa*, South Africa, 2007, pp. 75–82.

[3] D.J. Hand and K.Yu, "Idiot bayes ? not so stupid after all?," *International Statistical Review*, vol. 69, no. 3, pp. 385–399, 2001.

[4] P. Domingos and M.Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine-Learning*, vol. 29, pp. 103–130, 1997.

[5] A. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, Ltd., England, second edition, 2002.

[6] H. Chernoff, "A measure for asymptotic efficiency of a hypothesis based on a sum of observations," *The Annels of Mathematical Statistics*, vol. 23, pp. 493–507, 1952.

[7] T. Ito, "Approximate error bounds in pattern recognition," *Machine Intelligence*, vol. 7, pp. 369–372, 1972.

[8] M. Hellman, "Probability of error, equivocation, and chernoff bound," *IEEE Transactions on Information Theory*, vol. 16, pp. 368–372, 1970.

[9] P. Deijver, "On a new class of bounds on bayes risk in multihypthesis pattern recognition," *IEEE Transactions on Computers*, vol. 23, pp. 70–80, 1974.

[10] W. Hashlamoun, P. Varshney, and V. Samarasooriya, "A tight upper bound on the bayesian probability of error," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 220–225, 1994.

[11] H. Avi-Itzhak and T. Diep, "Arbitrarily tight upper and lower bounds on the bayesian probability of error," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 1, pp. 89–91, 1996.

[12] S.J. Press, "Linear combinations of non-central chi-square variates," *The Annels of Mathematical Statistics*, vol. 37, no. 2, pp. 480–487, 1966.

[13] M.H El Ayadi, M.S. Kamel, and F. Karray, "Toward a tight upper bound for the error probability of the binary gaussian classification problem," *Pattern Recognition*, vol. 41, pp. 2120–2132, 2008.

[14] B. Shah, "Distribution of definite and of indefinite quadratic forms from a non-central normal distribution," *The Annels of Mathematical Statistics*, vol. 34, no. 1, pp. 186–190, 1963.

[15] H. Ruben, "Probability content of regions under spherical normal distributions, iv: the distribution of homogeneous and non-homogeneous quadratic functions of normal variables," *The Annels of Mathematical Statistics*, vol. 33, no. 2, pp. 542–570, 1962.

[16] D. Raphaeli, "Distribution of noncentral quadratic forms in complex normal variables," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 1002–1007, 1996.

[17] L.J. Slatêr, *Confluent Hypergeometric Functions*, Cambridge University. Press, London, 1960.

An Introduction to Diffusion Maps

J. de la Porte†*, B. M. Herbst*†*, W. Hereman*⋆*, S. J. van der Walt*†

† Applied Mathematics Division, Department of Mathematical Sciences,
University of Stellenbosch, South Africa
⋆ Colorado School of Mines, United States of America
jolanidlp@googlemail.com, herbst@sun.ac.za,
hereman@mines.edu, stefan@sun.ac.za

Abstract

This paper describes a mathematical technique [1] for dealing with dimensionality reduction. Given data in a high-dimensional space, we show how to find parameters that describe the lower-dimensional structures of which it is comprised. Unlike other popular methods such as Principle Component Analysis and Multi-dimensional Scaling, diffusion maps are non-linear and focus on discovering the underlying manifold (lower-dimensional constrained "surface" upon which the data is embedded). By integrating local similarities at different scales, a global description of the data-set is obtained. In comparisons, it is shown that the technique is robust to noise perturbation and is computationally inexpensive. Illustrative examples and an open implementation are given.

1. Introduction: Dimensionality Reduction

The *curse of dimensionality*, a term which vividly reminds us of the problems associated with the processing of high-dimensional data, is ever-present in today's information-driven society. The dimensionality of a data-set, or the number of variables measured per sample, can easily amount to thousands. Think, for example, of a 100 by 100 pixel image, where each pixel can be seen to represent a variable, leading to a dimensionality of $10,000$. In such a high-dimensional feature space, data points typically occur sparsely, causes numerous problems: some algorithms slow down or fail entirely, function and density estimation become expensive or inaccurate, and global similarity measures break down [4].

The breakdown of common similarity measures hampers the efficient organisation of data, which, in turn, has serious implications in the field of pattern recognition. For example, consider a collection of $n \times m$ images, each encoding a digit between 0 and 9. Furthermore, the images differ in their orientation, as shown in Fig.1. A ***human***, faced with the task of organising such images, would likely first notice the different digits, and thereafter that they are oriented. The observer intuitively attaches greater value to parameters that encode larger variances in the observations, and therefore clusters the data in 10 groups, one for each digit. Inside each of the 10 groups, digits are furthermore arranged according to the angle of rotation. This organisation leads to a simple two-dimensional parametrisation, which significantly reduces the dimensionality of the data-set, whilst preserving all important attributes.

Figure 1: Two images of the same digit at different rotation angles.

On the other hand, a ***computer*** sees each image as a data point in $\mathbb{R}^{nm}$, an nm-dimensional coordinate space. The data points are, by nature, organised according to their position in the coordinate space, where the most common similarity measure is the Euclidean distance.

A small Euclidean distance between vectors almost certainly indicate that they are highly similar. A large distance, on the other hand, provides very little information on the nature of the discrepancy. This Euclidean distance therefore provides a good measure of *local similarity* only. In higher dimensions, distances are often large, given the sparsely populated feature space.

Key to non-linear dimensionality reduction is the realisation that data is often embedded in (lies on) a lower-dimensional structure or manifold, as shown in Fig. 2. It would therefore be possible to characterise the data and the relationship between individual points using fewer dimensions, if we were able to measure distances on the manifold itself instead of in Euclidean space. For example, taking into account its global structure, we could represent the data in our digits data-set using only two variables: digit and rotation.

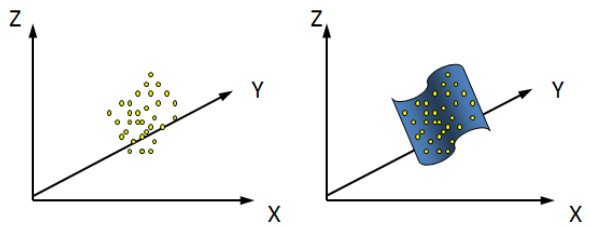


Figure 2: Low dimensional data measured in a high-dimensional space.

The challenge, then, is to determine the lower-dimensional data structure that encapsulates the data, leading to a meaningful parametrisation. Such a representation achieves dimensionality reduction, whilst preserving the important relationships between data points. One realisation of the solution is diffusion maps.

In Section 2, we give an overview of three other well known techniques for dimensionality reduction. Section 3 introduces diffusion maps and explains their functioning. In Section 4, we apply the knowledge gained in a real world scenario. Section 5 compares the performance of diffusion maps against the other techniques discussed in Section 2. Finally, Section 6 demonstrates the organisational ability of diffusion maps in an image processing example.

2. Other Techniques for Dimensionality Reduction

There exist a number of dimensionality reduction techniques. These can be broadly categorised into those that are able to detect non-linear structures, and those that are not. Each method aims to preserve some specific property of interest in the mapping. We focus on three well known techniques: Principle Component Analysis (PCA), multi-dimensional scaling (MDS) and isometric feature map (isomap).

2.1. Principal Component Analysis (PCA)

PCA [3] is a linear dimensionality reduction technique. It aims to find a linear mapping between a high dimensional space (n dimensional) and a subspace (d dimensional with $d < n$) that captures most of the variability in the data. The subspace is specified by d orthogonal vectors: the principal components. The PCA mapping is a projection into that space.

The principal components are the dominant eigenvectors (i.e., the eigenvectors corresponding to the largest eigenvalues) of the covariance matrix of the data.

Principal component analysis is simple to implement, but many real-world data-sets have non-linear characteristics which a PCA mapping fails to encapsulate.

2.2. Multidimensional Scaling (MDS)

MDS [6] aims to embed data in a lower dimensional space in such a way that pair-wise distances between data points, $X_{1..N}$, are preserved. First, a distance matrix D_X is created. Its elements contain the distances between points in the feature space, i.e. $D_X[i,j] = d(x_i, x_j)$. For simplicity sake, we consider only Euclidean distances here.

The goal is to find a lower-dimensional set of feature vectors, $Y_{1..N}$, for which the distance matrix, $D_Y[i,j] = d(y_i, y_j)$, minimises a cost function, $\rho(D_X, D_Y)$. Of the different cost functions available, *strain* is the most popular (MDS using *strain* is called "classical MDS"):

$$\rho_{\text{strain}}(D_X, D_Y) = ||J^T(D_X^2 - D_Y^2)J)||_F^2.$$

Here, J is the centering matrix, so that $J^T X J$ subtracts the vector mean from each component in X. The Frobenius matrix norm, $||X||_F$, is defined as $\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}|x_{ij}|^2}$.

The intuition behind this cost function is that it preserves variation in distances, so that scaling by a constant factor has no influence [2]. Minimising the strain has a convenient solution, given by the dominant eigenvectors of the matrix $-\frac{1}{2}J^T D_X^2 J$.

MDS, when using Euclidean distances, is criticised for weighing large and small distances equally. We mentioned earlier that large Euclidean distances provide little information on the global structure of a data-set, and that only local similarity can be accurately inferred. For this reason, MDS cannot capture non-linear, lower-dimensional structures according to their true parameters of change.

2.3. Isometric Feature Map (Isomap)

Isomap [5] is a non-linear dimensionality reduction technique that builds on MDS. Unlike MDS, it preserves geodesic distance, and not Euclidean distance, between data points. The geodesic represents a straight line in curved space or, in this application, the shortest curve along the geometric structure defined by our data points [2]. Isomap seeks a mapping such that the geodesic distance between data points match the corresponding Euclidean distance in the transformed space. This preserves the true geometric structure of the data.

How do we approximate the geodesic distance between points without knowing the geometric structure of our data? We assume that, in a small neighbourhood (determined by K-nearest neighbours or points within a specified radius), the Euclidean distance is a good approximation for the geodesic distance. For points further apart, the geodesic distance is approximated as the sum of Euclidean distances along the shortest connecting path. There exist a number of graph-based algorithms for calculating this approximation.

Once the geodesic distance has been obtained, MDS is performed as explained above.

A weakness of the isomap algorithm is that the approximation of the geodesic distance is not robust to noise perturbation.

3. Diffusion maps

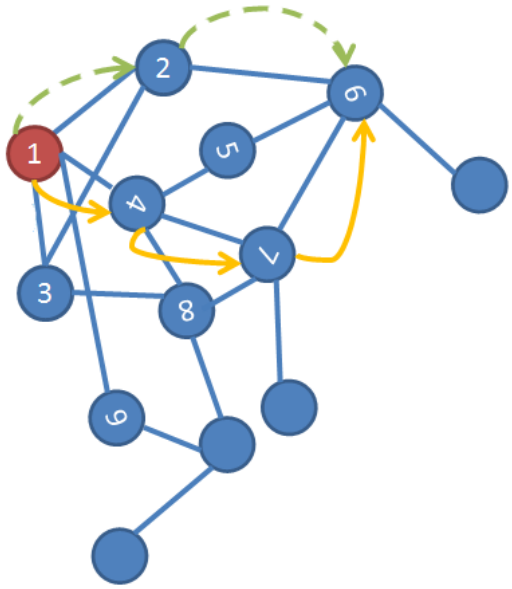


Figure 3: A random walk on a data set. Each "jump" has a probability associated with it. The dashed path between nodes 1 and 6 requires two jumps (i.e., two time units) with the probability along the path being $p(\text{node } 1,\ \text{node } 2)\, p(\text{node } 2,\ \text{node } 6)$.

Diffusion maps are a non-linear technique. It achieves dimensionality reduction by re-organising data according to parameters of its underlying geometry.

The connectivity of the data set, measured using a local similarity measure, is used to create a time-dependent diffusion process. As the diffusion progresses, it integrates local geometry to reveal geometric structures of the data-set at different scales. Defining a time-dependent diffusion metric, we can then measure the similarity between two points at a specific scale (or time), based on the revealed geometry.

A diffusion map embeds data in (transforms data to) a lower-dimensional space, such that the Euclidean distance between points approximates the diffusion distance in the original feature space. The dimension of the diffusion space is determined by the geometric structure underlying the data, and the accuracy by which the diffusion distance is approximated. The rest of this section discusses different aspects of the algorithm in more detail.

3.1. Connectivity

Suppose we take a random walk on our data, jumping between data points in feature space (see Fig. 3). Jumping to a nearby data-point is more likely than jumping to another that is far away. This observation provides a relation between distance in the feature space and probability.

The connectivity between two data points, x and y, is defined as the probability of jumping from x to y in one step of the random walk, and is

$$\text{connectivity}(x, y) = p(x, y). \tag{1}$$

It is useful to express this connectivity in terms of a non-normalised likelihood function, k, known as the diffusion kernel:

$$\text{connectivity}(x, y) \propto k(x, y). \tag{2}$$

The kernel defines a local measure of similarity within a certain neighbourhood. Outside the neighbourhood, the function quickly goes to zero. For example, consider the popular Gaussian kernel,

$$k(x, y) = \exp\left(-\frac{|x - y|^2}{\alpha}\right). \tag{3}$$

The neighbourhood of x can be defined as all those elements y for which $k(x, y) \geq \varepsilon$ with $0 < \varepsilon \ll 1$. This defines the area within which we trust our local similarity measure (e.g. Euclidean distance) to be accurate. By tweaking the kernel scale (α, in this case) we choose the size of the neighbourhood, based on prior knowledge of the structure and density of the data. For intricate, non-linear, lower-dimensional structures, a small neighbourhood is chosen. For sparse data, a larger neighbourhood is more appropriate.

The diffusion kernel satisfies the following properties:

1. k is symmetric: $k(x, y) = k(y, x)$
2. k is positivity preserving: $k(x, y) \geq 0$

We shall see in Section 8 that the first property is required to perform spectral analysis of a distance matrix, $K_{ij} = k(x_i, x_j)$. The latter property is specific to the diffusion kernel and allows it to be interpreted as a scaled probability (which must always be positive), so that

$$\frac{1}{d_X} \sum_{y \epsilon X} k(x, y) = 1. \tag{4}$$

The relation between the kernel function and the connectivity is then

$$\text{connectivity}(x, y) = p(x, y) = \frac{1}{d_X} k(x, y) \tag{5}$$

with $\frac{1}{d_X}$ the normalisation constant.

Define a row-normalised diffusion matrix, P, with entries $P_{ij} = p(X_i, X_j)$. Each entry provides the connectivity between two data points, X_i and X_j, and encapsulates what is known locally. By analogy of a random walk, this matrix provides the probabilities for a single step taken from i to j. By taking powers of the diffusion matrix[1], we can increase the number of steps taken. For example, take a 2×2 diffusion matrix,

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}.$$

Each element, $P_{i,j}$, is the probability of jumping between data points i and j. When P is squared, it becomes

$$P^2 = \begin{bmatrix} p_{11}p_{11} + p_{12}p_{21} & p_{12}p_{22} + p_{11}p_{12} \\ p_{21}p_{12} + p_{22}p_{21} & p_{22}p_{22} + p_{21}p_{12} \end{bmatrix}.$$

Note that $P_{11} = p_{11}p_{11} + p_{12}p_{21}$, which sums two probabilities: that of staying at point 1, and of moving from point 1 to point 2 and back. When making two jumps, these are all the paths from point i to point j. Similarly, P^t_{ij} sum all paths of length t from point i to point j.

3.2. Diffusion Process

As we calculate the probabilities P^t for increasing values of t, we observe the data-set at different scales. This is the diffusion process[2], where we see the local connectivity integrated to provide the global connectivity of a data-set.

With increased values of t (i.e. as the diffusion process "runs forward"), the probability of following a path along the underlying geometric structure of the data set increases. This happens because, along the geometric structure, points are dense and therefore highly connected (the connectivity is a function of the Euclidean distance between two points, as discussed in Section 2). Pathways form along short, high probability jumps. On the other hand, paths that do *not* follow this structure include one or more long, low probability jumps, which lowers the path's overall probability.

In Fig. 4, the red path becomes a viable alternative to the green path as the number of steps increases. Since it consists of short jumps, it has a high probability. The green path keeps the same, low probability, regardless of the value of t .

[1]The diffusion matrix can be interpreted as the transition matrix of an ergodic Markov chain defined on the data [1]

[2]In theory, a random walk is a discrete-time stochastic process, while a diffusion process considered to be a continuous-time stochastic process. However, here we study discrete processes only, and consider random walks and diffusion processes equivalent.

Figure 4: Paths along the true geometric structure of the data set have high probability.

3.3. Diffusion Distance

The previous section showed how a diffusion process reveals the global geometric structure of a data set. Next, we define a diffusion metric based on this structure. The metric measures the similarity of two points in the observation space as the connectivity (probability of "jumping") between them. It is related to the diffusion matrix P, and is given by

$$D_t(X_i, X_j)^2 = \sum_{u \epsilon X} |p_t(X_i, u) - p_t(X_j, u)|^2 \quad (6)$$

$$= \sum_k \left| P^t_{ik} - P^t_{kj} \right|^2. \quad (7)$$

The diffusion distance is small if there are many high probability paths of length t between two points. Unlike isomap's approximation of the geodesic distance, the diffusion metric is robust to noise perturbation, as it sums over all possible paths of length t between points.

As the diffusion process runs forward, revealing the geometric structure of the data, the main contributors to the diffusion distance are paths *along* that structure.

Consider the term $p_t(x, u)$ in the diffusion distance. This is the probability of jumping from x to u (for any u in the data set) in t time units, and sums the probabilities of all possible paths of length t between x and u. As explained in the previous section, this term has large values for paths along the underlying geometric structure of the data. In order for the diffusion distance to remain small, the path probabilities between x, u and u, y must be roughly equal. This happens when x and y are both well connected via u.

The diffusion metric manages to capture the similarity of two points in terms of the true parameters of change in the underlying geometric structure of the specific data set.

3.4. Diffusion Map

Low-dimensional data is often embedded in higher dimensional spaces. The data lies on some geometric structure or manifold, which may be non-linear (see Fig. 2). In the previous section, we found a metric, the diffusion distance, that is capable of approximating distances *along* this structure. Calculating diffusion distances is computationally expensive. It is therefore convenient to map data

points into a Euclidean space according to the diffusion metric. The diffusion distance in data space simply becomes the Euclidean distance in this new *diffusion space*.

A diffusion map, which maps coordinates between data and diffusion space, aims to re-organise data according to the diffusion metric. We exploit it for reducing dimensionality.

The diffusion map preserves a data set's intrinsic geometry, and since the mapping measures distances on a lower-dimensional structure, we expect to find that fewer coordinates are needed to represent data points in the new space. The question becomes which dimensions to neglect, in order to preserve diffusion distances (and therefore geometry) optimally.

With this in mind, we examine the mapping

$$Y_i := \begin{bmatrix} p_t(X_i, X_1) \\ p_t(X_i, X_2) \\ \vdots \\ p_t(X_i, X_N) \end{bmatrix} = P^T_{i*}. \tag{8}$$

For this map, the Euclidean distance between two mapped points, Y_i and Y_j, is

$$\begin{aligned} \|Y_i - Y_j\|^2_E &= \sum_{u \epsilon X} |p_t(X_i, u) - p_t(X_j, u)|^2 \\ &= \sum_k \left|P^t_{ik} - P^t_{kj}\right|^2 = D_t(X_i, Y_j),^2 \end{aligned}$$

which is the diffusion distance between data points X_i and X_j. This provides the re-organisation we sought according to diffusion distance. Note that no dimensionality reduction has been achieved yet, and the dimension of the mapped data is still the sample size, N.

Dimensionality reduction is done by neglecting certain dimensions in the diffusion space. Which dimensions are of less importance? The proof in Section 8 provides the key. Take the normalised diffusion matrix,

$$P = D^{-1}K,$$

where D is the diagonal matrix consisting of the row-sums of K. The diffusion distances in (8) can be expressed in terms of the eigenvectors and -values of P as

$$Y'_i = \begin{bmatrix} \lambda^t_1 \psi_1(i) \\ \lambda^t_2 \psi_2(i) \\ \vdots \\ \lambda^t_n \psi_n(i) \end{bmatrix}, \tag{9}$$

where $\psi_1(i)$ indicates the i-th element of the first eigenvector of P. Again, the Euclidean distance between mapped points $Y'_i$and Y'_j is the diffusion distance. The set of orthogonal left eigenvectors of P form a basis for the diffusion space, and the associated eigenvalues λ_l indicate the importance of each dimension. Dimensionality reduction is achieved by retaining the m dimensions associated with the dominant eigenvectors, which ensures that $\left\|Y'_i - Y'_j\right\|$ approximates the diffusion distance, $D_t(X_i, X_j)$, best. Therefore, the diffusion map that optimally preserves the intrinsic geometry of the data is (9).

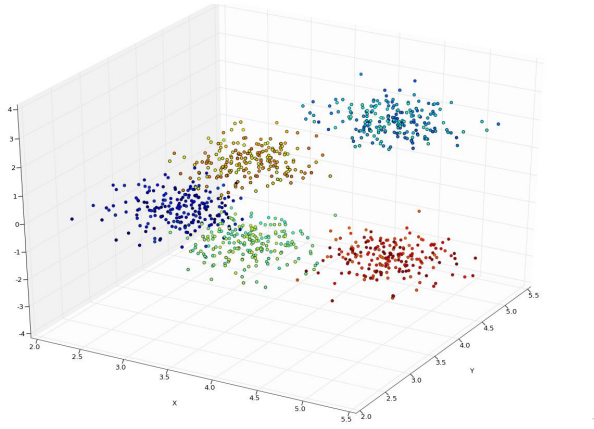


Figure 5: Original Data

4. Diffusion Process Experiment

We implemented a diffusion map algorithm in the Python programming language. The code was adapted for the machine learning framework Elefant (Efficient Learning, Large Scale Inference and Optimisation Toolkit), and will be included as part of the next release. The basic algorithm is outlined in Algorithm 1.

Algorithm 1 Basic Diffusion Mapping Algorithm

INPUT: High dimensional data set X_i, $i = 0 \ldots N-1$.

1. Define a kernel, $k(x, y)$ and create a kernel matrix, K, such that $K_{i,j} = k(X_i, X_j)$.
2. Create the diffusion matrix by normalising the rows of the kernel matrix.
3. Calculate the eigenvectors of the diffusion matrix.
4. Map to the d-dimensional diffusion space at time t, using the d dominant eigenvectors and -values as shown in (9).

OUTPUT: Lower dimensional data set Y_i, $i = 0..N-1$.

The experiment shows how the algorithm integrates local information through a time-dependent diffusion to reveal structures at different time scales. The chosen data-set exhibits different structures on each scale. The data consists of 5 clusters which, on an intermediate scale, has a noisy C-shape with a single parameter: the position along the C-shape. This parameter is encoded in the colours of the clusters. On a global scale the structure is one super-cluster.

As the diffusion time increases, we expect different scales to be revealed. A one dimensional curve characterised by a single parameter should appear. This is the position along the C-shape, the direction of maximum global variation in the data. The ordering of colours along the C-shape should be preserved.

4.1. Discussion

In these results, three different geometric structures are revealed, as shown in figures (5) and (6):

At $t = 1$, the local geometry is visible as five clusters. The diffusion process is still restricted individual clusters, as the probability of jumping to another in one time-step is small. Therefore, even with reduced dimensions, we can clearly distinguish the five clusters in the diffusion space.

At $t = 3$, clusters connect to form a single structure in diffusion space. Paths of length three bring them together. Being better connected, the diffusion distance between clusters decreases.

At this time scale, the one-dimensional parameter of change, the position along the C-shape, is recovered: in the diffusion space the order of colours are preserved along a straight line.

At $t = 10$, a third geometric structure is seen. The five clusters have merged to form a single super-cluster. At this time scale, all points in the observation space are equally well connected. The diffusion distances between points are very small. In the lower dimensional diffusion space it is approximated as zero, which projects the super-cluster to a single point.

This experiment shows how the algorithm uses the connectivity of the data to reveal geometric structures at different scales.

5. Comparison With Other Methods

We investigate the performance of those dimensionality reduction algorithms discussed in Section 2, and compare them to diffusion maps. We use a similar data set as in the previous experiment, the only difference being an increased variance inside clusters. Which of the algorithms detect the position along the C-shape as a parameter of change, i.e. which of the methods best preserve the ordering of the clusters in a one-dimensional feature space? Being linear techniques, we expect PCA and MDS to fail. In theory, isomap should detect the C-shape, although it is known that it struggles with noisy data.

5.1. Discussion

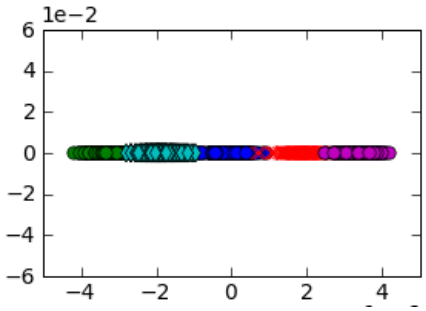


Figure 10: The one-dimensional diffusion space.

5.1.1. PCA

In observation space (Fig. (5)), most variation occurs along the z-axis. The PCA mapping in Fig. 7 preserves this axis as the first principal component. The second principal component runs parallel to $x = y$, and is orthogonal to the z-axis. The variation in the third dimension is orthogonal to these principal components, and preserves the C-shape. Once data is projected onto the two primary axes of variation, the ordering along the non-linear C-shape is lost. As expected, PCA fails to detect a single parameter of change.

5.1.2. MDS

Similar to PCA, MDS preserves the two axes along which the Euclidean distance varies most. Its one-dimensional projection fails to preserve the cluster order, as shown in Fig. 8. Suppose we had fixed the red data points in a one dimensional feature space (see Fig. 11), and wanted to plot the other data points such that Euclidean distances are preserved (this is the premise of MDS). Data points lying on the same radius would then be plotted on top of one another. Due to the non-linear structure of the data, MDS cannot preserve the underlying clusters.

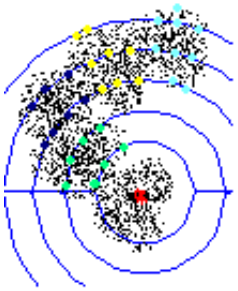


Figure 11: Failure of MDS for non-linear structures

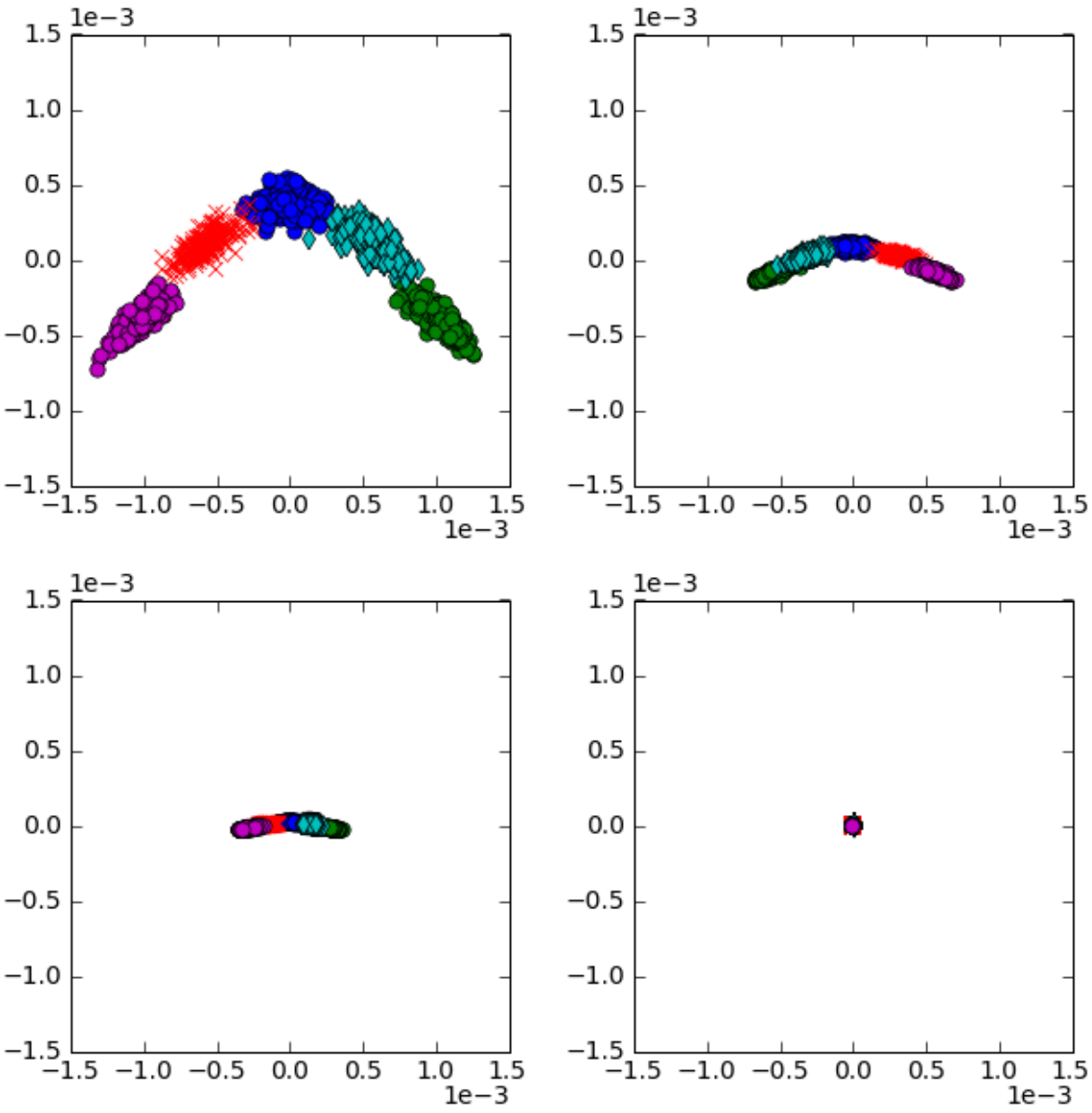


Figure 6: Projection onto diffusion space at times $t = 1$, $t = 2$, $t = 3$ and $t = 10$ (in clock-wise order, starting from the top-left).

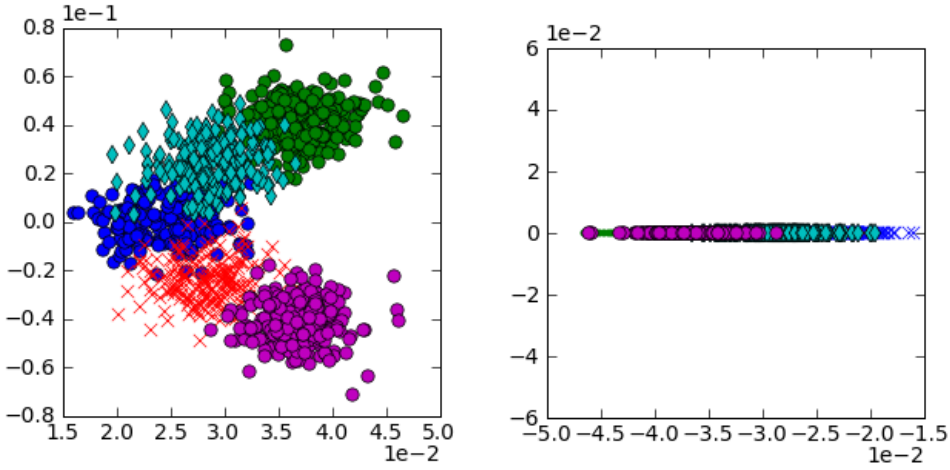


Figure 7: The two- and one-dimensional feature spaces of PCA.

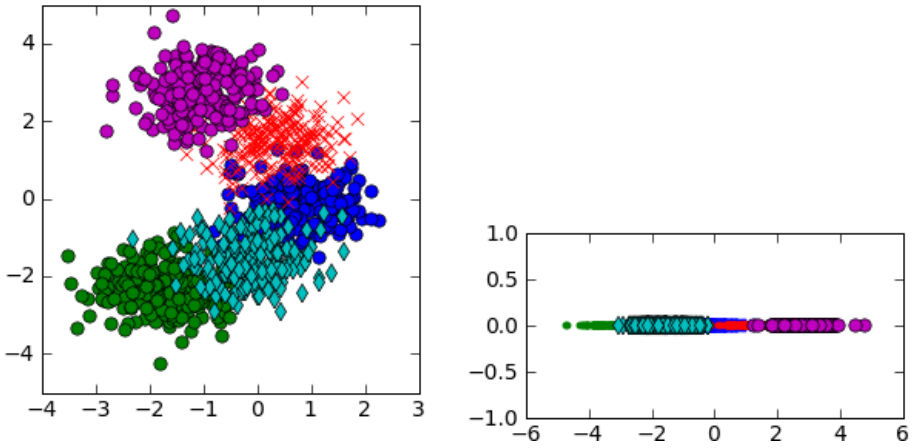


Figure 8: The two- and one-dimensional feature spaces of MDS.

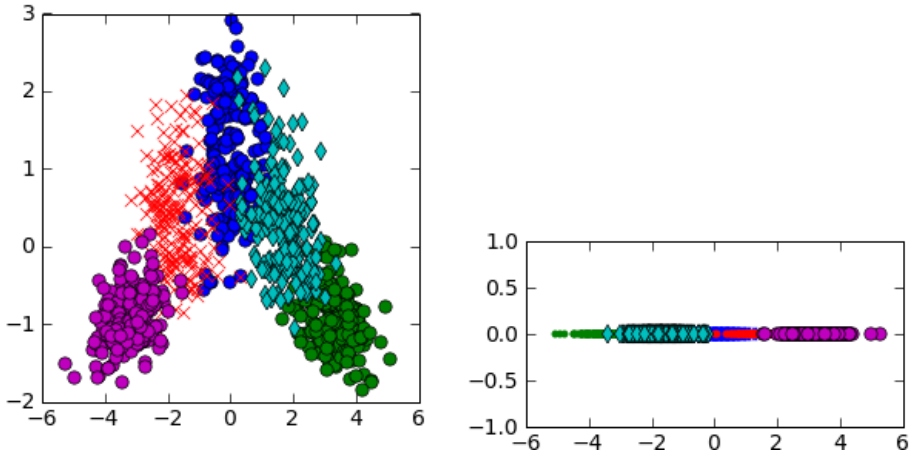


Figure 9: The two- and one-dimensional feature spaces of isomap.

5.1.3. Isomap

When determining geodesic distances, isomap searches a certain neighbourhood. The choice of this neighbourhood is critical: if too large, it allows for "short circuits" whereas, if it is too small, connectivity becomes very sparse. These "short circuits" can skew the geometry of the outcome entirely. For diffusion maps, a similar challenge is faced when choosing kernel parameters.

Isomap is able to recover all the clusters in this experiment, given a very small neighbourhood. It even preserve the clusters in the correct order. Two of the clusters overlap a great deal, due to the algorithm's sensitivity to noise.

5.1.4. Comparison with Diffusion Maps

Fig. 10 shows that a diffusion map is able to preserve the order of clusters in one dimension. As mentioned before, choosing parameter(s) for the diffusion kernel remains difficult. Unlike isomap, however, the result is based on a summation over all data, which lessens sensitivity towards kernel parameters. Diffusion maps are the only one of these techniques that allows geometric analysis at different scales.

6. Demonstration: Organisational Ability

A diffusion map organises data according to its underlying parameters of change. In this experiment, we visualise those parameters. Our data-set consists of randomly rotated versions of a 255×255 image template (see Fig. 12). Each rotated image represents a single, 65025-dimensional data-point. The data is mapped to the diffusion space, and the dimensionality reduced to two. At each 2-dimensional coordinate, the original image is displayed.

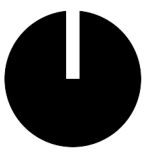


Figure 12: Template: 255 x 255 pixels.

6.1. Discussion

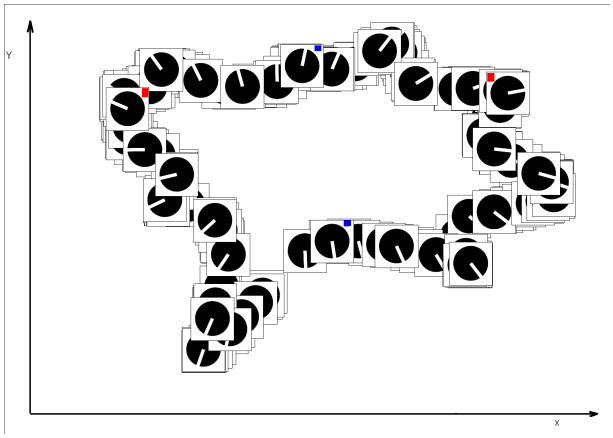


Figure 13: Organisation of images in diffusion space.

In the diffusion space, the images are organised according to their angle of rotation (see Fig. 13). Images with similar rotations lie close to one another.

The dimensionality of the data-set has been reduced from 65025 to only two. As an example, imaging running a K-means clustering algorithm in diffusion space. This will not only be much faster than in data space, but would likely achieve better results, not having to deal with a massive, sparse, high-dimensional data-set.

7. Conclusion

We investigated diffusion maps, a technique for non-linear dimensionality reduction. We showed how it integrates local connectivity to recover parameters of change at different time scales. We compared it to three other techniques, and found that the diffusion mapping is more robust to noise perturbation, and is the only technique that allows geometric analysis at differing scales. We furthermore demonstrated the power of the algorithm by organising images. Future work will revolve around applications in clustering, noise-reduction and feature extraction.

8. Proof

This section discusses the mathematical foundation of diffusion maps. The diffusion distance, given in (7), is very expensive to calculate but, as shown here, it can be written in terms of the eigenvectors and values of the diffusion matrix. These values can be calculated efficiently.

We set up a new *diffusion space*, where the coordinates are scaled components of the eigenvectors of the diffusion matrix. In the diffusion space, Euclidean distances are equivalent to diffusion distances in data space.

Lemma 1: *Suppose K is a symmetric, $n \times n$ kernel matrix such that $K[i,j] = k(i,j)$. A diagonal matrix, D,*

normalises the rows of K to produce a diffusion matrix

$$P = D^{-1}K. \tag{10}$$

Then, the matrix P', defined as

$$P' = D^{1/2}PD^{-1/2}, \tag{11}$$

1. *is symmetric*
2. *has the same eigenvalues as P*
3. *the eigenvectors, x'_k of P' are multiplied by $D^{-1/2}$ and $D^{\frac{1}{2}}$ to get the left ($e^TP = \lambda e^T$) and right eigenvectors ($Pv = \lambda v$) of P respectively.*

Proof: Substitute (10) into (11) to obtain

$$P' = D^{-1/2}KD^{-1/2}. \tag{12}$$

As K is symmetric, P' will also be symmetric.

We can make P the subject of the equation in (11),

$$P = D^{-1/2}P'D^{1/2}. \tag{13}$$

As P' is symmetric, there exists an orthonormal set of eigenvectors of P' such that

$$P' = S\Lambda S^T, \tag{14}$$

where Λ is a diagonal matrix containing the eigenvalues of P' and S is a matrix with the orthonormal eigenvectors of P' as columns.

Substituting (14) into (13),

$$P = D^{-1/2}S\Lambda S^TD^{1/2}.$$

Since S is an orthogonal matrix

$$\begin{aligned} P &= D^{-1/2}S\Lambda S^{-1}D^{1/2} \\ &= (D^{-1/2}S)\Lambda(D^{-1/2}S)^{-1} \qquad (15) \\ &= Q\Lambda Q^{-1}. \qquad (16) \end{aligned}$$

Therefore, the eigenvalues of P' and P are the same. Furthermore, the right eigenvectors of P are the columns of

$$Q = D^{-1/2}S, \tag{17}$$

while the left eigenvectors are the rows of

$$Q^{-1} = S^TD^{\frac{1}{2}}. \tag{18}$$

From (17) we see that the equation for the eigenvectors of P can be given in terms of the eigenvectors x'_k of P'. The right eigenvectors of P are

$$\mathrm{v}_k = D^{-\frac{1}{2}}\mathrm{x}'_k \tag{19}$$

and the left eigenvectors are

$$\mathrm{e}_k = D^{\frac{1}{2}}\mathrm{x}'_k. \tag{20}$$

From (15) we then obtain the eigen decomposition,

$$P = \sum_k \lambda_k \mathrm{v}_k \mathrm{e}_k^T. \tag{21}$$

When we examine this eigen decomposition further, we see something interesting. Eq. 21 expresses each row of the diffusion matrix in terms of a new basis: e_k, the left eigenvectors of the diffusion matrix.

In this new coordinate system in $\mathbb{R}^n$, a row i of P is represented by the point

$$M_i = \begin{bmatrix} \lambda_1\mathrm{v}_1[i] \\ \lambda_2\mathrm{v}_2[i] \\ \vdots \\ \lambda_n\mathrm{v}_n[i] \end{bmatrix},$$

where $\mathrm{v}_n[i]$ is the i-th component of the n-th right eigenvector. However, P is not symmetric, and so the coordinate system will not be orthonormal, i.e.

$$\mathrm{e}_k^T\mathrm{e}_k \neq 1$$

or, equivalently,

$$\mathrm{e}_k^T I\ \mathrm{e}_k \neq 1$$

and

$$\mathrm{e}_l^T I\ \mathrm{e}_k \neq 0 \text{ for } l \neq k.$$

This is a result of the scaling applied to the orthogonal eigenvectors of P' in (19) and (20). This scaling can be counter-acted by using a different metric Q, such that

$$\mathrm{e}_k^T Q \mathrm{e}_k = 1$$

and

$$\mathrm{e}_l^T Q\ \mathrm{e}_k = 0 \text{ for } l \neq k$$

where Q must be a positive definite, symmetric matrix. We choose

$$Q = D^{-1},$$

where D is the diagonal normalisation matrix. It satisfies the two requirements for a metric and leads to

$$\begin{aligned} \mathrm{e}_k^T Q\ \mathrm{e}_k &= \mathrm{e}_k^T(D^{-\frac{1}{2}})^T(D^{-\frac{1}{2}})\ \mathrm{e}_k \\ &= \mathrm{x}_k'^T\mathrm{x}'_k \\ &= 1, \end{aligned}$$

using (20). In the same way we can show that

$$\mathrm{e}_l^T D^{-1}\mathrm{e}_k = 0 \text{ for } l \neq k.$$

Therefore the left eigenvectors of the diffusion matrix form an orthonormal coordinate system of $\mathbb{R}^n$, given that the metric is D^{-1}. We define $\mathbb{R}^n$ with metric D^{-1} as the

diffusion space, denoted by $l_2(\mathbb{R}^n, D^{-1})$. For example, the Euclidean distance between two vectors, a and a', is normally

$$d(a,a')_{l_2} = d(a,a')_{l_2(\mathbb{R}^n,I)} = (a-a')^T(a-a')$$

in l_2. In $l_2(\mathbb{R}^n, D^{-1})$, it becomes

$$d(a,a')_{l_2(\mathbb{R}^n,D^{-1})} = (a-a')^T D^{-1}(a-a').$$

Lemma 2: *If we choose our diffusion coordinates as in (9), then the diffusion distance between the points in the original space is equal to the Euclidean distance in the diffusion space.*

Proof*:* We are required to prove that

$$\begin{aligned} D_t(x_i,x_j)^2 &= \|p_t(x_i,\cdot) - p_t(x_j,\cdot)\|^2_{\mathrm{l}^2(\mathbb{R}^n,I)} && (22)\\ &= \|M_i - M_j\|^2_{\mathrm{l}^2(\mathbb{R}^n,D^{-1})} && (23)\\ &= \sum_k \lambda_k^{2t}(\mathrm{v}_k[i]-\mathrm{v}_k[j])^2. && (24)\end{aligned}$$

Here, $p_t(x_i,x_j) = P_{ij}$ are the probabilities which form the components of the diffusion matrix. For simplicity we assume t = 1. Then

$$\begin{aligned} D(x_i,x_j)^2 &= \|p(x_i,\cdot) - p(x_j,\cdot)\|^2_{\mathrm{l}^2(\mathbb{R}^n,I)}\\ &= \|P[i,\cdot] - P[j,\cdot]\|^2_{\mathrm{l}^2(\mathbb{R}^n,I)}.\end{aligned}$$

According to the eigen decomposition in (21), this equals

$$\begin{aligned} &= \left|\sum_k \lambda_k \mathrm{v}_k[i]\mathrm{e}_k^T - \sum_{k\geq 0}\lambda_k \mathrm{v}_k[j]\mathrm{e}_k^T\right|^2\\ &= \left|\sum_k \lambda_k \mathrm{e}_k^T\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)\right|^2\\ &= \left|\sum_k \lambda_k \mathrm{x}_k'^T D^{\frac{1}{2}}\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)\right|^2\\ &= \left|\sum_k \lambda_k \mathrm{x}_k'^T\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)D^{\frac{1}{2}}\right|^2\end{aligned}$$

In $l_2(\mathbb{R}^n, D^{-1})$, the diffusion space, this distance becomes

$$\begin{aligned} &\left(\sum_k \lambda_k \mathrm{x}_k'^T\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)D^{\frac{1}{2}}\right) D^{-1}\left(\sum_m \lambda_m \mathrm{x}_m'^T\left(\mathrm{v}_m[i]-\mathrm{v}_m[j]\right)D^{\frac{1}{2}}\right)^T\\ =\ &\left(\sum_k \lambda_k \mathrm{x}_k'^T\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)D^{\frac{1}{2}}\right) D^{-1}\left(D^{\frac{1}{2}}\sum_m \lambda_m \mathrm{x}_m'\left(\mathrm{v}_m[i]-\mathrm{v}_m[j]\right)\right)\\ =\ &\sum_k \lambda_k \mathrm{x}_k'^T\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)\sum_m \lambda_m \mathrm{x}_m'\left(\mathrm{v}_m[i]-\mathrm{v}_m[j]\right)\end{aligned}$$

Since $\{x_k'\}$ is an orthonormal set,

$$\mathrm{x}_m'^T\mathrm{x}_k' = 0 \qquad \text{for} \quad m \neq k.$$

Therefore

$$D(x_i,x_j)^2 = \sum_k \lambda_k^2\left(\mathrm{v}_k[i]-\mathrm{v}_k[j]\right)^2. \qquad (25)$$

We've therefore shown that the diffusion distance, $D_t(x_i,x_j)^2$, is simply the Euclidean distance between mapped points, M_i and M_j, in diffusion space.

9. References

[1] R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[2] A. Ihler. Nonlinear Manifold Learning (MIT 6.454 Summary), 2003.

[3] I.T. Jolliffe. *Principal component analysis*. Springer-Verlag New York, 1986.

[4] S.S. Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.

[5] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[6] W.S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

Ensemble Feature Selection for Hyperspectral Imagery

Gidudu, A., Abe, B. and Marwala, T.

School of Electrical and Information Engineering
University of the Witwatersrand, 2050, South Africa
Anthony.Gidudu@wits.ac.za, Bolanle.Abe@students.wits.ac.za, Tshilidzi.Marwala@wits.ac.za

Abstract

The process of relating pixels in a satellite image to known land cover classes is referred to as image classification. As demonstrated in this paper, ensemble feature selection offers a unique approach to land cover mapping, especially for very high dimensional hyperspectral data. Ensemble classification is premised on ensuring diversity among the base classifiers and adopting appropriate means of combining their outputs into a single classification result. This paper explores ensemble feature selection as a means of ensuring diversity for land cover mapping of hyperpectral data. Results show that random selection of features (bands) yielded the best results as compared to building base classifiers depending on search algorithms or as was used in this case, sequentially arranging the features into base classifiers. Of the combination techniques, the single best technique yielded better results than majority vote, however in most cases the difference between the results was not significant.

1. Introduction

The extraction of land cover information from satellite imagery has been one of the major beneficiaries of developments in machine learning. Techniques such as support vector machines, neural networks, fuzzy logic, genetic algorithms etc. which have taken root in remote sensing studies owe their origin to advancements in computational intelligence (and by extension machine learning). Ensemble classification is another such technique that has taken root in machine learning and is making inroads in image classification for land cover mapping. In literature, ensemble classification goes by several names such as multiple classifier systems, committee of classifiers, mixture of experts and ensemble based systems (Polikar [1]). The essence of ensemble classification is to have a final classification result 'in consultation' with a group of classifiers. It is akin to getting a second, third (or more) opinion about a financial, medical or social decision one may have to make (Polikar [1]). For ensemble systems to perform effectively, it has been shown that the constituent base classifiers need to have diversity in their predictions (Opitz [2]; Tsymbal *et al.* [3]; Foody *et al.* [4]). One way of ensuring diversity in ensemble classifiers is in the use of different feature subsets or so called ensemble feature selection (Tsymbal *et al.* [3]). In land cover mapping this would entail having an ensemble of different spectral band combinations and having the final classification result based on a pre-stipulated 'consensus' (e.g. plurality vote) of the different band combinations (Chen *et al.* [5]). Previous work on the application of ensemble classification to land cover mapping has focused mostly on ensuring diversity by using different classifiers. In this paper, the application of ensemble feature classification is explored on hyperspectral data. Hyperspectral data by its nature consists of a very high dimensional feature space (e.g. 200 bands/features) and presents an ideal situation to explore the use of ensemble feature classification for land cover mapping. This paper is organized as follows: section 2.0 gives an overview of ensemble classification, section 3 briefly discusses support vector machines which are the classifiers of choice in this research, section 4 presents the developed methodology in executing

this work, while sections 5 and 6 highlight the results, discussions and conclusions thereof.

2. Ensemble Classification

As alluded to before, ensemble classification is a multiple classifier system in which the aim is to combine the outputs of several classifiers in order to derive an accurate classification (Foody *et al.* [4]). There is a general consensus that ensemble classifiers yield favorable results compared to those of single systems for a broad range of applications (Bruzzone *et al.* [6]). However, Foody *et al.* [4] and Liu *et al.* [7] argue that whereas the adoption of an ensemble based approach may typically yield a classification with an accuracy that is higher than that of the least accurate classifier used in the ensemble, it may not necessarily be better than each of the base (constituent) classifiers.

Polikar [1] raises a number of reasons justifying the need for an ensemble approach. One reason is that, combining the output of several classifiers by say averaging may reduce the risk of an unfortunate selection of a poor performing classifier (Polikar [1]). Polikar [1] further states that whereas the final output of the ensemble may not beat the performance of the best classifier in the ensemble, it certainly reduces the risk of making a particularly poor choice. In his second justification, Polikar [1] proposes an ensemble approach in the face of large volumes of data, especially in cases where the amount of data may be too large to be handled by a single classifier. Partitioning the data into smaller subsets and training different classifiers with different portions of data and combining the outputs using an intelligent combination rule could potentially prove to an efficient approach (Polikar [1]).

The functionality of ensemble systems involves generating the individual base classifiers and devising a means of combining the outputs of these base classifiers. One way of ensuring improved performance of the ensemble system is to ensure that the individual classifiers make errors differently (Polikar [1]). The premise is that if each classifier makes errors differently, i.e. that there is diversity among the base classifiers, then a strategic combination of these classifiers can reduce the total error (Polikar [1]). Diversity in ensemble systems can be achieved through using different: training datasets, classifiers, features or training parameters (Polikar [1]). Chen *et al.* [5] categorizes ensemble classification into those based on several different learning algorithms and those based on just one. The former involves using several classifiers on the same dataset. The drawback of this ensemble system is to have to handle different classifiers which increases the complexity of the processing (Chen *et al.* [5]). In the second categorization, only one classifier is used and the ensemble is created by changing the training set. Two popular examples of this include bagging or bootstrap aggregating (Breiman [8]) and Adaboost or reweighting boosting (Freund *et al.* [9]).

Under the second categorization is an effective approach for generating an ensemble system by the use of different feature subsets or the so called ensemble feature selection (Opitz [2]). Varying the feature subsets used to generate the base classifier potentially promotes diversity since the classifiers tend to err in different subspaces of the instance space (Oza *et al.* [10]; Tsymbal *et al.* [3]). Some of the techniques used to identify features to be used in ensemble systems include genetic algorithms (Opitz [2]), exhaustive search methods and random selection of feature subsets (Ho [11]).

Equally important to the generation of an ensemble is how the base classifier outputs are to be combined. There are two basic approaches in literature which have been suggested as means of integrating ensemble output (Tsymbal *et al.* [3]): a combination approach and secondly a selection approach. A range of methods are available for the combination of information from multiple classifiers (Giancinto *et al.* [12]; Valentini *et al.* [13]; Huang *et al.* [14]). Some of the methods include majority voting (Chen *et al.* [5]), weighted majority voting (Polikar [1]) or more sophisticated methods like consensus theory (Benediksson *et al.* [15] and stacking (Džeroski *et al.* [16]). A number of selection approaches have also been proposed to solve the integration of ensemble data (Tsymbal *et al.* [3]). One of the most popular and simplest selection techniques is Cross Validation Majority (CVM) also called single best. In CVM, the cross

validation accuracy for each base classifier is estimated using the training set and then the classifier with the highest accuracy is selected Tsymbal *et al.* [3].

3. Support Vector Machines

Support Vector Machines (SVMs) are a supervised classification technique having their roots in Statistical Learning Theory. Given a training dataset, the decision boundary between the individual classes is a linear discriminant placed midway between the classes and is expressed as (Foody *et al.* [4]): $f(x) = \text{sign}(\sum_{i=1}^{r} \propto_i y_i k(x, x_i) + b)$ where y_i defines the classes, α_i, i = 1,2,…,r are the Lagrange multipliers, b is bias and $k(x,x_i)$ is a kernel function. In many practical cases, a linear discriminant between the training data classes is not feasible and in order to cater for this nature of data, it is nonlinearly projected into a higher dimension space using the kernel $k(x,x_i)$. Placing a linear discriminant in this high dimension feature space is equivalent to placing a nonlinear discriminant in the previous space. Examples of kernels that can be used for this purpose include: polynomial, sigmoid and Gaussian functions. For each kernel, corresponding parameters are obtained through cross validation before the eventual classification. A more detailed treatise of SVMs can be found in references such as Vapnik [17], Christianini *et al.* [18] and Campbell [19].

4. Methodology

4.1 Data Description

The hyperspectral data used in this paper was sourced from the AVIRIS sensor [20] and represents Indiana's Indian Pines in the United States of America. It is a freely accessible online dataset which comes with accompaigning ground truth data. Of the 224 bands, 4 were discarded because they contained zeros and of the remaining bands only 180 were used in this research. The rest of the bands were left out because of being affected by atmospheric distortion (Bazi *et al.* [21]). The classes of interest included; alfalfa, corn-notill, corn-minimum till, corn, grass/pasture, grass/trees, grass/pasture-mowed, hay-windrowed, oats, soybeans-notill, soybeans-minimum till, soybean-clean, wheat, woods, building-grass-tree-drives, stone-steel towers. These classes were selected in reference to the ground truth data.

4.2 Research Design

Based on Chen *et al.* [5]'s categorization, this paper focused on the ensemble approach dependent on one learning algorithm (In this case Gaussian SVMs), with diversity being enforced through using different feature (band) combinations. Two ensemble feature selection techniques were used namely exhaustive search and random selection of feature subsets. The evaluation function for the exhaustive search was the Bhattacharyya Distance separability index (Bhattacharyya [22]). The results of the base classifiers in each ensemble were combined using two methods; majority voting and an adaptation of Cross Validation majority (CVM) also called single best. In CVM, cross validation data is used as a basis for selecting the best out of the whole ensemble. In this paper, this was modified to consider the final classification results of each base classifier instead. For comparison, another ensemble was derived by sequentially grouping subsequent bands into 10 base classifiers. i.e. bands 1-18 made up the first base classifier, bands 19 – 36 the second base classifier etc, making a total of 10 base classifiers for all the 180 bands.

For each base classifier and corresponding ensemble, classification was carried out in MATLAB with the results being imported into IDRISI Andes for data integration and generation of a land cover map. Classification accuracies were then calculated for each derived land cover map, by making comparisons between the predicted output from the base and ensemble classifiers and the ground truth data. These results were then used as the basis upon which to evaluate ensemble feature classification and its corresponding effect on land cover mapping.

5. Results and Discussions

The Table 1 shows the results of the different ensembles considered. The classification accuracy is given in terms of the Kappa coefficient of agreement (Cohen [23]), which is a measure of

how well the derived map compares with ground truth data. It ranges from 0 to 1 with 0 implying no agreement between predicted land cover and ground truth, and 1 indicating complete agreement.

All the ensembles had ten base classifiers, the figure ten having been arbitrarily chosen. The base classifiers in Ensembles 1, 2, 3 and 4 consisted of 10, 14, 18 and 18 features (bands) respectively, each with different band combinations (feature configurations). Ensembles 1 and 2 were derived from an exhaustive search strategy, with the ten best base classifiers being selected based on their separability indices. Ensemble 3 was constituted by sequentially arranging the 180 bands into ten base classifiers, each with 18 features. On the other hand, all the features constituting the base classifiers in Ensemble 4 were randomly selected.

Table 1: A summary of the classification accuracy of the various ensembles considered

Ens.	**1**	**2**	**3**	**4**
BC 1	0.6209	0.6214	0.4591	0.6176
BC 2	0.6134	0.6323	0.4737	0.6531
BC 3	0.6112	0.6264	0.3383	0.6084
BC 4	0.6232	0.6418	0.3937	0.6605
BC 5	0.6128	0.6317	0.4141	0.6276
BC 6	0.6149	0.6323	0.4687	0.6314
BC 7	0.6125	0.6281	0.4885	0.5803
BC 8	0.6190	0.6242	0.5288	0.6425
BC 9	0.6202	0.6168	0.4067	0.6151
BC 10	0.6338	0.6435	0.3593	0.5989
MV	**0.6212**	**0.6314**	**0.4707**	**0.6482**
SB	**0.6338**	**0.6418**	**0.5288**	**0.6605**

Where Ens. – Ensemble, BC – Base Classifier, MV – Majority Vote, SB – Single Best

From Table 1 it can be observed that in all cases single best had better results than majority voting. It is also observed that in general, results from ensemble 3 were the poorest, while ensemble 4 yielded the best results. To get a better appreciation of the differences between these results, a binomial test of significance was carried out for each ensemble to ascertain the pairwise difference between majority voting and single best, results of which are illustrated in Table 2.

Table 2: Binomial Test of Significance between Majority Vote and Single Best

Ensemble	**|z|**
1	0.99
2	0.82
3	4.43
4	0.99

In the simple case of determining if there is a difference between two classifications (2 sided test), the null hypothesis (H_o) that there is no significant difference will be rejected if $|Z| > 1.96$ (Congalton *et al*., [24]; Rosenfield *et al*. [25]; Congalton *et al*. [26]). In this case, it is only in ensemble 3 that there is a significant difference between majority vote and single best approaches. The same test was carried out to establish if there was any significant difference between the different ensembles, the results of which are shown in Table 3 and 4. Table 3 depicts the pairwise difference between the ensembles based on the majority vote values, while Table 4 refers to single best values. In both tables E1, E2, E3 and E4 refer to Ensembles 1,2,3 and 4 respectively.

Table 3: Binomial tests of significance between the different ensembles based on majority vote values

	E1	**E2**	**E3**	**E4**
E1	0			
E2	0.80	0		
E3	11.65	12.47	0	
E4	2.14	1.33	13.84	0

Table 4: Binomial tests of significance between the different ensembles based on single best values

	E1	**E2**	**E3**	**E4**
E1	0			
E2	0.63	0		
E3	8.15	8.79	0	
E4	2.13	1.49	10.31	0

From Tables 3 and 4, it can be seen that the results of ensemble 4 are significantly better than the results from ensemble 1 and 3. Whereas the results of ensemble 4 are better than ensemble 2, the difference is not significant. The results from

ensemble 3 are significantly worse than all the results of the ensembles 1,2 and 4.

Of the ensembles considered, evidently the one based on random selection yielded the best classification results. Sequentially selecting bands into base classifiers yielded significantly poorer results. Feature selection resulted in better classification results compared to sequentially selecting the features, however ensemble 2 performed better than ensemble 1. This may have been as a result of using more features in each base classifier. The difference however was not significant

6. Conclusions

The results show that to effect ensemble classification through feature selection for hyperspectral data, generation of base classifiers can best be done using the random selection of features. This however comes with a disadvantage of not being able to exactly replicate previous results. The other methods used in this research however provided a more 'controlled environment' to explore feature selection. Of the said methods, building the base classifiers through sequentially arranging the features resulted in the poorest results. Feature selection through exhaustive search always yielded comparatively better results. Of Ensembles 1 and 2, Ensemble 2 yielded better results. As mentioned before this may be as a result of the base classifiers in Ensemble 2 having more features than Ensemble 1. The significance of the number of features per base classifier pales when it is observed that Ensemble 3 which had 18 features per base classifier performed poorer than Ensembles 1 and 2 which had 10 and 14 features per base classifier. Of the combination methods, apart from Ensemble 3 which proved to be a poorly constituted ensemble, there was no significant difference between majority voting and single best. However, single best always yielded comparatively better results.

7. Acknowledgements

The authors would like to acknowledge the support of the University of the Witwatersrand, Department of Science and Technology and the anonymous reviewers.

8. References

[1] Polikar, R. 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, pp 21 – 44

[2] Opitz, D. 1999. Feature Selection for Ensembles. In *Proceedings of the 16th National Conference on Artificial/ Intelligence (AAAI)*, Orlando-Florida, USA, pp 379-384

[3] Tsymbal, A., Pechenizkiy, M., and Cunningham, P. 2005. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1), pp 83 – 98

[4] Foody, G.M., Boyd, D.S. and Sanchez-Hernandez, C. 2007. Mapping a specific class with an ensemble of classifiers. *International Journal of Remote Sensing,* 28(8), pp 1733 – 1746

[5] Chen, Y., Crawford, M. and Ghosh, J. 2007. Knowledge Based Stacking of Hyperspectral Data for Land Cover Classification, In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, pp. 316-322

[6] Bruzzone, L., Cossu, R. and Vernazza, G., 2004. Detection of land-cover transitions by combining multidate classifiers. *Pattern Recognition Letters* 25, pp. 1491 – 1500

[7] Liu, B., Cui, Q., Jiang, T. and Ma, S. 2004. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatic,* 5:136

[8] Breiman, L. 1996. Bagging predictors. *Machine Learning*, 24(2), pp 123 – 140

[9] Freund, Y., and Schapire, R. 1996. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine*

Learning, Bari, Italy, pp 148 – 156. (Morgan Kaufmann)

[10] Oza, C. N. and Tumer, K. 2008. Classifier ensemble: Select real – world applications. *Information Fusion*, vol. 9, pp 4 – 20

[11] Ho, T. K. 1998. The random subspace method for constructing decision forests. IEEE Transactions of Pattern Analysis and Machine Intelligence, 20 (8), pp 832 – 844

[12] Giacinto, G., & Roli, F. 2001. Design of effective neural network ensembles for image classification processes. *Image Vision and Computing Journal*, 19:9/10, pp 699 – 707

[13] Valentini, G., and Masulli, F. 2002. Ensembles of learning machines. In: Neural Nets WIRN Vietri, *Lecture Notes in Computer Sciences*, edited by; Tagliaferri, R and Marinaro, M., vol. 2486, pp 3 - 19

[14] Huang, Z and Lees, B.G., (2004). Combining non-parametric models for multisource predictive forest mapping. *Photogrammetric Engineering and Remote Sensing,* vol.70, pp 415 – 425

[15] Benediktsson, J.A. and Swain, P.H. 1992. Consensus theoretic classification methods. *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, pp. 688 - 704

[16] Džeroski, S. and Zenko, B. 2004. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning*. 54(3) pp 255 – 273 (Hingham, MA, USA: Kluwer Academic Publishers)

[17] Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. (New York: Springer-Verlag)

[18] Christianini, N., and Shawe-Taylor, J. 2000. *An introduction to support vector machines: and other kernel-based learning methods*. (Cambridge and New York: Cambridge University Press)

[19] Campbell, C. 2000. *An Introduction to Kernel Methods, Radial Basis Function Networks: Design and Applications*. (Berlin: Springer Verlag)

[20] AVIRIS, Airborne visible/infrared imaging spectrometer. [Online]. Available: http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/

[21] Bazi, Y., and Melgani, F. 2006. Toward an optimal SVM classification system for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing,* vol. 44, Issue 11, pp 3374 – 3385

[22] Bhattacharyya, A. 1943. On a measure of divergence between two statistical populations defined by probability distributions, *Bull. Calcutta Math. Soc.*, vol. 35, pp 99 – 109

[23] Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement,* 20, pp 37 – 46

[24] Congalton, R. G., Oderwald, R. G., and Mead, R. A. 1983. Assessing Landsat classification accuracy using discrete multivariate analysis statistical techniques. *Photogrammetric Engineering and Remote Sensing,* 49, pp 1671 – 1678

[25] Rosenfield, G.H, and Fitzpatrick-Lins, K. 1986. A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering and Remote Sensing*, 52, pp 223 – 227

[26] Congalton, R. G., and Green, K. 1998. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices.* (Boca Raton, Florida: Lewis Publishers)

The hitchhiker's guide to the particle filter

McElory Hoffmann, Karin Hunter, Ben Herbst

Department of Mathematical Sciences
University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa
{mcelory, karin, herbst}@sun.ac.za

Abstract

Suppose one wants to model a dynamic process that is contaminated by noise, *i.e.* one seeks the state of the process given some noisy measurements. From a Bayesian point of view, the aim is to find the joint probabilistic density function (pdf) of the state and measurement vector; a complete solution for the problem. Conceptually this problem can be solved by the recursive Bayes filter. If the relevant pdfs are Gaussian and the processes are linear, this conceptual solution is the Kalman filter. However, for more general cases, *e.g.* the case when processes are non-linear and the pdfs are multi-modal, the exact solution is intractable due to insolvable integrals. Monte Carlo methods provide a numerical solution for these intractable integrals. The Monte Carlo approximation of the recursive Bayes filter is known as the particle filter.

The concepts presented here have been extensively investigated in the literature. Our aim is to provide a concise summary of the theory of particle filters, together with an application in tracking and references for further reading.

1. Introduction

Suppose one wants to model a dynamic process that is contaminated by noise, for example tracking an object through an image sequence. The process is usually described by a state vector at time t denoted by $\boldsymbol{x}_t \in \mathbb{R}^{n_x}$. Furthermore, suppose the state vector $\boldsymbol{x}_t$ is not observed directly, but is known through some noisy measurements $\boldsymbol{z}_t \in \mathbb{R}^{n_z}$ and knowledge of the dynamic evolution of the system. Using all the available information, *i.e.*, all measurements and knowledge of the dynamic process, the aim is to find the best possible estimate for the state $\boldsymbol{x}_t$.

In particular, we assume that the states evolve according to

$$\boldsymbol{x}_t = \boldsymbol{f}_{t-1}\left(\boldsymbol{x}_{t-1}, \boldsymbol{v}_{t-1}\right) \tag{1}$$

where $\boldsymbol{f}_{t-1}$ is a known, possibly non-linear function and $\boldsymbol{v}_{t-1}$ is the process noise. The target state is related to the measurements via the measurement equation

$$\boldsymbol{z}_t = \boldsymbol{g}_t\left(\boldsymbol{x}_t, \boldsymbol{w}_t\right) \tag{2}$$

where $\boldsymbol{g}_t$ is again a known, possibly non-linear function and $\boldsymbol{w}_t$ is the measurement noise. The state equation (1) describes the transitional probability, $p\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}\right)$, whereas the likelihood $p\left(\boldsymbol{z}_t|\boldsymbol{x}_t\right)$ is depicted by the measurement equation (2).

A special case is the linear Gaussian dynamic system when equations (1) and (2) reduce to

$$\boldsymbol{x}_t = F_{t-1}\boldsymbol{x}_{t-1} + \boldsymbol{v}_{t-1} \tag{3}$$

$$\boldsymbol{z}_t = G_t\boldsymbol{x}_t + \boldsymbol{w}_t, \tag{4}$$

with $\boldsymbol{v}_{t-1}$ and $\boldsymbol{w}_t$ Gaussian distributed random variables. The exact solution to equations (3) and (4) is given by the Kalman filter [1]. In this case F_{t-1} is called the state transition matrix and G_t the measurement matrix.

The stochastic filtering problem described by Equations (1) and (2) can also be depicted as a Bayesian network, illustrated in Figure 1. Here we clearly see that the current state $\boldsymbol{x}_t$ depends on the previous state $\boldsymbol{x}_{t-1}$ and the measurement at time t, $\boldsymbol{z}_t$, depends on the state $\boldsymbol{x}_t$, but is independent of the measurements at other time steps. This is consistent with equations (1) and (2).

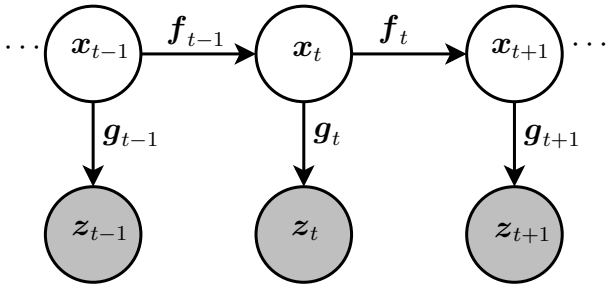


Figure 1: Graphical model of stochastic filtering

The aim of stochastic filtering is thus to find the pdf $p\left(\boldsymbol{x}_t|\boldsymbol{z}_t\right)$ that is a complete solution for the problem. However, this is often intractable since typically $p\left(\boldsymbol{x}_t|\boldsymbol{z}_t\right)$ is a density function and not a point estimate. Nevertheless, it is instructive to understand the exact conceptual solution. Here we use the conceptual solution (discussed in Section 2.1) as a starting point in the development of the ideas underlying a Monte Carlo approximation to the problem: the particle filter (presented in Section 2.3). En route we present the Kalman filter as the exact solution to the special case described by Equations (3) and (4) in Section 2.2 and review Monte Carlo methods in Section 2.3.1. Algorithmic issues are discussed in Section 3, followed by an example in Section 4.

The concepts presented here have been extensively investigated in the literature. The goal of this paper is to provide a concise summary of the theory of particle filters, together with an application in tracking to aid in the understanding of the topic. We also provide references for further investigation.

2. Recursive Bayes Filter

In this section we discuss the conceptual solution to the stochastic filtering problem described by Equations (1) and (2) [2, 3], the Kalman filter as the exact solution of a special case and the particle filter as a numerical approximation.

2.1. The conceptual solution

We use the notation $\boldsymbol{x}_{0:t}$ to denote the set of states, up to and including the state at time t, *i.e.*, $\boldsymbol{x}_{0:t} \triangleq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$. (An alternative notation used in the literature is $X_t \triangleq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$, for example in [4] and [5].)

Implicit in the model described above, are the assumptions that the states follow a first order Markov process, that is $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, and that the measurements are independent of each other.

The goal in solving the stochastic filtering [6] problem in a Bayesian framework, is finding the posterior pdf of the states given the measurements, $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$. This posterior pdf contains all the information about the hidden states and can thus be used to find estimates of the state. The recursive Bayesian filter provides a formal way to propagate the posterior pdfs over time if an initial condition is assumed.

In order to see how the recursive Bayesian filter operates, let us consider the posterior pdf $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$ at time t. Using Bayes' rule[1], we obtain

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t}) = \frac{p(\boldsymbol{z}_{0:t}|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_{0:t})}.$$

By first using the definition of $\boldsymbol{z}_{0:t}$, followed by the product rule[2] and another application of Bayes' rule, we have

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t}) &= \frac{p(\boldsymbol{z}_t, \boldsymbol{z}_{0:t-1}|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t, \boldsymbol{z}_{0:t-1})} \\ &= \frac{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1}, \boldsymbol{x}_t)\, p(\boldsymbol{z}_{0:t-1}|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{z}_{0:t-1})} \\ &= \frac{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1}, \boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{x}_t)}{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{x}_t)}. \end{aligned}$$

Cancelling terms and using the initial assumptions of independence, we obtain a recursive formula for the posterior pdf,

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})}. \tag{5}$$

Note that, even though we have assumed $\boldsymbol{z}_t$ to be independent of $\boldsymbol{z}_{0:t-1}$, we leave the denominator as $p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})$. This is purely to make the following derivations easier.

The recursive formula for the posterior pdf (5) consists of the prior $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$, the likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and the model evidence $p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})$. Using the state transition pdf $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, the posterior at time $t-1$, $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{0:t-1})$, and marginalising[3] over $\boldsymbol{x}_{t-1}$ the prior is written as

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\; p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{0:t-1})\, d\boldsymbol{x}_{t-1}. \tag{6}$$

The likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is the probability of the measurement given the current state, *i.e.*, how likely the measurement $\boldsymbol{z}_t$ is. The evidence normalises the pdf and is therefore calculated as

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1}) = \int p(\boldsymbol{z}_t|\boldsymbol{x}_t)\; p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})\, d\boldsymbol{x}_t. \tag{7}$$

Using the posterior pdf at time t, it is possible to calculate several estimates for the state. One such estimate is the conditional mean

$$\overline{\boldsymbol{x}}_{j|k} \triangleq \mathbb{E}[\boldsymbol{x}_j|\boldsymbol{z}_{0:k}] = \int \boldsymbol{x}_j \cdot p(\boldsymbol{x}_j|\boldsymbol{z}_{0:k})\; d\boldsymbol{x}_j; \tag{8}$$

another is the conditional variance

$$P_{j|k} \triangleq \mathbb{E}\left[\left(\boldsymbol{x}_j - \overline{\boldsymbol{x}}_{j|k}\right)\left(\boldsymbol{x}_j - \overline{\boldsymbol{x}}_{j|k}\right)^T |\boldsymbol{z}_{0:k}\right]. \tag{9}$$

In all the cases we will consider, $j \leqslant k$.

The recursive Bayesian filter is seldom implemented because the analytical solutions of (6) and (7) are intractable.

2.2. The Kalman filter

Thus far we have presented the conceptual solution to the recursive Bayesian filter. In the special case of a linear Gaussian system, the recursive Bayesian filter reduces to the Kalman filter [1, 7]. Here we present the Kalman filter, as viewed from the recursive Bayes point [2, 3]. Our derivation is similar to Chen [8].

The Kalman filter operates in two steps (this can also be said for the recursive Bayes filter). In the first step, we calculate the likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and prior $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$ without seeing the measurement $\boldsymbol{z}_t$. We will refer to this step, as the time update. In the second step, the measurement update (5) is updated in the light of the new measurement, using the values obtained during the time update. These two steps are repeated one after another in order to obtain a recursive formulation.

For the linear Gaussian system, we assume that the process and measurement models are given by (3) and (4) respectively, listed again for convenience:

$$\begin{aligned} \boldsymbol{x}_t &= F_{t-1}\boldsymbol{x}_{t-1} + \boldsymbol{v}_{t-1} \\ \boldsymbol{z}_t &= G_t\boldsymbol{x}_t + \boldsymbol{w}_t. \end{aligned}$$

We denote a Gaussian distribution with mean $\boldsymbol{m}$ and covariance C as $\mathcal{N}(\boldsymbol{m}, C)$. Using this notation, we assume that $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, Q_t)$ and $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, R_t)$ and that $\boldsymbol{v}_t$ and $\boldsymbol{v}_{t'}$ are independent for $t \neq t'$. Similarly, $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t'}$ are assumed to be independent for $t \neq t'$. We also assume that the noise $\boldsymbol{v}_t$ and $\boldsymbol{w}_t$ are independent. These assumptions imply that $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ are Gaussian random variables and that they are independent at different time steps. Since $\boldsymbol{x}_t$ is a Gaussian random variable, it is only necessary to calculate the mean $\overline{\boldsymbol{x}}_{t|t}$ and the covariance $P_{t|t}$ to fully describe the pdf of $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$. The reader is reminded that the mean (first order moment) and covariance (second order moment) are sufficient statistics for a Gaussian distribution.

Given our notation, the goal of Kalman filtering is to propagate $\boldsymbol{x}_{t-1} \sim \mathcal{N}\left(\overline{\boldsymbol{x}}_{t-1|t-1}, P_{t-1|t-1}\right)$ to $\boldsymbol{x}_t \sim \mathcal{N}\left(\overline{\boldsymbol{x}}_{t|t}, P_{t|t}\right)$ using all available information. This will be done in two steps as described above.

2.2.1. Time update

Since $p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})$ in (5) is only a normalising factor, we will only calculate the statistics for the prior and likelihood.

Consider the likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$. The mean is given by

$$\begin{aligned} \overline{\boldsymbol{z}}_t &\triangleq \mathbb{E}[\boldsymbol{z}_t|\boldsymbol{x}_t] \\ &= \mathbb{E}[G_t\boldsymbol{x}_t + \boldsymbol{w}_t|\boldsymbol{x}_t] \\ &= G_t\mathbb{E}[\boldsymbol{x}_t|\boldsymbol{x}_t] \\ &= G_t\boldsymbol{x}_t. \end{aligned} \tag{10}$$

For the covariance we have that

$$\mathbb{E}\left[\left(\boldsymbol{z}_t - \overline{\boldsymbol{z}}_t\right)\left(\boldsymbol{z}_t - \overline{\boldsymbol{z}}_t\right)^T |\boldsymbol{x}_t\right] = R_t. \tag{11}$$

[1] Bayes' rule: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$.

[2] Product rule: $P(A, B) = P(A|B)\, P(B)$.

[3] Also known as the Chapman-Kolmogorov equation.

The second term to consider in (5) is $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$. The mean can be shown to be

$$\begin{aligned}\overline{\boldsymbol{x}}_{t|t-1} &= \mathbb{E}[\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1}] \\ &= \mathbb{E}\left[F_t\boldsymbol{x}_{t-1|t-1} + \boldsymbol{v}_{t-1}|\boldsymbol{z}_{0:t-1}\right] \\ &= F_t\mathbb{E}[\boldsymbol{x}_{t-1}|\boldsymbol{z}_{0:t-1}] + \mathbb{E}[\boldsymbol{v}_{t-1}|\boldsymbol{z}_{0:t}] \\ &= F_t\overline{\boldsymbol{x}}_{t-1|t-1}. \end{aligned} \tag{12}$$

To derive the covariance, we first consider the state prediction error $\widetilde{\boldsymbol{x}}_{t|t-1}$. This is given by

$$\begin{aligned}\widetilde{\boldsymbol{x}}_{t|t-1} &= \boldsymbol{x}_t - \overline{\boldsymbol{x}}_{t|t-1} \\ &= F_t\boldsymbol{x}_{t-1} + \boldsymbol{v}_{t-1} - F_t\overline{\boldsymbol{x}}_{t-1|t-1} \\ &= F_t\widetilde{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{v}_{t-1}. \end{aligned} \tag{13}$$

Now the covariance can be calculated as

$$\begin{aligned}P_{t|t-1} &= \mathbb{E}\left[\widetilde{\boldsymbol{x}}_{t|t-1}\left(\widetilde{\boldsymbol{x}}_{t|t-1}\right)^T\right] \\ &= \mathbb{E}\left[\left(F_t\widetilde{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{v}_{t-1}\right)\left(F_t\widetilde{\boldsymbol{x}}_{t-1|t-1} + \boldsymbol{v}_{t-1}\right)^T\right] \\ &= F_t\mathbb{E}\left[\widetilde{\boldsymbol{x}}_{t-1|t-1}\left(\widetilde{\boldsymbol{x}}_{t-1|t-1}\right)^T\right]F_t^T + \mathbb{E}\left[\boldsymbol{v}_{t-1}\boldsymbol{v}_{t-1}^T\right] \\ &= F_tP_{t-1|t-1}F_t^T + Q_t. \end{aligned} \tag{14}$$

So far we have updated (5) without a new measurement. Now we observe $\boldsymbol{z}_t$ and use it to adjust the pdf $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$. This is discussed next.

2.2.2. Measurement update

From the information calculated in the time step, we have

$$\begin{aligned}p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t}) &\propto p(\boldsymbol{z}_t|\boldsymbol{x}_t)\,p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1}) \\ &= \mathcal{N}(G_t\boldsymbol{x}_t, R_t)\,\mathcal{N}\left(\overline{\boldsymbol{x}}_{t|t-1}, P_{t|t-1}\right). \end{aligned} \tag{15}$$

Our goal is to find the sufficient statistics, *i.e.*, the mean and the covariance of (15). The mean of a Gaussian pdf is equivalent to the value that maximises the underlying exponential. This is the same value where all the derivatives of (15) vanish. Thus

$$\frac{\partial \log p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})}{\partial \boldsymbol{x}_t} = 0 \tag{16}$$

when $\boldsymbol{x}_t = \overline{\boldsymbol{x}}_{t|t}$. Solving (16) gives

$$\overline{\boldsymbol{x}}_{t|t} = \overline{\boldsymbol{x}}_{t|t-1} + K_t\left(\boldsymbol{z}_t - G_t\overline{\boldsymbol{x}}_{t|t-1}\right) \tag{17}$$

where

$$K_t = F_tP_{t|t-1}G_t^T\left(G_tP_{t|t-1}G_t^T + R_t\right)^{-1}. \tag{18}$$

In order to calculate the covariance, we again consider the state prediction error,

$$\begin{aligned}\widetilde{\boldsymbol{x}}_{t|t} &= \boldsymbol{x}_t - \overline{\boldsymbol{x}}_{t|t} \\ &= \boldsymbol{x}_t - \overline{\boldsymbol{x}}_{t|t-1} - K_t\left(\boldsymbol{z}_t - G_t\overline{\boldsymbol{x}}_{t|t-1}\right) \\ &= \widetilde{\boldsymbol{x}}_{t|t-1} - K_t\left(G_t\boldsymbol{x}_t + \boldsymbol{w}_t - G_t\overline{\boldsymbol{x}}_{t|t-1}\right) \\ &= \widetilde{\boldsymbol{x}}_{t|t-1} - K_t\left(G_t\widetilde{\boldsymbol{x}}_{t|t-1} + \boldsymbol{w}_t\right) \\ &= \left(I - K_tG_t\right)\widetilde{\boldsymbol{x}}_{t|t-1} - K_t\boldsymbol{w}_t. \end{aligned} \tag{19}$$

Now we have that

$$\begin{aligned}P_{t|t} &= \mathbb{E}\left[\widetilde{\boldsymbol{x}}_{t|t}\left(\widetilde{\boldsymbol{x}}_{t|t}\right)^T\right] \\ &= \left(I - K_tG_t\right)P_{t|t-1}\left(I - K_tG_t\right)^T + K_tQ_tK_t^T. \end{aligned} \tag{20}$$

$P_{t|t}$ is in the Joseph norm, (20) can be rewritten to other forms used elsewhere in the Kalman filter literature.

In summary, during the time update we propagate the pdf $p(\boldsymbol{x}_{t-1}|\boldsymbol{z}_{0:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$ using (12) and (14). Then a new measurement becomes available. By using (17), (18) and (20) we propagate the pdf $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$.

2.3. The Particle Filter

As we discussed in Section 2.1, exact inference in the Bayesian filter is not in general possible due to the intractable integrals. In general, $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$ could be multivariate, multi-modal or even non-standard, in these cases one has to resort to Monte Carlo techniques to approximate the integrals. Hence we proceed providing an overview of Monte Carlo (MC) methods. They form the cornerstone for the numerical approximations of the recursive Bayesian filter. Thereafter we apply the MC techniques to the recursive Bayesian filter resulting in Sequential Importance Sampling (SIS) [9, 4, 5], also known as the particle filter.

2.3.1. Monte Carlo Methods

Loosely following the notation of Bishop [10], we provide an overview of Monte Carlo (MC) methods.

In the MC framework, we wish to estimate the expected value of a function $f(\boldsymbol{x})$ with pdf $p(\boldsymbol{x})$,

$$\mathbb{E}[f] = \int f(\boldsymbol{x})\,p(\boldsymbol{x})\,d\boldsymbol{x}. \tag{21}$$

We assume that N independent samples $\boldsymbol{x}^{(i)}$, with $i = 1, \ldots N$, drawn from $p(\boldsymbol{x})$ are available. Then the expectation in (21) is approximated by

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{x}^{(i)}\right), \tag{22}$$

that is by the empirical mean of the samples under the function f.

The MC techniques suffer from several problems. Amongst others, it may difficult or impossible to sample from p; in this case one can use importance sampling. The idea behind importance sampling is to use a proposal density function $q(\boldsymbol{x})$ that is easy to sample from, instead of $p(\boldsymbol{x})$. The support of the proposal pdf should be the same as $p(\boldsymbol{x})$, *i.e.*,

$$p(\boldsymbol{x}) > 0 \implies q(\boldsymbol{x}) > 0. \tag{23}$$

Now we sample N independent samples from $q(\boldsymbol{x})$. We can write the expectation in (21) as

$$\begin{aligned}\mathbb{E}[f] &= \int f(\boldsymbol{x})\,p(\boldsymbol{x})\,d\boldsymbol{x} \\ &= \int f(\boldsymbol{x})\,\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})\,d\boldsymbol{x} \\ &\approx \frac{1}{N}\sum_{l=1}^{N}\frac{p\left(\boldsymbol{x}^{(i)}\right)}{q\left(\boldsymbol{x}^{(i)}\right)}f\left(\boldsymbol{x}^{(i)}\right). \end{aligned} \tag{24}$$

The importance sampling estimate in (24) is similar to the MC estimate (22). The only difference is the additional factor, $\frac{p\left(\boldsymbol{x}^{(i)}\right)}{q\left(\boldsymbol{x}^{(i)}\right)}$ that corrects the bias since we are not sampling from

$p(\boldsymbol{x})$; we define this as the importance weights

$$w^{(i)} \triangleq \frac{p\left(\boldsymbol{x}^{(i)}\right)}{q\left(\boldsymbol{x}^{(i)}\right)}. \tag{25}$$

Suppose further that p can only be evaluated up to a normalising constant, such that

$$p(\boldsymbol{x}) = \frac{\widetilde{p}(\boldsymbol{x})}{Z_p},$$

where $\widetilde{p}$ can be easily evaluated and Z_p is the normalising constant. Similarly, we assume that q can be evaluated up to a normalising constant Z_q where $\widetilde{q}$ can be easily evaluated,

$$q(\boldsymbol{x}) = \frac{\widetilde{q}(\boldsymbol{x})}{Z_q}.$$

Then we calculate the MC estimate as

$$\begin{aligned}\mathbb{E}[f] &= \int f(\boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \frac{Z_q}{Z_p}\int f(\boldsymbol{x})\, \frac{\widetilde{p}(\boldsymbol{x})}{\widetilde{q}(\boldsymbol{x})} q(\boldsymbol{x})\, d\boldsymbol{x} \\ &\approx \frac{Z_q}{Z_p}\frac{1}{N}\sum_{i=1}^{N} \widetilde{w}^{(i)} f\left(\boldsymbol{x}^{(i)}\right)\end{aligned} \tag{26}$$

where $\widetilde{w}^{(i)} = \frac{\widetilde{p}\left(\boldsymbol{x}^{(i)}\right)}{\widetilde{q}\left(\boldsymbol{x}^{(i)}\right)}$.

We proceed by calculating the MC estimate for the normalising factor as

$$\begin{aligned}\frac{Z_p}{Z_q} &= \frac{1}{Z_q}\int \widetilde{p}(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \int \frac{\widetilde{p}(\boldsymbol{x})}{\widetilde{q}(\boldsymbol{x})} q(\boldsymbol{x})\, d\boldsymbol{x} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \widetilde{w}^{(i)},\end{aligned} \tag{27}$$

and hence the weights are given by

$$w^{(i)} = \frac{\widetilde{w}^{(i)}}{\frac{1}{N}\sum_{m=1}^{N} \widetilde{w}^{(m)}}. \tag{28}$$

This result should be emphasised. Equation (27) tells us that if p and q can only be evaluated up to a normalising constant, we can find an approximation for this constant by normalising the importance weights. We will use this fact to simplify the equations when we derive the particle filter.

2.3.2. Sequential Importance Sampling (SIS)

At this point we have introduced all the numerical techniques that is used to approximate the recursive Bayesian filter. The fundamental idea of particle filtering is to approximate the pdf $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$ by a weighted sample set S_t. Thus, suppose N samples $\boldsymbol{x}_t^{(i)}$ from the pdf $p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t})$ are available, with a weight $w_t^{(i)}$ associated with each sample $\boldsymbol{x}_t^{(i)}$ normalised such that $\sum_{i=1}^{N} w_t^{(i)} = 1$. Using (21) and (24) we have that

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{0:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta\left(\boldsymbol{x}_t - \boldsymbol{x}_t^{(i)}\right). \tag{29}$$

To derive the recursive formulation, using a notation similar to Ristic *et al.* [5], we begin by solving a more general problem approximating the pdf $p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t})$. We assume that N samples $\boldsymbol{x}_{0:t-1}^{(i)}$ with associated weights $w_{t-1}^{(i)}$ are available approximating the posterior $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{0:t-1})$. Using Bayes rule, we write the posterior at time t as

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t}) = \frac{p(\boldsymbol{z}_{0:t}|\boldsymbol{x}_{0:t})\, p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{z}_{0:t})}. \tag{30}$$

Consider the likelihood. Using standard rules of probability we have that

$$\begin{aligned}p(\boldsymbol{z}_{0:t}|\boldsymbol{x}_{0:t}) &= p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t-1})\, p(\boldsymbol{z}_{0:t-1}|\boldsymbol{x}_{0:t}) \\ &= p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t-1})\, \frac{p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t-1})\, p(\boldsymbol{z}_{0:t-1})}{p(\boldsymbol{x}_{0:t})}.\end{aligned} \tag{31}$$

Substituting (31) in (30) yields

$$\begin{aligned}p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t}) &= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{0:t-1})\, p(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})} \\ &= \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{0:t-1})} p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{0:t-1}) \\ &\propto p(\boldsymbol{z}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\, p(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{0:t-1}).\end{aligned} \tag{32}$$

Since we use importance sampling, the importance weights in (25) are

$$w_t^{(i)} \propto \frac{p\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{z}_{0:t}\right)}{q\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{z}_{0:t}\right)}. \tag{33}$$

Here we make use of the fact that we can calculate the importance weights only up to a normalising factor and by normalising them, we get an MC approximation for the normalising factor.

We assume that the proposal density factorises as

$$q(\boldsymbol{x}_{0:t}|\boldsymbol{z}_{0:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{0:t})\, q(\boldsymbol{x}_{0:t-1}|\boldsymbol{z}_{0:t-1}). \tag{34}$$

By substituting (32) and (34) in (33), we obtain

$$\begin{aligned}\boldsymbol{w}_t^{(i)} &\propto \frac{p\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{z}_{0:t}\right)}{q\left(\boldsymbol{x}_{0:t}^{(i)}|\boldsymbol{z}_{0:t}\right)} \\ &\propto \frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}_t^{(i)}\right) p\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}\right) p\left(\boldsymbol{x}_{0:t-1}^{(i)}|\boldsymbol{z}_{0:t-1}\right)}{q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)}, \boldsymbol{z}_{0:t}\right) q\left(\boldsymbol{x}_{0:t-1}^{(i)}|\boldsymbol{z}_{0:t-1}\right)} \\ &= w_{t-1}^{(i)} \frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}_t^{(i)}\right) p\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}\right)}{q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)}, \boldsymbol{z}_{0:t}\right)}\end{aligned} \tag{35}$$

We make one further assumption that $q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)}, \boldsymbol{z}_{0:t}\right) = q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{z}_t\right)$, and hence the importance weights are given by

$$\boldsymbol{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(\boldsymbol{z}_t|\boldsymbol{x}_t^{(i)}\right) p\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}\right)}{q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{z}_t\right)}. \tag{36}$$

The resulting algorithm is summarised in Figure (2).

A common simplification often used is to choose the proposal density q as the transitional density, *i.e.*, $q\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{z}_t\right) = p\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}\right)$. Then the importance weights simplify to

$$w_t^{(i)} \propto w_{t-1}^{(i)} p\left(\boldsymbol{z}_t|\boldsymbol{x}_t^{(i)}\right). \tag{37}$$

- Input: Samples $\boldsymbol{x}_{t-1}^{(i)}$ with weights $w_{t-1}^{(i)}$, $i = 1, \ldots, N$. Measurement $\boldsymbol{z}_t$.
- FOR $i = 1 : N$
  - Sample $\boldsymbol{x}_t^{(i)}$ from $q\left(\boldsymbol{x}_t^{(i)} | \boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{z}_t\right)$.
  - Evaluate the importance weights using (36).
- END FOR
- Normalise weights using (28).

Figure 2: Sequential Importance Sampling

3. Algorithmic issues

When the algorithm in Figure 2 is implemented, many times all but one of the weights become zero. The result is that the algorithm performs badly in practice.

Doucet *et al.* [11] proved that when the proposal density q is written as in (34), the variance of the weights increases over time, resulting in unavoidable degeneracy. A measure of the degeneracy phenomenon is the effective sampling size

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}\left(w_t^{(i)}\right)^2}. \tag{38}$$

Note that $\widehat{N}_{eff} = 1$ when all but one weight is zero; $\widehat{N}_{eff} = N$ if the weights are uniform.

A remedy to the problem is resampling: Whenever the effective sampling size falls below a certain threshold N_{thr}, a new set of particles is sampled from the current set, each sample proportional to its weight, *i.e.*, a new sample $\boldsymbol{x}_t^{(i)*}$ is chosen such that

$$P\left\{\boldsymbol{x}_t^{(i)*} = \boldsymbol{x}_t^{(j)}\right\} = w_t^{(j)}. \tag{39}$$

Several resampling methods exist. One can implement resampling directly obeying (39); other alternatives include systematic resampling [12] and residue sampling [13]. Embedding resampling in SIS yields Sequential Importance Resampling (SIR). The algorithm is given in Figure 3.

- Input: Samples $\boldsymbol{x}_t^{(i)}$ with weights $w_t^{(i)}$ obtained from SIS[$\boldsymbol{x}_{t-1}^{(i)}$, $w_{t-1}^{(i)}$, $\boldsymbol{z}_t$].
- Calculate $\widehat{N}_{eff}$ using (38).
- IF $\widehat{N}_{eff} < N_{thr}$
  - Resample such that (39) holds. Any technique can be used.
- END IF

Figure 3: Sequential Importance Sampling

4. An example in tracking

A wide variety of computer vision applications rely on accurate object tracking. For example, video surveillance, traffic monitoring, image sequence (*e.g.* facial expression) analysis and human-computer interfaces all require tracking as a core component. Particle filters are suited for tracking since they can handle multi-modal and non-linear systems. In this example, we consider the simpler case of tracking an object's position in an image [14, 15], rather than the complete outline of the object [4]. Tracking only the position of an object, is also known as blob tracking. Our example follows a discussion similar to the blob tracker implemented by Fleck and Straßer [16].

Using a particle filter for object tracking, requires the specification of the state vector $\boldsymbol{x}_t^{(i)}$ for each particle, the transitional density and the measurement likelihood. Furthermore, the proposal density is chosen as the transitional density leading to the simpler weight update (37).

Since only the object's centre is tracked, the state vector merely consists of the $x - y$ position as well as the corresponding velocities $(\dot{x}, \dot{y})$. Hence the state vector at time t is

$$\boldsymbol{x}_t^{(i)} = \left[x^{(i)}, y^{(i)}, \dot{x}^{(i)}, \dot{y}^{(i)}\right]^T. \tag{40}$$

Note that each particle i has an $x-y$ position and corresponding velocity.

The transitional density predicts the movement of the object from one frame to the next. Not much is known about the movement and therefore a constant velocity model

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{v}_{t-1} \tag{41}$$

is assumed. Here $\boldsymbol{v}_{t-1}$ is zero-mean Gaussian noise with variance σ_v^2. The noise term captures the uncertainty of the moving object. A can be experimentally specified or calculated by a training procedure [17].

Before we can describe the likelihood, it is necessary to discuss the representation of the object being tracked. For blob tracking, the object in question is represented by the histogram of the colour pixels. The pixels around the point $\left(x^{(i)}, y^{(i)}\right)$ are sampled at a given time step t in the HSV space. Next the corresponding histograms $h_{HS}^{(i)}(b)$ and $h_V^{(i)}(b)$ are calculated. Here b is a variable over the bin numbers. The histograms $h_{HS}^{(i)}$ and $h_V^{(i)}$ are then combined into a single histogram $h_t^{(i)}(b)$ using alpha blending. The latter histogram is compared with the histogram $h_0(b)$ of the object calculated at the beginning of the tracking procedure using

$$\rho\left[h_t^{(i)}(b), h_0(b)\right] = \sum_b \sqrt{h_t^{(i)}(b)\, h_0(b)}. \tag{42}$$

Now we calculate the Bhatthacharyya distance

$$d_t^{(i)} = \sqrt{1 - \rho\left[h_t^{(i)}(b), h_0(b)\right]} \tag{43}$$

between the two histograms. The measurement likelihood is finally given by

$$w_t^{(i)} = \exp\left(-\frac{\left(d_t^{(i)}\right)^2}{2\sigma^2}\right). \tag{44}$$

From the likelihood we see that if $\boldsymbol{x}_t^{(i)}$ is close to the object being tracked in the image, the similarity value will be large. Therefore $d_t^{(i)}$ will be small and the corresponding weight $w_t^{(i)}$ will be larger than if $\boldsymbol{x}_t^{(i)}$ is far away from the object being tracked.

5. Conclusions

We discussed the estimation of a process state if only noisy measurements are seen. The recursive Bayesian filter provides a solution to this problem. However, this is seldom implemented due to intractable integrals. In the special case of a linear Gaussian system, the Kalman filter though is the exact solution to the problem, *i.e.*, the recursive Bayesian filter reduces to the Kalman filter. Otherwise one has to use numerical approximations; when Monte Carlo methods are used the result is the particle filter. We demonstrated using the particle filter through a common problem in computer vision.

6. Acknowledgements

The financial assistance of the South African National Research Foundation towards this research is gratefully acknowledged. Opinions expressed and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF.

7. References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] Y. Ho and R. Lee, "A bayesian approach to problems in stochastic estimation and control," *Automatic Control, IEEE Transactions on*, vol. 9, no. 4, pp. 333–339, Oct 1964.

[3] P. Maybeck, *Stochastic models estimation and control Vol. 1 Mathematics in science and engineering*. Academic Press, 1979.

[4] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter—particle filters for tracking applications*, 1st ed. Artech House, 2004.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Application to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[7] P. Zarchan and H. Musoff, *Fundamentals Of Kalman Filtering: A Practical Approach*. Aiaa, 2005.

[8] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *adaptive Syst. Lab., McMaster Univ., Hamilton, ON, Canada.[Online]. Available: http://soma. crl. mcmaster. ca/zhechen/download/ieee bayesian.ps*, 2001.

[9] A. Doucet and N. De Freitas, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[10] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[12] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 1–25, 1996.

[13] J. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *JOURNAL-AMERICAN STATISTICAL ASSOCIATION*, vol. 93, pp. 1032–1044, 1998.

[14] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "A color-based particle filter," in *First International Workshop on Generative-Model-Based Vision GMBV'02, in conjuction with ECCV'02*, 2002, pp. 53–60.

[15] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*. London, UK: Springer-Verlag, 2002, pp. 661–675.

[16] S. Fleck and W. Straßer, "Adaptive probabilistic tracking embedded in a smart camera," in *IEEE Embedded Computer Vision Workshop (ECVW) in conjunction with IEEE CVPR 2005*, vol. 3, 2005, p. 134.

[17] A. Blake and M. Isard, *Active contours*. Springer-Verlag, London, 1998.

Impact Assessment of Missing Data Imputation Models

Dan Golding, Tshilidzi Marwala

School of Electrical & Information Engineering University of the Witwatersrand
Private Bag 3, 2050, Johannesburg

dan@golding.za.net

Abstract

The problem of missing data has recently gained the attention of the artificial intelligence field. Whilst much investigation has been done into many different methods, there is no standardized way of assessing the success of an estimator. At present there are two main approaches of measuring performance. One considers accuracy and the other, statistics. This paper shows that both methods are somewhat lacking and proposes an alternative, impact assessment. This method considers how the data will be used in decision making, and measures the impact that the estimates have on these decisions. The paper uses this methodology to compare two methods, k-Nearest Neighbours (kNN) and one based on auto-encoder neural networks. HIV sero-prevalence data is used, making a Generalized Linear Model (GLM) appropriate for impact assessment. The GLM provides insight into the systems that accuracy and statistical measures do not, such as the impact of estimating each variable and a true reflection of the effect of multiple missing variables. kNN shows a better accuracy whilst the neural network approach performs better statistically. Impact assessment shows the two methods to be equally matched, with kNN handling multiple missing values with more success.

1. Introduction

From machine monitoring to sociological surveying, any field dependent on collecting data from the environment is affected by the problem of missing data. most of modelling techniques fail when presented with incomplete data. Ignoring incomplete instances is an unnecessary sacrifice of information that leads to a bias within the model [1]. Many methods have been developed to estimate these missing data without adulterating the integrity of the set. Options span from statistical methods such as Multiple Imputation (MI) to machine learning based approaches and have shown impressive results [2].

Whilst these methods produce viable outcomes, there is at present no standardized way to benchmark the systems. This paper will use HIV sero-prevalence data from antenatal clinics to compare the various assessment procedures from the literature. This data was chosen as much prior investigation in estimating missing data from it has been conducted [3,4,5] and because the data is heterogeneous, i.e. consists of both nominal and numerical attributes. Assessments based on accuracy are compared with those based on statistics. A novel approach to benchmarking the estimator is proposed by considering the purpose of the data. Data such as this is most often used by governmental agencies or insurance firms to assess the risk of HIV infection and analyse its causes. A Generalized Linear Model (GLM) is used to ascertain the probability of being HIV positive based on demographic inputs [6]. By comparing a target instance passed through this model with a simulation of missing data, the *impact* of the estimation paradigm can be better understood.

Two imputation models are compared, an Auto-encoder Neural Network (ANN) and Genetic Algorithm (GA) based approach and k-Nearest Neighbours (kNN). The ANN-GA paradigm is the most investigated imputation scheme for this data set. [7] has shown that refining the approach to be a local search method, that is to only base its estimates on instances with a high similarity instead of considering the entire data set, out-performed the global search version. kNN is another local search technique that has found wide application in the field of missing data [12]. To better assay the suitability of an impact assessment approach, the GLM is benchmarked against mode substitution and random imputation.

The report begins with a background section (2) explaining the problem of missing data in further detail and describing the most common methods, as well as presenting the data set. This is followed by explanations of the ANN-GA (3) and of kNN imputation (4). An explanation of the three approaches to assessment (5) precedes the results (6). Finally these results are critically analysed and the models are compared (7).

2. Background

2.1. What is missing data?

Consider a set of records coming either from measurements of surveys. Each record, or instance, may contain multiple attributes. A data set is thus easily visualized by considering a matrix whose rows represent instances and whose columns are attributes. Table 1 shows the antenatal clinic data in this manner. Missing data refers to instances in which some attributes are unknown, such as rows 2 and 4 in Table 1.

The problem of missing data has always afflicted researchers. Up until the development of the EM algorithm in 1977 [8], missing data was handled mostly through editing [1]. [9] formalized the problem in 1976 and described three separate *mechanisms* to explain how data can go missing. This work gained widespread attention amongst the statistical world with the publication of [10] in 1986 which refined these ideas, one year prior to the proposal of multiple imputation, a procedure that has arguably become the most popular method for handling missing data. *Mechanism* refers to how the fact that data are missing is related to the actual values of the data. These mechanisms are:

1) *Missing Completely At Random (MCAR)* is the mechanism for a missing datum if its reason for being missing is independent of both observed and unobserved data.
2) *Missing At Random (MAR)* is the mechanism for a missing datum if its reason for being missing is dependent on observed data only.
3) *Missing Not At Random (MNAR)* is the mechanism for a missing datum if its reason for being missing is dependent on unobserved data (such as the missing value itself).

2.2. Popular methods for estimating missing data

This section will detail some of the popular methods available for estimating missing data. More conclusive lists can be found in [1,5,11]. One can consider these methods in two classes: statistical and machine learning.

2.2.1. Statistical methods of data imputation

- *Case Deletion*: Instances with missing data are simply removed from the set. This is only suitable when the percentage of incomplete data within the set is negligible. This method introduces bias into the set and sacrifices potentially useful information [1].
- *Mean, Median or Mode Substitution*: Where a missing attribute is replaced with the mean, median, or for discrete or nominal data, the mode of that attribute from the rest of the set. This method does not predict physically plausible results [11].
- *Hot Deck Imputation*: Assigns a value based on the class of the missing instance. Most commonly this value is chosen randomly from other complete instances sharing an output class [11].
- *Regression Models*: Uses the set of complete data to build a regression model, which is essentially an analytical equation of a hyper-plane describing the data [11]. Regression models breakdown when a simple analytical equation cannot describe the data.
- *Expectation Maximization (EM)*: An advanced statistical algorithm that iteratively and simultaneously estimates the model parameters and the missing values using a maximum likelihood approach [2].
- *Multiple Imputation (MI)*: Multiple imputation is one of the most popular methods for estimating missing data at present. It tries multiple different estimates for every missing value and uses the now completed sets to find the optimal choice [1].

2.2.2. Machine Learning Methods of Data Imputation

- *k-Nearest Neighbours*: kNN is a hot deck method that uses instance based learning to find estimates. kNN is explained in detail in section 4.
- *Decision Trees*: A basic form of machine learning classifier. A tree structure is grown with a route to every possible outcome [5,12].
- *Auto-encoder Neural Networks*: NN create a regression model of the data to find optimal inputs using GA. This method is explained in detail in section 3.
- *Support Vector Machines (SVM)*: SVM form a supervised learning paradigm based on statistical learning theory. These models can handle both classification and regression with a high accuracy [4].

2.3. HIV sero-prevalence data set

Tests are performed on data collected by the South African government in 2001 [13]. To assess the sero-prevalence of HIV, surveys were conducted in public antenatal clinics across the country. Missing data from this set is mostly due to non-response from the participants. Around 25% of instances reported contain some missing data. Many investigations into imputing data from this survey have been performed in the past [2-5,7]. However, all these studies use different metrics to assess success. This is explored further in section 5.

Table 1 shows a sample from the data set. Each instance is comprised of 8 attributes. The following is a brief summary of the attributes:

- *Province*: Nominal data expressing which of South Africa's 9 provinces the data was collected in.
- *Race*: Nominal data expressing a subject's race.
- *Age*: The subject's age at the time of the survey.
- *Education*: The highest school grade completed by a subject. 0 means no schooling while 13 implies a tertiary education.
- *Gravidity*: The number of times a subject has fallen pregnant.
- *Parity*: The number of times a subject has given birth.
- *Father's Age*: The age of the father responsible for the current pregnancy.
- *HIV*: HIV status.

Table 1: *Sample of data set*

Pro	Rac	Age	Edu	Gra	Par	Fat	HIV
GP	AF	23	8	4	2	38	0
MP	WH	20	?	1	0	23	0
NW	AF	40	2	3	1	50	1
WC	CO	?	12	6	3	?	1

3. Neural network based imputation

This method, proposed in [3], uses Neural Networks (NN) to create a model of the data set. The model is trained such that if its inputs are from the data set, its outputs will equal these inputs, however if the inputs do not fit the statistical nature of the set upon which the model was built, there will be a discrepancy between the inputs and the outputs. GA is used to find replacements for missing inputs so as to minimize the error between the inputs and the outputs. Further details on NN and GA can be found in [14] and [15] respectively.

This missing data imputation paradigm proposed by [3] trains an ANN on the complete data set. The incomplete data are then passed into this model (in other words the known attributes X_k), the remaining unknown inputs (the missing values, X_u) are found using the GA. The error between all the inputs and outputs of the model is used as the GA fitness function. GA runs until this error reaches a small enough threshold. Fig. 1 illustrates this system.

The system was refined by [7] to be a local paradigm which showed an improvement in the system's accuracy. Instead of training the ANN on the global data set, the data was broken into categories. Nominal attributes have

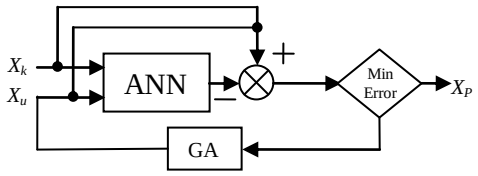


Figure 1: *ANN-GA model for missing data imputation*

obvious categories, i.e. race has African, White etc. Linear attributes are broken up into ranges by grouping for maximum homogeneity. An instance with missing data now falls into multiple categories, for instance if age is missing it falls into all the age categories but in one specific category for each other attribute. An ANN is trained using data only from these categories. Further reading on this paradigm can be found in [2,3,7]

4. k-Nearest neighbours imputation

kNN is an instance based machine learning classification algorithm [12]. Basically, instances are classified by considering the class of the k nearest instances, making it a local paradigm. In the case of nominal or discrete data, the choice is based on the mode. This makes kNN a hot deck method as it chooses a value from an existing similar instance. So while the results might not be perfect they are unlikely to be nonsensical. The kNN method is mostly used in applications such as microarrays where there is a physical difference which decides what is meant by nearest. However the algorithm has also shown promising results on survey data if an apt distance metric is chosen.

kNN fails on nominal data if the Euclidean distance is used. It makes no sense to define a distance between nominal attributes, such as race, using a spatial metric. [16] performed a thorough study concerning different distance metrics suitable for heterogeneous data. The conclusion for data where all the linear data is discrete, such as the HIV set, is to use the Value Difference Metric (VDM), explained in section 4.1.

4.1. Value difference metric

VDM provides a proven distance metric for symbolic variables. The distance between two points x and y according the VDM is given by equation 1.

$$VDM_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q \qquad (1)$$

Where

- $N_{a,x}$ is the number of instances in the set with value x for attribute a.
- $N_{a,x,c}$ is the number of instances in the set with value x for attribute a belonging to output class c. Similarly for $N_{a,y}$ and $N_{a,y,c}$.
- C is the total number of output classes. Note that the assumption is that there is only one output attribute, in this case it is chosen to be HIV status meaning there are 2 output classes in total, HIV positive and HIV negative.
- q is a constant usually chosen as 1 or 2 [16].

VDM can be understood as follows. When considering similarity between two nominal attributes, the output class is an essential consideration. If the output class is type of fruit, and one attribute is colour, for an apple it would make sense that green and red are closer than say, orange [16]. However if the output class is a mango, then green is closer to orange than it is to red. This same thought can be applied to discrete attributes such as age. HIV status in South Africa in 2001 showed a greater prevalence for people in there twenties than it did for teenagers or those above forty [13]. Thus ages 16 and 45 can be considered *closer* than 16 and 22 even though the age gap is far greater. A limitation is that the output class cannot be estimated.

5. Assessment

The focus of this paper is not to compare two methods of missing data estimation, but to develop a suitable method to do so. There has been extensive research in missing data imputation using this data set, however no two papers report their results in the same manner. The approaches in the literature can be separated into two classes, those who use accuracy as an assessment [4,5,7] and those who use statistics [2,3]. The statistics approach taken is to calculate the correlation coefficient, or a Mean Square Error (MSE), between the predicted and the target variables. These statistics are practically accuracy measures as they consider how the values between the target and prediction differ. Bearing in mind that the goal of missing data imputation should not be on finding the exact value every time, but rather on imputing values that maintain the integrity of the entire set [1]. It would be more useful to measure how the probability distributions of the two differ. However to consider quantifying accuracy of a data estimator a futile task is rather short sighted. In data sets coming from surveys, where the action to be taken will involve the entire set at once, accuracy may be unimportant, data sets of scientific measurements such as DNA microarrays or machine monitors are more likely to use individual instances to make decisions thus accuracy becomes an essential consideration.

This paper proposes a third class of assessment for missing data imputations, impact assessment. This method considers what the data set itself is used for, simulates these experiments, and quantifies the impact that estimated data has compared with real measurements. Results based on all three assessment classes are presented in section 6. The remainder of this section will justify how metrics were chosen for each.

5.1. Accuracy

Accuracy is a measure of how closely the estimations mimic their goals. The percentage of estimates that lie within a specific range away from the target is used to quantify this. It is impossible to determine what thresholds will be considered and so this paper measures accuracy with 4 such ranges. These are:

- Percentage predicted exactly
- Percentage predicted to within 1 unit
- Percentage predicted to within 2 units
- Percentage predicted to within 5 units

5.2. Statistics

The goal of a statistical assessment is to determine how alike the properties of the distribution of the target set is to the predictions. Correlation coefficients consider how alike the values are rather than the global properties of the set and this are not suitable as accuracy has already been considered. This assessment considers the following statistics and measures the percentage difference between the targets and the predictions:

- *Mean* - The mean gives information regarding a bias or constant offset.
- *Standard Deviation (SD)* - The SD quantifies the spread of the data around the mean.
- *Quartiles* - The first quartiles is the value which 25% of the data is below, the second is where 50% of the data is below (also know as the median) and the third, 75%. Quartiles can give insight into the skewness of a distribution.
- *Kolmogorov-Smirnov Statistic (KS)* - The KS statistic is defined as the maximum absolute difference between two cumulative distributions. It is essentially a goodness of fit test. The hypothesis in this case is that the estimations have the same distribution as the targets. The KS statistic quantifies the similarity of the distributions [17].

5.3. Impact Assessment

A Generalized Linear Model (GLM) is used to calculate the probability of an HIV positive status given the other 7 inputs described in section 2.3. The discrepancy of this probability given estimated data is an excellent measure of the impact that estimation has. This is for two reason: firstly GLM are widely used on this type of data [6], and secondly the model is built using a large set of complete data thus modelling the global properties of the set yet allows for instance based assessment.

The models find the random distribution of a response variable (HIV status in this case) by relating the predictor variable (all the others) to the expected value of the distribution, $E(\mathbf{Y})$. This process is described by equation 2. Here $\mathbf{X}$ represents the predictor variables and Y, the response. β is a linear combination of the unknown parameters typically found by maximum likelihood and g is known as the link function.

$$E(\mathbf{Y}) = g^{-1}(\mathbf{X}\beta) \tag{2}$$

For a binary distribution such as HIV status, the most common link function is the logit function shown in equation 3. A GLM with this class of link function exactly describes logistic regression. The advantage of GLM over logistic regression is that GLM is generalized and thus appropriate for a wider range of applications. It should be noted that contrary to what the name may imply, GLMs are used for modelling nonlinear data.

$$g(p) = \ln\left(\frac{p}{1-p}\right) \tag{3}$$

The impact that estimated variables leave on the GLM is quantified via the MSE of the results of passing the target data through the GLM and passing the predictions through. The GLM is trained on data that was not used to simulate missing data, meaning all the target data for this test is unseen by the model giving it no advantage.

6. Experiments and Results

This section describes the procedures taken to test the missing data estimators and presents these results. An interpretation of these results is presented in section 7. Two experiments were performed, the first considering only a single attribute missing per instance and the second considering multiple missing attributes. Both these experiments only simulate MCAR and MAR data. MNAR data is a far more complex problem to set up and analyse the results from. The experiments make no distinction between MCAR and MAR data as it is very difficult to make this distinction on a raw data set. The experiments set out to compare the ANN-GA and kNN methods as well as to compare the different methods of assessment.

6.1. Experiment 1 - single missing attribute

For this experiment, each attribute was simulated as missing one at a time. The original data set is cleaned of all missing values via case deletion. Outliers are considered missing values here. The same percentage that is missing in the original data set of the attribute in question is then deleted randomly from this new complete set. This way, the target values for the estimates are known which allows the success of the estimator to be quantified. This procedure is repeated 3 times for each variable, so as to get a truer reflection of the system response, and the results shown are the mean of these runs. Due to the nature of the algorithms, the ANN-GA does not estimate province or race and the kNN does not estimate HIV status. Tables 2 and 3 present the accuracy findings for the ANN-GA and kNN respectively. The bold numbers indicate which system gives a higher corresponding accuracy for each specific case. The statistical results for each system are presented in tables 4 and 5. Note that HIV is expressed as specificity.

Table 6 presents the results from the impact assessment test. In addition to testing ANN-GA and kNN, the GLM method is benchmarked against a random imputation, i.e. a random value within the range of each variable is taken as the estimate, and mode substitution. Bold values represent the method with the least impact for that particular attribute.

6.2. Experiment 2 - multiple missing attributes

To assess the system response to more than one missing value in an attribute, the same experiments described in section 6.1 are repeated but with multiple missing attributes. The results presented show the effect that having age missing in conjunction with the 4 other linear variables has. Other combinations of two missing variables are not reported as they offer little new information and require a large amount of space.
For readability, only one accuracy measure and one statistic is shown for each attribute. These are chosen to reflect the other measures. They are percentage to within 2 units for accuracy and the KS statistic for statistics. Table 9 shows the result of the GLM. Bold values in these tables highlight results that differ significantly to experiment 1.

Both systems perform without a noticeable change in accuracy or statistics for up to 4 missing variables. The impact of estimation increases slightly with every variable deleted.

Table 2: *ANN-GA accuracy with single missing attributes measured as the percentage measured to within n units of the target*

n	Age	Edu	Gra	Par	Fat
0	10.93	13.98	52.78	77.87	7.41
1	32.22	38.70	90.46	**95.74**	25.46
2	51.76	54.44	97.13	**99.07**	43.52
3	**82.22**	79.17	**100**	**100**	**78.24**

Table 3: *kNN accuracy with single missing attributes measured as the percentage measured to within n units of the target*

n	Age	Edu	Gra	Par	Fat
0	**11.67**	**30.00**	**81.30**	**84.03**	**9.14**
1	**35.34**	**53.64**	**93.73**	93.24	**27.84**
2	**54.07**	**70.71**	**97.50**	96.25	**44.14**
3	81.67	**91.73**	99.66	99.63	73.46

Table 4: *Percentage difference for ANN-GA statistics with single missing attributes of targets with predictions*

Statistic	Age	Edu	Gra	Par	Fat
Mean	**0.54**	**6.03**	-19.15	**-10.28**	**0.66**
Standard Deviation	**71.88**	143.07	60.53	**47.47**	**69.02**
KS Statistic	**0.037**	**0.1472**	0.1185	0.0389	**0.1019**
1st Quartile	**-9.52**	**-25.00**	-100	0	-12.00
Median	0	**0**	0	0	**0**
3rd Quartile	**6.67**	0.25	0	0	**8.57**

Table 5: *Percentage difference for kNN statistics with single missing attributes of targets with predictions*

Statistic	Age	Edu	Gra	Par	Fat
Mean	2.55	-15.1	**12.15**	11.88	5.66
Standard Deviation	72.48	**61.63**	**57.68**	65.45	79.30
KS Statistic	0.0531	0.2664	**0.0753**	**0.0269**	0.1210
1st Quartile	-9.67	-31.25	**0**	0	**-8.00**
Median	0	-10.00	0	0	3.37
3rd Quartile	9.43	0	0	0	12.28

Table 6: *Mean square error of targets vs. predictions passed through a generalized linear model for single missing attributes*

Paradigm	Age	Edu	Gra	Par	Fat
ANN-GA	6.64	0.31	1.44	1.69	0.69
kNN	6.56	0.17	1.12	3.45	1.56
Mode Substitution	13.99	0.68	2.88	24.76	1.70
Random Imputation	82.99	0.80	28.29	122.35	11.62

Table 7: A*ccuracy of both age and each other attribute missing measured as the percentage predicted to within 2 units of the target*

	Edu	Gra	Par	Fat
ANN-GA	60.20	96.97	98.79	**20.00**
ANN-GA Age	47.27	**38.38**	47.88	**29.29**
kNN	73.32	97.98	96.97	37.98
kNN Age	51.92	53.54	51.31	46.06

Table 8: *KS Statistic of both age and each other attribute missing*

	Edu	Gra	Par	Fat
ANN-GA	0.0869	0.1859	0.0444	**0.2384**
ANN-GA Age	0.0303	**0.1475**	0.0505	**0.1697**
kNN	0.02889	0.0505	0.0141	0.0990
kNN Age	0.0545	0.0646	0.0566	0.0970

Table 9: *Mean square error of targets vs. predictions passed through a generalized linear model for both age and each other attribute missing*

	Edu	Gra	Par	Fat
ANN-GA	0.26	1.35	2.39	**4.72**
ANN-GA Age	6.70	**13.06**	8.86	**18.80**
kNN	0.14	0.51	4.43	1.29
kNN Age	5.20	4.95	4.55	5.79

7 Comparisons and Discussion

Some interesting conclusions can be drawn from both experiments. From the first experiment, kNN outperforms ANN-GA in terms of accuracy, but ANN-GA shows a better statistical response. Whilst the results do prove the effectiveness of kNN on survey data, they make it very difficult to decide which system exhibits a better response. The impact assessment results however, show the systems to be very equally matched. Thus it appears the GLM test provides a useful and sensible summary of the results from both accuracy and statistics.

Furthermore, the tests of mode substitution and random imputation show that the GLM method can distinguish between imputation methods. The results are as expected, with random imputation showing a poor response, and mode substitution performing better than random imputation but worse than the more sophisticated methods.

The GLM test makes a distinction between the impact of estimating different attributes. Neither of the other assessment methods show this. The results show that whilst the accuracy and statistical predictions of age appear to be within the range of the other attributes, the impact that estimating age has is far greater than any of the other variables. The impact of estimating education level is much lower, showing almost no impact even for random imputation. This indicates that anyone using this data should estimate age with caution, but need not be too concerned about missing education fields.

The tests for multiple missing variables reveal that whilst both systems perform well, kNN does not suffer from the same losses in accuracy that ANN-GA does when two correlated variables are missing. This sort of accuracy loss is best illustrated by the case of age and father's age being missing. Accuracy and statistical success metrics indicate multiple missing values have a negligible effect on the performance of the systems. This has also been reported in [7] and [3]. However, the impact assessment reveals that the response does in fact get poorer for each variable that is missing. This makes sense as, even though the accuracy of each estimate remains unchanged, the accuracy is always lower than a true value.

To summarize, using an impact assessment technique, such as GLM, provides greater insight than either accu-

racy or statistical metrics. The GLM method has shown that whilst kNN and ANN-GA are equally matched for a single missing attribute, kNN tends to perform better on multiple missing values. Furthermore it has illustrated the how estimating each variable has a different impact, with age having the largest and education the lowest. Thus it would appear that an impact assessment technique is preferable as a method to verify an estimation model and GLMs are suitable as such for this data set.

8. Conclusions

Whilst there is now a multitude of paradigms to choose from to address the problem of missing data, there is no standard as to how they are assessed. Thus it is difficult to know which method to apply for a given set of data. This paper has shown that the two main classes of assessment in the literature, namely accuracy measures and statistical measures, have inherent flaws. A novel approach, impact assessment, is suggested. This approach considers how the data is used for decision making. In the case of the HIV sero-prevalence data studied in this paper, impact assessment is done through a Generalized Linear Model (GLM) built to find the probability of being HIV positive given the other attributes as inputs. A neural network approach (ANN-GA) is compared with k-Nearest Neighbours (kNN). The accuracy and statistical measures disagreed as to which was superior. GLM showed them to be equally matched. The impact assessment measure was tested against mode substitution and random imputation. As expected, random imputation showed a very poor response whilst mode substitution placed between random and the sophisticated imputation methods. Impact assessment has been proved to be a more effective measure of the success of data imputation than methods used in the past. It can show which attributes have the greatest effect, provides a truer reflection of the losses incurred with multiple missing points, and provides a useful summary of accuracy and statistical test. The successful imputation of HIV sero-prevalence data allows for more accurate models to be built. The accuracy of such a model is essential for society, as it is what HIV policy is based on. This research has developed a method to successfully choose the best missing data estimation technique for this data set, which can aid government policy better control the HIV/AIDS epidemic. Both ANN-GA and kNN proved effective as data estimator for HIV sero-prevalence data, showing far better results than mode substitution or random imputation. Impact assessment such as GLM has shown to be a superior method to assess the success of an imputation scheme when compared with past methods.

9. References

[1] J. L. Schafer, J. W. Graham, "Missing data: our view of the state of the art", Psychological Methods, Vol. 7, p 147–177, 2002.

[2] F. Nelwamondo, S. Mohamed, T. Marwala, "Missing data: a comparison of neural networks and expectation maximization techniques", Research articles current science, Vol. 93, p 1514-1521, 2007.

[3] M. Abdella, T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data"' Database in Computers and Artificial Intelligence, Vol. 24, 2005.

[4] V. N. Marivate, F. V. Nelwamondo, T. Marwala, "Autoencoder, principal component analysis and support vector regression for data imputation", Proceedings of the 17th World Congress of The International Federation of Automatic Control Seoul, Korea, p 682-689, 2008.

[5] G. Ssali, T. Marwala, "Estimation of missing data using computational intelligence and decision trees", ArXiv e-prints, Vol. 709, 2007.

[6] M. A. Waclawiw, K. Liang, "Prediction of random effects in the generalized linear model", Journal of the American Statistical Association, Vol. 88, p 171 -178, 1993.

[7] F. V. Nelwamondo, D. Golding, T. Marwala, "Data categorization for missing data estimation using neural networks: a dynamic programming-like approach", lecture notes in computer science, Springer-Verlang, 2008. (to appear)

[8] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion)", Journal of the Royal Statistical Society, Series B, 39, p 1–38, 1977.

[9] D. B. Rubin, "Inference and missing data", Biometrika, 63, p 581-592, 1976.

[10] R. J. A Little, D. B. Rubin, "Statistical analysis with missing data", John Wiley & Sons, 1986.

[11] J. M. Engles, P. Diehr, "Imputation of missing longitudinal data: a comparison of methods", Journal of Clinical Epidemiology 56, p 968–976, 2003.

[12] G. E. A. P. A. Batista, M. C. Monard, "A study of k-nearest neighbour as an imputation method", HIS conference proceedings, 2002.

[13] M. E. Tshabalala-Msimang, "Summary report: national HIV and syphilis sero-prevalence survey of women attending public ante-natal clinics in South Africa,2001", Department of Health, Republic of South Africa, 2001.

[14] T. Munakata, "Fundamentals of the new artificial intelligence - neural, evolutionary, fuzzy and more", Springer-Verlang, 2008.

[15] D. E. Goldberg, "Genetic algorithms in search, optimization & machine learning", Addison-Wesley Publishing Co, 1989.

[16] D. R. Wilson, T. R. Martinez, "Improved heterogeneous distance functions", Journal of Artificial Intelligence Research 6, p 1-34, 1997.

[17] I. M. Chakravarti, R. G. Laha, J. Roy, "Handbook of Methods of Applied Statistics. Vol. I: Techniques of Computation", Descriptive Methods and Statistical Inference, John Wiley and Sons, 1967.

A note on the separability index

Linda Mthembu, Tshilidzi Marwala

Department of Electrical and Information Engineering, University of the Witwatersrand
Johannesburg, South Africa
linda.mthembu@student.wits.ac.za, t.marwala@ee.wits.ac.za

Abstract

In discriminating between objects from different classes, the more separable these classes are the less computationally expensive and complex a classifier can be used. One thus seeks a measure that can quickly capture this separability concept between classes whilst having an intuitive interpretation on what it is quantifying. A previously proposed separability measure, the separability index (SI) has been shown to intuitively capture the class separability property very well. This short note highlights the limitations of this measure and proposes a slight variation to it by combining it with another form of separability measure that captures a quantity not covered by the Separability Index.
Keywords: Classification, separability, margins

1. Introduction

In object categorization/classification one is given a dataset of objects from different classes from which to discover a class-distinguishing-pattern so as to predict the classification of new, previously unseen objects [1,7]. This will only be possible if the main justification pillar of induction systems which is based on the dictum; "similar objects tend to cluster together" is true. This process of discovering a pattern in the dataset is further complicated by the fact that the dataset often cannot immediately be visualized to determine the class distribution. This could be due to the datasets' high dimensionality. Discovering a method that can distil such information, *without* running multiple sets of computationally expensive classifiers, would be advantageous.

This method should quantify how the classes are distributed with respect to each other; are there class overlaps, are there multiple modes within the classes and are there many outliers etc? We thus seek a simple measure that can concisely capture some of these aspects of the classes to gauge the complexity of classifier to be implemented. The notion of a 'simpler classifier' relates to the complexity of the discrimination function. A simpler function e.g. linear is preferred over a more complex polynomial function as stated by Occam's razor. The complexity of a classifier is also determined by the number of irrelevant features in the dataset. The original dataset input space – defined by the number of expertly measured attributes - is often not the optimal in terms of producing clearly separable/non-overlapping classes. A subset of this space can often produce a substantially separable set of classes which in turn results in a simpler discriminating function. Searching for an optimal sub-space can be considered an optimization problem whose criterion function is the maximization of some predefined separability measure. A recent review and comment on this area of research is presented in [4 and 6]. One measure, the separability index (SI), that intuitively measures the class overlap was previously introduced in [3, 8] and was shown to be efficient in a number of popular machine learning datasets in [3, 5].

The separability index measure estimates the average number of instances in a dataset that have a nearest neighbour with the same label. Since this is a fraction the index varies between 0-1 or 0-100%. Another separability measure, based on the class distance or margin is the Hypothesis margin (HM), introduced in [2]. It measures the distance between an object's nearest neighbour of the same class (near-hit) and a nearest neighbour of the opposing class (near-miss) and sums over these. This means the larger the near-miss distance and smaller the near-hit values, the larger the hypothesis margin will be.

This note is only concerned with the above two mentioned measures' limitations. In the next section we show with a simple example the behaviour of both the SI and HM. We highlight the advantages and disadvantages of SI and HM then we propose a hybrid of the two measures. The resulting measures' pseudo code and behaviour are presented.

2. Separability

2.1 Behaviour of separability measures

In this section the behaviour of both measures is simulated in an example where the separation of two Gaussian clusters is incrementally increased. This is

taken to simulate the process of searching for an optimal feature space in a given high dimensional dataset. Figure 1 shows two Gaussian clusters that are initially overlapping with a SI of 0.54 or 54%.

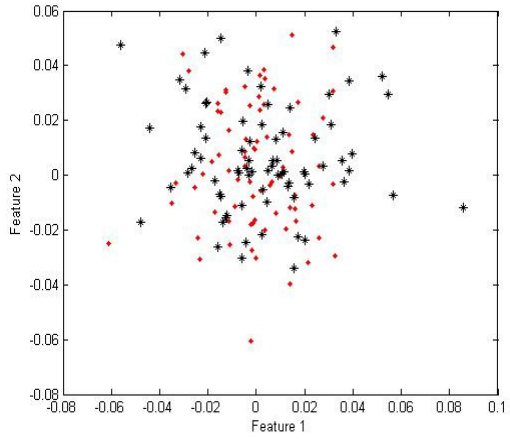


Figure 1: Two initially overlapping classes

These clusters are incrementally separated, by varying one cluster's centre distance from the other. Figure 2 shows the point where the SI measure is 1or 100%; a quadratic or cubic discriminator will certainly be enough to *cleanly* partition the clusters whereas a linear classifier might not without misclassification. Figure 3 shows a state where the two clusters are visually more fully separated than in figure 2 and certainly a linear function will be an adequate classifier for such class separations. Figure 4 shows the variation of the separability index with the increasing cluster distance.

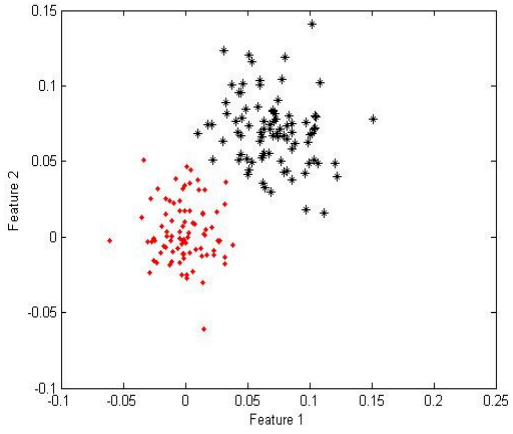


Figure 2: The Separability index is 100%

When the class separation distance increases beyond 0.015 units the SI still reports a separability of 1. It is clear from this figure that the SI is limited in capturing extreme class separability information which could result when a feature sub-space with fewer features than that at 100% separability is discovered in the optimization.

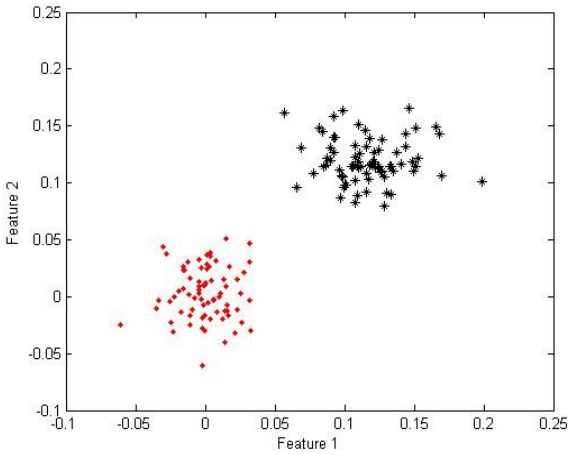


Figure 3: Increased class separability

The SI measure is informative about the separability of the clusters below full separability (<=1) but is no longer informative when the classes separate further which can arise in practise. This is to be expected since the separability index does not measure class distances per se. The hypothesis margin on the other hand, shown in figure 5, keeps on measuring with no real informative limit on the quantity it is measuring except that the class separation distance is increasing. What is required is a measure that has the ability to *intuitively* inform on the class separability below 100%, a characteristic of the separability index and has the ability to continue measuring after 100% class separability, a characteristic of the hypothesis margin.

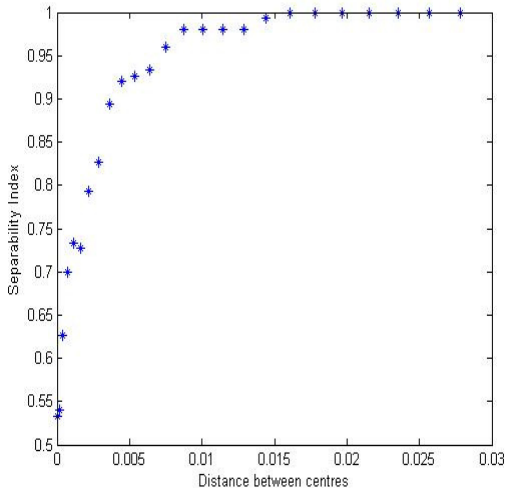


Figure 4: Separability index results on the two Gaussian clusters as the centre distance is increased

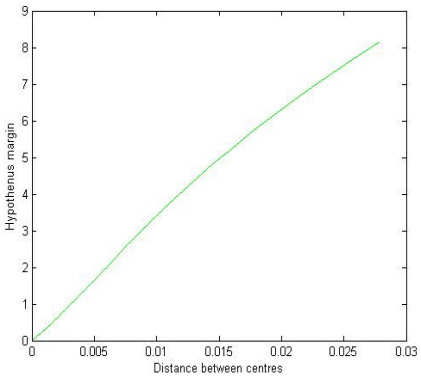


Figure 5: Hypothesis margin results on the two Gaussian clusters as the centre distance is increased

3. The Hybrid separability measure

Merging the two measures will consist of two parts; the original SI and *modified* HM parts. The HM is modified by only initializing it when the separability index measures a separability of 1. While the SI is below 1 the HM is set to zero and once the SI is equal to 1 the HM is activated. Subsequent hypothesis margin distances are then calculated as ratios with respect to the HM when the SI was 1.

In this hybrid measure the SI part will capture all the sub-spaces, from feature selection, where the class separability increases until unity then the modified HM part will capture the fact that the clusters are still separating further. This way the hybrid separability measure captures the overall class separability in terms of distance and instance overlap. Figure 5 shows the pseudo code for the proposed algorithm:

```
hm = hypothesis margin; % original hypothesis margin
si  = separability index;  % separability index
if si < 1
    hybrid = 100*Si;   % hybrid measure equal SI when
                       % SI is less than 1.
    hm_ratio =0;       % hypothesis ratio
    hm = 0;            % hypothesis margin
    counter =0;
elseif si = 1
    counter =counter  +1;

   if counter =1     % first time SI is 1 capture the
     ih = hm;       % hypothesis margin distance to be
              % the reference for subsequent distances
    end
  hm_ratio = hm/ih;          % hypothesis ratio
  hybrid = 100*hm_ratio;  % hybrid measure
end
```

Figure 5: Pseudo code for hybrid measure

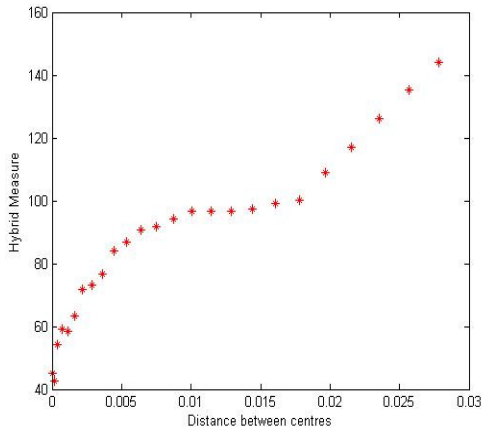


Figure 6: Hybrid measure on the two Gaussian clusters as the centre distance is increased

Figure 6 shows the behaviour of the hybrid separability measure. The SI part is still retained and now the HM part is incorporated as a fraction which is converted to a percentage so as to integrate with the SI measure. The hypothesis margin is now a more informative measure of the class separation. Table 1 below presents a portion of the above simulation results. After the separability index reaches 1 the hypothesis ratio information is relayed to the hybrid measure by multiplying by 100. The separation distance can still be extracted from the hybrid measure.

SI	HM	HM RATIO	Hybrid
			(%)
0.908	1.5431	0	90.8046
0.9368	1.962	0	93.6782
0.954	2.4002	0	95.4023
0.9598	2.8622	0	95.977
0.9828	3.3595	0	98.2759
0.9885	3.8828	0	98.8506
1	4.4158	1	100
1	4.952	1.1214	112.1431
1	5.4955	1.2445	124.4502
1	6.0419	1.3682	136.8238
1	6.5924	1.4929	149.2898
1	7.1469	1.6185	161.8487
1	7.7037	1.7446	174.457
1	8.2627	1.8712	187.1161

Table 1: A sub-set of the simulation results

Intuitive interpretation, in the new measure, is not completely lost and can be derived from the last two columns of table 1. Once the hybrid measure reports separabilities of more than 100% then a different perspective on separability can be induced; the reported quantity will then be the percentage ratio of the class separability distances. A value of 124% can be read to mean the classes are one point two four *times further* apart than they were when the SI index was 100%.

This retains the intuitive notion of average distance between classes (measured by the hypothesis margin (HM)) albeit it is measured from a different reference point, the point at which the separability index (SI) measures 100%.

4. Conclusion

This note highlights the advantages and disadvantages of two previously proposed separability measures; the separability index and the hypothesis margin. A hybrid measure is formed from the two and the good properties of the individual measures are retained in the new measure which overcomes the limitations of the previous measures. A simple simulation example exposes the problem of the two measures and performance results of the new measure are presented on the same example. Some intuitive interpretation can still be developed from the new measure.

5. Acknowledgements

This research was supported by the financial assistance of the National Research Foundation of South Africa.

6. References

[1] R.O. Duda, P. E. Hart and D.G. Stork. Pattern Classification (2nd edition) John Wiley and Sons, 2000.

[2] R. Gilad-Bachrach, A. Navot and N. Tishby. Margin based feature selection- Theory and algorithms. Proc. 21st International Conference on Machine Learning (ICML), Banff, Canada 2004.

[3] Greene J.R. Feature subset selection using Thornton's separability index and its applicability to a number of sparse proximity-based classifiers. In proceedings of the Pattern Recognition Association of South Africa, 2001.

[4] Guyon I and Elisseeff A. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3 pages 1157-1182 (2003).

[5] L. Mthembu and J.R Greene. A comparison of three separability measures. In Proc of the 15th Annual symposium of the Pattern Recognition Association of South Africa (PRASA), November 2004, Grabouw, South Africa.

[6] A. Navot, R. Gilad-Bachrach, Y. Navot and N. Tishby. Is feature selection still necessary? In Saunders C and Grobelnik M and Gunn S and Shawe-Taylor J. Editors Latent structure and feature selection techniques: Statistical and Optimisation perspectives workshop (2006).

[7] T. Mitchell. Machine Learning. Published by McGraw Hill, 1997, ISBN 0070428077.

[8] C. Thornton. Truth from Trash: How Learning Makes Sense. Published by MIT Press, 2002, ISBN 0262700875, 9780262700870.

Extending DTGolog to Deal with POMDPs

Gavin Rens[1,2], Alexander Ferrein[3], Etienne van der Poel[1]

[1] School of Computing, Unisa, Pretoria, South Africa
[2] Knowledge Systems Group, Meraka Institute, CSIR, Pretoria, South Africa
[3] Knowledge-Based Systems Group, RWTH Aachen University, Aachen, Germany

grens@csir.co.za
ferrein@cs.rwth-aachen.de
evdpoel@unisa.ac.za

Abstract

For sophisticated robots, it may be best to accept and reason with noisy sensor data, instead of assuming complete observation and then dealing with the effects of making the assumption. We shall model uncertainties with a formalism called the *partially observable Markov decision process* (POMDP). The planner developed in this paper will be implemented in *Golog*; a theoretically and practically 'proven' agent programming language. There exists a working implementation of our POMDP-planner.

1. Introduction

If a robot or agent can perceive every necessary detail of its environment, its model is said to be *fully observable*. In many practical applications this assumption is good enough for the agent to fulfill its tasks; it is nevertheless unrealistic. A more accurate model is a *partially observable model*. The agent takes into account that its sensors are imperfect, and that it does not know every detail of the world. That is, the agent can incorporate the probabilities of errors associated with its sensors, and other uncertainties inherent in perception in the real world, for example, obscured objects. If an agent or robot cannot represent the uncertainties inherent in perception, it has to *assume* perfect perception. This assumption either might lead to spurious conclusions or the necessity for additional methods that keep the agent's reasoning reasonable. For sophisticated robots or agents, it may be best to accept and reason with noisy sensor data.

One model for reasoning under uncertainty with partial observability is the *partially observable Markov decision process* (POMDP). In this paper we present POMDP models based on the robot programming and planning language Golog [1]. In particular, we extend DTGolog [2], a Golog dialect. DTGolog employs a notion of perfect perception; we extent it with a notion of graded belief.

The rest of the paper is organised as follows. In the next section we briefly introduce the situation calculus and present the robot programming and planning language DTGolog, before we formally define POMDPs in Section 3. In Section 4 we present some related work. Section 5 introduces the predicate $BestDoPO$ which defines the semantics of the POMDP planner in Golog. Section 6 presents a simple example of how planning under partial observability is conducted. We conclude with Section 7.

2. The Situation Calculus and DTGolog

The situation calculus is a first order logic dialect for reasoning about dynamical systems based on agent actions. The outcomes of a bout of reasoning in the situation calculus are meant to have effects on the environment outside the agent. When an agent or robot performs an action, the truth value of certain predicates may change. Predicates whose value can change due to actions are called *fluents*. Fluents have the *situation term* s as the last argument.

A special function symbol do is defined in the situation calculus. $do(a, s)$ is the name of the situation (that the agent is in) given the agent does action a in situation s. Note that $do(a_2, do(a_1, s))$ is also a situation term, where a_2 and a_1 are actions.

To reason in the situation calculus, one needs to define an initial knowledge base (KB). The only situation term allowed in the initial KB is the special *initial situation* S_0. S_0 is the situation before any action has been done.

There are two more formulas that need our attention:

1. The *precondition axioms* are formulas of the form $Poss(a, s)$, which means action a is possible in situation s ($\neg Poss(a, s)$ means it is not possible). Precondition axioms need to be defined for each action.
2. *Successor-state axioms* are formulas that define how fluents' values change due to actions. There needs to be a successor-state axiom for each fluent, and each such successor-state axiom mentions only the actions that have an effect on the particular fluent.

Please refer to [3] for a detailed explication of the situation calculus, including a description of the famous *frame problem* and how the *basic action theory* is a solution to this problem. Alternatively, refer to [4] for a one-chapter coverage of the situation calculus.

Decision-theoretic Golog (DTGolog) [2] is an extension to Golog to reason with probabilistic models of uncertain actions. The formal underlying model is that of fully observable Markov decision processes (MDPs).

Golog is an agent programming language (APL) developed by [1]. It is based on the situation calculus. It has most of the constructs of regular procedural programming languages (iteration, conditionals, etc.). What makes it different from other programming languages is that it is used to specify and control *actions* that are intended to be executed in the real world or a simulation of the real world. That is, Golog's main variable type is the action (not the number).

Complex actions can be specified by combining atomic actions. The following are all complex actions (where a subscripted is an atomic action and φ is a sentence):

- while φ do a_1 (iteration of actions);
- φ ? : a_1 (test action);
- if φ then a_1 else a_2 (conditional actions);
- $a_1; a_2; \ldots; a_k$ (sequence of actions);
- $a_1 \mid a_2$ (nondeterministic choice of actions);
- $\wp x.(a_1)$ (nondeterministic finite choice of arguments—of x in a_1);

$Do(A, s, s')$ holds if and only if the complex action A can terminate legally in s' when started in situation s.

The DTGolog algorithm is defined with $BestDo$ predicates, taking on the role of Golog's Do. The DTGolog interpreter however, does not simply 'perform' the program (complex action) given it, but calculates an optimal policy based on an optimization theory: the forward search value iteration algorithm for fully observable MDPs. [1] capture the nondeterministic aspect of MDPs with predicates $stochastic$, and $prob$. $prob(n, p, s)$ determines the probability p with which action n is the outcome in some situation s. (In this section we define prob as a *function* that *returns* the probability.) Let $choice'(a) \doteq \{n_1, \ldots, n_k\}$ (derived from $stochastic$) be the k actions that nature could 'choose' (the actual action performed) for the agent's *intended* action a. For stochastic action a,

$$BestDo(a; r, s, h, \pi, v, pr) \doteq$$
$$\exists \pi', v'.BestDoAux(choice'(a), a, rest, s, h, \pi', v', pr) \wedge$$
$$\pi = a; senseEffect(a), \pi' \wedge v = reward(s) + v'.$$

$a; r$ is the input program, with a the first action in the program and r the rest of the program; s is the situation term; the agent designer needs to set the number of steps (actions) h for which a policy is sought—the *planning horizon*; π returns the policy; v is the expected reward for executing π; pr returns the probability with which the input program will be executed as specified, given the policy and given the effects of the environment. $senseEffect(a)$ is a pseudo-action included in the formalism to ensure that the formalism stays in the fully observable MDP model. $BestDoAux$ deals with each of the possible realizations of a stochastic action:

$$BestDoAux(\{n_1, \ldots, n_k\}, a, r, s, h, \pi, v, pr) \doteq$$
$$\neg Poss(n_1, s) \wedge BestDoAux(\{n_2, \ldots, n_k\}, a, r, s, h, \pi, v, pr) \vee$$
$$Poss(n_1, s) \wedge$$
$$\exists \pi', v', pr'.BestDoAux(\{n_2, \ldots, n_k\}, a, r, s, h, \pi', v', pr') \wedge$$
$$\exists \pi_1, v_1, pr_1.BestDo(r, do(n_1, s), h - 1, \pi_1, v_1, pr_1) \wedge$$
$$senseCond(n_1, \varphi_1) \wedge \pi = \textbf{if } \varphi_1 \textbf{ then } \pi_1 \textbf{ else } \pi' \textbf{ endif} \wedge$$
$$v = v' + v_1 \cdot prob(n_1, a, s) \wedge pr = pr' + pr_1 \cdot prob(n_1, a, s).$$

For any action n, $senseCond(n, \varphi)$ supplies a sentence φ that is placed in the policy being generated. φ holds if and only if the value returned by the sensor can verify that action n was performed.

When either of two actions δ_1 and δ_2 can be performed, the policy associated with the action that produces the greater value (current sum of rewards) is preferred and that action is included in the determination of the final policy π. This formula captures the idea that is at the heart of the *expected value maximization* of decision theory:

$$BestDo([\delta_1|\delta_2]; r, s, h, \pi, v, pr) \doteq$$
$$\exists \pi_1, v_1, pr_1.BestDo(\delta_1; r, s, h, \pi_1, v_1, pr_1) \wedge$$
$$\exists \pi_2, v_2, pr_2.BestDo(\delta_2; r, s, h, \pi_2, v_2, pr_2) \wedge$$
$$((v_1, \delta_1) \geq (v_2, \delta_2) \wedge \pi = \pi_1 \wedge v = v_1 \wedge pr = pr_1) \vee$$
$$((v_1, \delta_1) < (v_2, \delta_2) \wedge \pi = \pi_2 \wedge v = v_2 \wedge pr = pr_2)).$$

3. POMDP defined

3.1. The model

In partially observable Markov decision processes (POMDPs) actions have nondeterministic results and observations are uncertain. In other words, the effect of some chosen action is somewhat unpredictable, yet may be predicted with a probability of occurrence. And the world is not directly observable; some data are observable, and the agent infers how likely it is that the state of the world is in some specific state. The agent thus believes to some degree—for each possible state—that it is in that state, but it is never certain exactly which state it is in. Furthermore, a POMDP is a *decision* process and thus facilitates making decisions as to which actions to take, given its previous observations and actions.

Formally, a POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, b_0 \rangle$ with the following seven components (see e.g., [5, 6]): (1) $\mathcal{S} = \{s_0, s_1, \ldots, s_n\}$ is a finite set of states of the world; the state at time t is denoted s^t; (2) $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a finite set of actions; (3) $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ is the *state-transition function*, giving for each world state and agent action, a probability distribution over world states; (4) $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the *reward function*, giving the immediate reward that the agent can gain for any world state and agent action; (5) $\Omega = \{o_0, o_1, \ldots, o_m\}$ is a finite set of observations the agent can experience of its world; (6) $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Pi(\Omega)$ is the *observation function*, giving for each agent action and the resulting world state, a probability distribution over observations; and (7) b_0 is the initial probability distribution over all world states in $\mathcal{S}$.

An important function is the function that updates the agent's belief: [5] call this function the *state estimation* function $SE(b, a, o)$. b is a set of pairs (s, p) where each state s is associated with a probability p, that is, b is a probability distribution over the set S of all states. b can be called a *belief state*. SE is defined as

$$b^t(s') = \frac{\mathcal{O}(s', a, o) \sum_{s \in S} T(s, a, s') b^{t-1}(s)}{Pr(o|a, b)}, \qquad (1)$$

where $b^t(s')$ is the probability of the agent being in state s' at time-step t. (Action and observation subscripts have been ignored.) Equation (1) is derived from the Bayes Rule. $Pr(o|a, b)$ in the denominator is a normalizer; it is constant with time. SE returns a new belief distribution for every action-observation pair. SE captures the Markov assumption: a new state of belief depends only on the immediately previous observation, action and state of belief.

3.2. Determining a policy

For any set of sequences of actions, the sequence of actions that results in the highest expected reward is preferred. The *optimality prescription* of utility theory states: Maximize "the expected sum of rewards that [an agent] gets on the next k steps," [5]. That is, an agent should maximize $E\left[\sum_{t=0}^{k-1} r_t\right]$ where r_t is the reward received on time-step t.

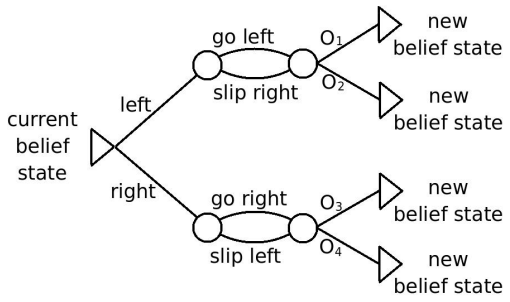


Figure 1: One tier of a POMDP-decision-tree.

When the states an agent can be in are belief states, we need a reward function over belief states. We derive $Rb(a, b)$ from the reward function over world states, such that a reward is proportional to the probability of being in a world state:

$$Rb(a, b) = \sum_{s \in S} R(a, s) \times b(s). \quad (2)$$

Now the aim of using POMDP models is to determine recommendations of 'good' actions or decisions. Such recommendations are called a *policy*. Formally, a policy (π) is a function from a set B of all belief states the agent can be in, to a set of actions: $\pi : B \rightarrow A$. That is, actions are *conditioned* on beliefs. So given b_0, the first action a' is recommended by π. But what is the next belief state? This depends on the next observation. Therefore, for each observation associated with a', we need to consider a different belief state. Hence, the next action, a'', actually depends on the observations associated with (immediately after) a'. In this sense, a policy can be represented as a *policy tree*, with nodes being actions and branches being observations. The above function is thus transformed to $\pi : \mathcal{O} \rightarrow A$. Now once we *have* a policy, it is independent of the agent's beliefs, except its initial belief.

Let $V_{\pi,t}(s)$—the *value function*—be the expected sum of rewards gained from starting in world state s and executing policy π for t steps. If we define a value function over *belief* states as $Vb_{\pi,t}(b) = \sum_{s \in S} V_{\pi,t}(s) \times b(s)$, we can define the *optimal* policy $\pi*$ with planning horizon h (set $t = h$) as

$$\pi* = argmax_{\pi}(Vb_{\pi,h}(b_0)) \quad (3)$$

(from the initial belief state)—the policy that will advise the agent to perform actions (given any defined observation) such that the agent gains maximum rewards (after h actions).

To implement Equation (3), the authors make use of a decision tree (there are other methods). DTGolog uses a similar approach: *forward search value iteration*. An example sub-decision-tree (one tier) is shown in Figure 1. This example is based on an environment and agent model where the agent can only go left or right and each of its two actions has two possible realizations in the environment; also, the agent may make two kinds of observations (O_1 and O_2) if it chose to go left, and another two kinds of observations (O_3 and O_4) if it chose to go right.

Belief states (triangles) in the decision tree are decision nodes, that is, at these nodes, the agent can choose an action (make a decision). Circles are chance nodes, that is, certain events occur, each with a probability (chance) such that any one event at one chance node will definely happen (probabilities of branches leaving a chance node, sum to 1).

In Decision Analysis (see e.g., [7]), we roll back a decision tree to 'decide' the action. In any decision tree, for each action-observation pair, there is a tier of sub-decision-trees. That is, when considering N actions in a row, a decision tree with N tiers would be required. There is a unique path from the initial decision node to each leaf node, and at each belief state encountered on a path, a reward is added, until (and including) the leaf belief state. At this point, the agent knows the total reward the agent would get for reaching that final state of belief. Each of the belief states is reachable with some probability.

At each decision node, a choice is committed to. We iteratively roll back—from last decision nodes to first decision node. The agent can in this way decide at the first decision node, what action to take. Each subtree rooted at the end of the branches representing the agent's potential action, has an associated expected reward. The action rooted at the subtree with the highest expected reward, should be chosen.

As the decision tree is rolled back, the best decision/action is placed into the policy, conditioned on the most recent possible observations. Using such a *policy tree* (generated from a decision tree), the agent can always choose the appropriate action given its last observation. This is the essence of the theory on which our POMDP planner is based.

4. Related work

In the following, we present some related work dealing with reasoning under uncertainty. As there exists a large body of work in this field, we concentrate in particular on approaches for reasoning under uncertainty in the situation calculus and Golog.

[8]'s idea of representing beliefs is simple yet important. Intuitively, their aim is to represent an agent's uncertainty by having a notion of which configuration of situations are currently possible; the possible worlds framework. Then further, each possible world is given a likelihood weight. With these notions in place, they show how an agent can have a belief (a probability) about any sentence in any defined situation. Their work does not, however, cover planning.

Reiter [3] describes how to implement MDPs as well as POMDPs in the situation calculus. He defines the language *stGolog*, which stands for 'stochastic Golog'. Nevertheless, Reiter does not provide a method to automatically generate (optimal) policies, given a domain and optimization theory; he only provides the tools for the designer to program by hand policies for partially observable decision domains.

Grosskreutz shows how the Golog framework "can be extended to allow the projection of high-level plans interacting with noisy low-level processes, based on a probabilistic characterization of the robot's beliefs," [9]. He calls his extension to Golog *pGolog*. The belief update of a robot's epistemic state is also covered by [9]. (PO)MDPs are not employed in pGolog. Instead, he does probabilistic projection of specific programs. He does however make use of expected utility to decide between which of two or three or so programs to execute (after simulated scenarios).

In [11], Ferrein and Lakemeyer present the agent programming language *ReadyLog*. Approximately ten years after Golog's birth, ReadyLog combines many of the disparate useful features of the various dialects of Golog into one package. ReadyLog has been implemented and successfully used in robotic soccer competitions and a prototype domestic robot.

Whereas DTGolog [2] models MDPs—a useful model in robotics, as most robots operate in environments where actions have uncertain outcomes—our new dialect models *belief-MDPs*. A belief-MDP is one perspective of POMDPs, where the states that are being reasoned over are *belief* states and not the *world* states of MDPs. More detail concerning the semantics of DTGolog is given in Section 2.

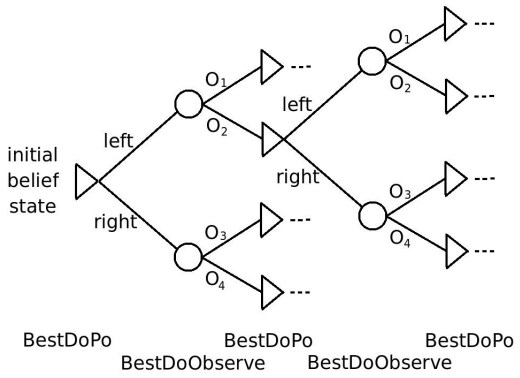


Figure 2: $BestDoPO$ represented as a POMDP-decision-tree.

Very related to our approach is the approach of [10]. Finzi and Lukasiewicz present a game-theoretic version of DTGolog to operate in partially observable domains. They call this extension *POGTGolog*. As far as we know, this is the only Golog dialect that can take partially observable problems as input, that is, that has some kind of POMDP solver for agent action planning. POGTGolog deals with multiple agents. Our work is different from theirs, as we concentrate on the single agent case and our agent is not restricted to game theory. For developers who prefer a Golog dialect for agent programming, but desire their robots or agents to operate with POMDP information, these developers cannot easily modify POGTGolog to work with single robots. Our work is not only a simplification of [10]; rather, we extend DTGolog, and use several elements in POGTGolog—either directly or for inspiration.

5. Semantics of POMDPs in Golog

In this section we describe our extension to the original forward search value iteration algorithm as proposed in [2]. In the following, we extend the approach of DTGolog in such a way that it can also deal with partially observable domains. In particular, instead of using $BestDo$, we introduce a predicate $BestDoPO$ to operate on a belief state rather than on a world state. $BestDoPO(p, b, h, \pi, v, pr)$ takes as arguments a Golog program p, a belief state b and a horizon h, which determines the solution depths of the algorithm. The policy π as well as its value v and the success probability pr are returned by the algorithm.

The relation of $BestDoPO$ to a POMDP-decision-tree can be seen in Figure 2. The stochastic outcomes of actions has been suppressed for ease of presentation.

An example of how $BestDoPO$ may be called initially—with a program that allows the agent to choose between three actions a_1, a_2, a_3 (without constraints), with b_0 the initial belief state and with the user or agent requiring advice for a sequence of seven actions—is $BestDoPO(\mathbf{while}\ true\ \mathbf{do}\ [a_1 \mid a_2 \mid a_3], b_0, 7, \pi, v, pr)$.

5.1. Basic definitions and concepts

A belief state b contains the elements (s, p); each element/pair is a possible (situation calculus) situation s together with probability p (as in [10]).

We use the idea of [10] and assume that an action is possible in a belief state, when it it possible in the situation which is part of the belief state, that is, $PossAct(a, b)$ iff $PossAct(a, s)$ (we rename the traditional $Poss$ to $PossAct$). We add the predicate $PossObs(o, a, s)$ to the action theory, which specifies when an observation o is possible (perceivable) in situation s, and define $PossObs(o, a, b)$ iff $PossObs(o, a, s)$, which defines when the observation is possible in belief state b, given an action a. The reader should clearly distinguish between preconditions for observations, $PossObs(o, a, s)$ and for actions, $PossAct(a, s)$. It is important to note that the b' in $PossObs(o, a, b')$ is the belief state reached *after* action a was executed. That is, if a was executed in b and b' is the new state reached, then $PossObs(o, a, b')$ says whether it is possible to observe *o after* a has been executed.

Next, we define a function symbol called $probNat(n, a, s)$ that is similar in meaning to the state transition function T of a Markov process. Our definition 'returns' a probability. It applies to all of nature's choices n, where s is the state in which the agent performs action a. Similarly, we introduce the function $probObs(o, a, s)$; the probability that o will be observed in s after a was executed in the previous situation.

Finally, we define $belObs$, which is the probability that the agent will observe some specified observation given its current beliefs and the sensor it activated: $belObs(o, a, b) \doteq \sum_{(s',p')\in b} p' \cdot probObs(o, a, s')$.

In the next section we briefly sketch our solution algorithm which calculates optimal policies under partial observability.

5.2. The partially observable $BestDo$

This subsection presents the key formulas in the definition of $BestDoPO$.

Considering possible observations after an action, we branch on all possible observations, given the robot's intended action a. $choiceObs'(a)$ 'returns' the set of observations that the robot may perceive: $\{o|choiceObs(o, a, s)$ for all $s \in S\}$. The reward function R is defined by (Eq. 2).

Probabilistic observation

$$
\begin{aligned}
&BestDoPo(a; rest, b, h, \pi, v, pr) \doteq \\
&\quad \neg PossAct(a, b) \wedge \pi = Stop \wedge v = 0 \wedge pr = 0 \ \vee \\
&\quad PossAct(a, b) \wedge \\
&\quad \exists \pi', v'. BestDoObserve(choiceObs'(a), \\
&\qquad a, rest, b, h, \pi', v', pr) \ \wedge \\
&\quad \pi = a; \pi' \wedge v = R(b) + v'.
\end{aligned}
$$

After a certain action a and a certain observation o_k, the next belief state is reached. At the time when the auxiliary procedure $BestDoObserve$ is called, a specific action, the set of nature's choices for that action and a specific observation associated with the action are under consideration. These elements are sufficient and necessary to update the agent's current beliefs. Inside $BestDoObserve$, the belief state (given a certain action and observation history) is updated via a belief state transition function (similar in vein to the state estimation function of Section 3, and the successor-state axiom for likelihood weights as given in [8]).

Belief update function

$$
\begin{aligned}
&b_{new} = BU(o, a, b) \doteq \\
&\quad \textbf{for each } (s, p) \in b \\
&\qquad \exists n, s^+, p^+. (s^+, p^+) \in b_{temp} : s^+ = do(n, s) \wedge \\
&\qquad choiceNat(n, a, s) \wedge PossAct(n, s) \wedge \\
&\qquad p^+ = p \cdot probObs(o, a, s^+) \cdot probNat(n, a, s) \\
&\quad \textbf{end for each} \\
&\quad b_{new} = normalize(b_{temp}).
\end{aligned}
$$

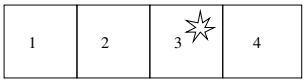


Figure 3: Four-state world; four states in a row. Initially the agent believes it is in each state with probabilities $[0.04|0.95|0.00|0.01]$ corresponding to state position.

A major difference between the POMDP model as defined in Section 3 and the POMDP model we define here for the situation calculus, is that here the belief state is not a probability distribution over a *fixed* set of states. If a situation (state) was part of the belief state to be updated, it is removed from the new belief state, and situations (states) that are 'accessible' for the removed situation via $choiceNat(n, a, s)$ *and* are executable via $PossAct(n, s)$, are added to the new belief state. Because non-executable actions result in situations being discarded, the 'probability' distribution over all the situations in the new belief state may not sum to 1; the distribution thus needs to be normalized.

$senseCond$ is mentioned in the definition of $BestDoObserve$: It is similar to the the definition in Section 2, only, here it is defined for observations instead of actions.

$BestDoPO$ is recursively called with the remaining program and with the horizon h decremented by 1. Also note that the recursive $BestDoPO$ will now operate with the updated belief b'. In the following definition, $\{o_k\}$ is a single (remaining) observation in the set returned by $choiceObs'$.

Observations possible

$$
\begin{aligned}
&BestDoObserve(\{o_k\}, a, rest, b, h, \pi, v, pr) \doteq \\
&\quad \neg PossObs(o_k, a, b) \wedge \pi = Stop \wedge v = 0 \wedge pr = 0 \vee \\
&\quad PossObs(o_k, a, b) \wedge b' = BU(o_k, a, b) \wedge \\
&\quad \exists \pi', v', pr'. BestDoPo(rest, b', h-1, \pi', v', pr') \wedge \\
&\quad senseCond(o_k, \varphi_k) \wedge \pi = \varphi_k?; \pi' \wedge \\
&\quad v = v' \cdot belObs(o_k, a, b) \wedge pr = pr' \cdot belObs(o_k, a, b).
\end{aligned}
$$

When the set of observations has more than one observation in it, the formula definition is slightly different, but similar to the one above: the first branch of possible observations is processed, and the other branches in the remainder of the set are processed recursively.

When the planning horizon has reached zero or when all actions have been 'performed' (no remaining actions in the input program), there will be no further recursive calls.

Conditional statement and test action formulas are similar to those of Golog, except that the 'condition' or 'test statement' respectively, are with respect to the agent's current belief state, and probabilities involved in these formulas are influenced in proportion to the agent's *degree of belief* [8] in the respective statements (see [10] for details). Sequential composition and conditional iteration are defined as one would expect according to complex actions in Golog.

6. A Simple Example

A very simple example follows to illustrate how $BestDoPO$ calculates an optimal policy. We use a four-state world as depicted in Figure 3. The agent's initial belief state is $b_0 = \{(s1, 0.04), (s2, 0.95), (s3, 0.0), (s4, 0.01)\}$. The only actions available to the agent are $left$ and $right$. We define the actions' stochasticity with $\forall n, a, s. choiceNat(n, a, s) \equiv TRUE$, with associated probabilities:

$$probNat(left, left, s) = probNat(right, right, s) = 0.9$$
$$probNat(right, left, s) = probNat(left, right, s) = 0.1$$

The probability that any of the actions will cause an observation of nothing ($obsnil$) is 1: $probObs(obsnil, a, s) = p \equiv (a = left \vee a = right) \wedge p = 1$. The corresponding definition for choice of observations is $choiceObs(obsnil, a, s) \equiv (a = left \vee a = right)$.

Let the fluent $At(loc(x), s)$ denote the location of the agent. It's successor-state axiom is defined by

$$
\begin{aligned}
&At(loc(x), do(a, s)) \equiv \\
&\quad a = left \wedge (At(loc(x+1), s) \wedge x \neq 1) \\
&\qquad \vee (At(loc(x), s) \wedge x = 1) \vee \\
&\quad a = right \wedge (At(loc(x-1), s) \wedge x \neq 4) \\
&\qquad \vee (At(loc(x), s) \wedge x = 4) \vee \\
&\quad At(loc(x), s) \wedge (a \neq left \wedge a \neq right).
\end{aligned}
$$

For simplicity, we allow all actions and observations all of the time, that is, $\forall a, s. PossAct(a, s) \equiv TRUE$ and $\forall a, s. PossObs(a, s) \equiv TRUE$.

Finally, we specify the sensing condition predicate and the reward function. $senseCond(obsnil, \psi) \equiv \psi = OutcomeIs(nil, sensor\_value))$ with $OutcomeIs(obsnil, sensor\_value) \equiv TRUE$ and $reward(s) =$ **if** $At(loc(3), s)$ **then** 1 **else** -1; hence, the agent's goal should be location 3.

Assume, the agent is equipped with the following program; an initial input for $BestDoPO$:

$$
\begin{aligned}
&BestDoPO(\textbf{while}\ (true\ \textbf{do}\ [left \mid right]), \\
&\quad \{(s1, 0.04), (s2, 0.95), (s3, 0.0), (s4, 0.01)\}, 1, \pi, v, pr);
\end{aligned}
$$

the algorithm must computing a one-step optimal policy.

After the iterative component of the program is processed, the following call is made, as per the definition of $BestDoPO$ for the nondeterministic choice of actions:

$$
\begin{aligned}
&BestDoPO([left \mid right]; rest, b_0, 1, \pi, v, pr) \\
&\quad \exists \pi_1, v_1, pr_1. BestDoPO(left; rest, b_0, 1, \pi_1, v_1, pr_1) \wedge \\
&\quad \exists \pi_2, v_2, pr_2. BestDoPO(right; rest, b_0, 1, \pi_2, v_2, pr_2) \wedge \\
&\quad ((v_1, left) \geq (v_2, right) \wedge \pi = \pi_1 \wedge v = v_1 \wedge pr = pr_1) \vee \\
&\quad (v_1, left) < (v_2, right) \wedge \pi = \pi_2 \wedge v = v_2 \wedge pr = pr_2)),
\end{aligned}
$$

where $rest$ is **while** $(true\ \textbf{do}\ [left \mid right])$. Then the recursive $BestDoPO$s make use of the "Probabilistic observation" definition of the formula. Because–by the action precondition axioms for this example–$left$ and $right$ are always executable, the following portion (times two) of the formula are applicable:

$$\exists \pi', v'. BestDoObserve(choiceObs'(left), \tag{4}$$
$$left, rest, b_0, 1, \pi', v', pr) \wedge \tag{5}$$
$$\pi = left; \pi' \wedge v = R(b_0) + v' \tag{6}$$

and
$$\exists \pi', v'. BestDoObserve(choiceObs'(right), \tag{7}$$
$$right, rest, b_0, 1, \pi', v', pr) \wedge \tag{8}$$
$$\pi = right; \pi' \wedge v = R(b_0) + v'. \tag{9}$$

For Lines (4) and (5) the following portion of the "Observations possible" definition is applicable:

$$b' = BU(obsnil, left, b_0) \wedge$$
$$\exists \pi', v', pr'.BestDoPO(rest, b', 1-1, \pi', v', pr') \wedge$$
$$senseCond(obsnil, \phi) \wedge \pi = \phi?; \pi' \wedge$$
$$v = v' \cdot belObs(obsnil, left, b_0) \wedge$$
$$pr = pr' \cdot belObs(obsnil, left, b_0).$$

In this formula (portion), ϕ unifies with $OutcomeIs(obsnil, sensor\_value)$ and because the recursive call to $BestDoPO$ has a zero horizon, $\pi' = nil$, and thus $\pi = (OutcomeIs(obsnil, sensor\_value))?; nil$.

The updated belief is an input to a 'zero horizon' call and will therefore be used to determine v'; we calculate the new belief state $b' = BU(obsnil, left, b_0)$ now (we work out only the first new element of b' in detail):

$$(s^+, p^+) \in b_{temp} : s^+ = do(left, s_1) \wedge\ p^+ = 0.04 \times 1 \times 0.9.$$

Because all actions are possible, the only effect that normalization (in the update function) has, is to remove $(do(left, s3), 0.0)$ and $(do(right, s3), 0.0)$ from the new belief state, because of their zero probabilities. $BU(obsnil, left, b_0)$ results in

$$b' = \{(do(left, s1), 0.036), (do(right, s1), 0.004),$$
$$(do(left, s2), 0.855), (do(right, s2), 0.095),$$
$$(do(left, s4), 0.009), (do(right, s4), 0.001)\}.$$

$belObs(obsnil, left, b_0) = (0.04)(1)+(0.95)(1)+(0.0)(1)+(0.01)(1) = 1$ and hence $v = v' \times 1$, and $pr = pr' \times 1$. Due to the 'zero horizon' call, $v' = R(b') = (-1)(.036) + (-1)(.004) + (-1)(.885) + (1)(.095) + (1)(.009) + (-1)(.001) = -0.822$ and $pr' = 1.0$. Therefore, $v = -0.822$, and $pr = 1.0$.

Now we can instantiate Line (6) as follows: $\pi = left; OutcomeIs(obsnil, sensor\_value))?; nil \wedge v = -1 + (-0.822)$. Similarly, we can instantiate Line (9) as $\pi = right; OutcomeIs(obsnil, sensor\_value))?; nil \wedge v = -1 + (0.712)$.

Then finally, we find that $((-0.822, left) < (0.712, right)$ and return the policy $\pi = right; OutcomeIs(obsnil, sensor\_value))?; nil$, with total expected reward $v = -0.288$ and program success probability $pr = 1$.

Note that for the sake of clarity, we assumed noise-free perceptions. It should be clear though, that our algorithm can deal with noisy perceptions as well.

Considering that the agent believed to a relatively high degree that it was initially just left of the 'high-reward' location, and given that its observations are complete and its actions are not extremely erroneous, we would expect the agent's first move to be rightwards, as indeed, the policy recommends.

7. Discussion and Conclusion

In this paper we have given a formal semantics for an action planner that can generate control policies for agents in partially observable domains. The language we used for the specification is the agent programming language DTGolog. Much of the semantics is similar to [10]. Their approach is however not for a single-agent domain.

An example was presented that showed in detail the processes involved in generating a policy for an agent with probabilistic beliefs in a partially observable and stochastic domain.

We implemented the POMDP planner in ECL^iPS^e *Prolog*. The implementation was set up for two toy worlds: a four-state world where the states are all in a row, and a five-by-five grid world. In both cases, an agent must find a 'star'. Preliminary experiments with the implementation showed the potential for practical application of the planner presented in this paper: the results of the experiments showed that the policies generated are reasonable, and overall, the planner seems to work correctly. However, benchmarking and comparison to other similar planners (for problems in similarly stochastic and noisy domains) still needs to be conducted.

8. References

[1] Levesque, H., Reiter, R., Lespérance, Y., Lin, F., and Scherl, R., "GOLOG: A Logic programming language for dynamic domain", *Journal of Logic Programming*, 31:59–84, 1997.

[2] Boutilier, C., Reiter, R., Soutchanski, M., and Thrun, S., "Decision-theoretic, high-level agent programming in the situation calculus", in *Proceedings AAAI-2000*, 2000, pp. 355–362.

[3] Reiter, R., *Knowledge in action: logical foundations for specifying and implementing dynamical systems*, Massachusetts/England: MIT Press, 2001.

[4] Brachman, R. J. and Levesque, H. J., *Knowledge representation and reasoning*, California: Morgan Kaufmann, 2004.

[5] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R., "Planning and acting in partially observable stochastic domains", *Artificial Intelligence*, 101(1–2):99–134, 1998.

[6] Pineau, J., *Tractable planning under uncertainty: exploiting structure*, Robotics Institute, Carnegie Mellon University, 2004. Unpublished doctoral dissertation.

[7] Clemen, R. T., and Reilly, T., *Making hard decisions*, California: Duxbury, 2001.

[8] Bacchus, F., Halpern, J. Y., and Levesque, H. J., "Reasoning about noisy sensors and effectors in the situation calculus", *Artificial Intelligence*, 111(1–2):171–208, 1999.

[9] Grosskreutz, H., *Towards more realistic logic-based robot controllers in the Golog framework*, Knowledge-Based Systems Group, Rheinisch-Westfälischen Technischen Hochschule, 2002.

[10] Finzi, A, and Lukasiewicz, T., "Game-theoretic agent programming in Golog under partial observability", in *KI 2006: Advances in Artificial Intelligence*, 2007, pp. 113–127.

[11] Ferrein, A. and Lakemeyer G. "Logic-based robot control in highly dynamic domains.", *Journal of Robotics and Autonomous Systems, Special Issue on Semantic Knowledge in Robotics* 2008. to appear.

[12] Bonet, B. and Geffner, H., "Planning and control in artificial intelligence: a unifying perspective", *Applied Intelligence*, 14(3): 237–252, 2001.

[13] Poole, D., "Planning and acting in partially observable stochastic domains", *Linköping Electronic Articles in Computer and Information Science*, 3(8), 1998.

Acoustic cues identifying phonetic transitions for speech segmentation

D.R. van Niekerk and E. Barnard

Human Language Technologies Research Group, Meraka Institute, CSIR, Pretoria /
School of Electrical, Electronic and Computer Engineering,
North-West University, Potchefstroom, South Africa
dvniekerk@csir.co.za, ebarnard@csir.co.za

Abstract

The quality of corpus-based text-to-speech (TTS) systems depends strongly on the consistency of boundary placements during phonetic alignments. Expert human transcribers use visually represented acoustic cues in order to consistently place boundaries at phonetic transitions according to a set of conventions. We present some features commonly (and informally) used as aid when performing manual segmentation and investigate the feasibility of automatically extracting and utilising these features to identify phonetic transitions. We show that a number of features can be used to reliably detect various classes of phonetic transitions.

1. Introduction

Defining exact boundaries between phonetic segments in speech is difficult, especially in those contexts where co-articulation between neighbouring phones renders boundary definition somewhat ambiguous. Nevertheless, for the purposes of spoken language research and system development, a pragmatic approach is necessary in order to define such boundaries as accurately and consistently as possible. Research into the development of corpus-based text-to-speech systems has suggested that consistency (in addition to accuracy) of boundary placements is an important factor when considering the eventual quality of these systems [1, 2].

Most early development of speech corpora involved manual effort by language or phonetics experts with a significant amount of experience in identifying phonetic segments from visual and auditory information. This reliance on expert human involvement has endured, despite advances in speech recognition and machine learning techniques applied to automating this task. As much is evident when one considers that high quality corpora are still manually checked by such individuals [3].

The expert manual transcription procedure can be viewed as a two-stage process, where the transcriber initially identifies segments based on the acoustic properties (aided by visual representations thereof) and subsequently refines boundary placements between contiguous segments by considering sets of consistent acoustic cues based on the transition context (defined by broad phonetic classes).

The application of Hidden Markov Models (HMMs) to phonetic segmentation can be likened to the first stage of the expert procedure described above and in cases where such models are sufficiently trained, this leads to boundary placements which for the most part are fairly similar to the "ideal" locations [4]. This is especially true when manually segmented data exists with which to bootstrap the process involved in training HMMs.

Nevertheless, a large amount of research has been done on further reducing the discrepancies between HMM based and manually obtained boundaries (i.e. "boundary refinement") [5, 6, 7]. This has been justified by the observation that manually segmented and refined automated methods usually result in better quality synthesis when compared to baseline methods [8, 9].

The implementation of the boundary refinement stage has largely involved the application of statistical machine learning techniques relying on samples of manually segmented data in order to "learn" the conventions of expert transcribers without explicitly considering the underlying process or considerations taken into account. This has proved successful, with researchers reaching levels of accuracy rivalling what can be expected when compared to discrepancies between independently verified alignments by experts [5].

Unfortunately, the feasibility of applying techniques such as these is limited in the context of developing corpora toward building systems for languages where resources and expertise are scarce. This is the case for two primary reasons:

- Corpora are designed minimally in order to minimise effort in text selection (it is difficult to find reliable electronic texts for these languages) and expertise required during recording and annotation. This results in corpora where some phonetic contexts simply do not have sufficient observations in order to train adequate acoustic models.
- No manually checked corpora pre-exist in most of the languages of the developing world, because of a lack of skilled persons to perform such tasks. Corpora which are hand checked are small and have mostly been produced by persons with limited background and training.

For the purposes of developing relevantly annotated corpora with the goal of building high quality spoken language systems, it is thus worthwhile investigating the automated extraction and application of acoustic cues to identify phonetic transitions in much the same way as a human transcriber would. To this end we identify important features and the feasibility of extracting phonetic events from such features. The identification of reliable acoustic cues would have the following advantages for automated corpus development:

- Boundary candidates obtained in this way can serve as an independent point of reference for judging the quality of alignments (whether automatically or manually obtained).
- These boundary candidates can be integrated into an automated procedure in order to refine boundary placements or improve the quality of training acoustic mod-

els, taking into consideration a specific protocol with the end goal of the segmented corpus in mind.

In this paper we present an initial analysis of the effectiveness of various cues for detecting phonetic events in different contexts in order to determine the feasibility and potential impact of applying this information. Section 2 describes the identification of potential features, Section 3 describes the experimental setup including the details of identifying boundary candidates. Finally, we report on the results obtained (Section 4) and conclude with a discussion in Section 5.

2. Acoustic features

In large resource collection efforts the development of annotated corpora has typically been realised by the collaboration of a large number of trained individuals. The collaboration of multiple individuals is essential in order to complete the sizable task of manually verifying the quality of phonetic alignments within acceptable time-frames, and to have reliable methods of quality assurance.

Due to the ambiguities which exist at phonetic transitions, it is common to define protocols for the placement of phonetic boundaries based on broad phonetic class categories in order to ensure the consistency of the end result across different individuals [10, 3].

Typical protocols incorporate practical guidelines for the identification of phonetic boundaries based on acoustic cues exhibited by various features that can be extracted or calculated and displayed. This includes the signal energy, estimated fundamental frequency, periodicity (voicing), extracted formant contours, spectral characteristics and waveform shape. Instructions on boundary placement range from complex and highly conditional (e.g. when transcribing approximants, some suggest observing the formants, F3 and F4 for "energy reduction") to relatively simple and clearly defined (e.g. place a phonetic boundary "just prior to the burst of energy" when transcribing a stop consonant). Considering this and initial experiments on how reliably one can estimate or extract all of these features, we have concentrated on the following features for the automatic identification of segmentation cues:

- Signal intensity,
- Fundamental frequency (f0),
- Signal envelope, and
- Cepstral distances.

Due to difficulties in reliably determining the number of formants present as well as the exact contours, we have chosen to rely on the use of a "cepstral distance" measure (defined in Section 3.3.5) which we hope will identify changes in the formants and general spectral changes with sufficient accuracy.

3. Experimental setup

We employed the *Praat* [11] and *HTK* [12] software packages to aid in extracting features from three sets of manually annotated audio recordings representing typical minimally designed TTS corpora (see Table 1).

Language	Gender	Utterances	Duration	Phones
Afrikaans	Male	134	21 mins.	12341
isiZulu	Male	150	19 mins.	8559
Setswana	Female	332	44 mins.	26010

Table 1: Reference data sets

3.1. Broad phonetic classes

The most practical and relevant view of phonetic transition contexts for this study is based on broad phonetic categories. All segment labels in the above-mentioned corpora are thus mapped to one of the following labels in accordance with International Phonetic Alphabet (IPA) definitions: *affricate*, *approximant*, *click*, *fricative*, *nasal*, *pause*, *stop*, *trill* and *vowel*.

The *pause* label is used both with reference to long pauses (typically only occurring at the beginning and end of utterances) and short segments associated with little signal energy such as glottal stops and closures.

3.2. Generating boundary candidates

In general, boundary candidates are established by firstly calculating or estimating contours for the particular feature and either using this contour directly where applicable or deriving a subsequent contour representing the slope by means of numerical differentiation. After obtaining the appropriate representation, we employ a simple peak detection algorithm in order to generate boundary candidates at specific time instants. We briefly present these methods below.

3.2.1. Numerical differentiation

In order to obtain a relatively smooth contour suitable for subsequent peak detection to be effective, we firstly calculate the difference between each sample of the original contour x to obtain a new sequence of differences x_d defined for time instants in-between the original time instants. An odd number N of "difference samples" are framed resulting in a frame x_{df} for each time instant. From this the gradient is determined by first windowing the frame with a simple exponential window function:

$$w[n] = 2^{-|n-\frac{N-1}{2}|}, \tag{1}$$

obtaining a frame with weighted differences x_{dfw}:

$$x_{dfw}[n] = x_{df}[n]w[n], \tag{2}$$

and calculating the slope at t (the time instant at the center of the frame) by averaging the weighted differences in each frame:

$$x'[t] = \frac{1}{N}\sum_{n=1}^{N} x_{dfw}[n]. \tag{3}$$

3.2.2. Peak detection

For detecting local extrema that are of interest during candidate identification, we frame the relevant contour, obtaining an odd number of samples that constitute each frame and simply flag the time instant of the central sample within the frame if it is a global extremum within the frame.

3.3. Acoustic cues

Taking into account the observations in Section 2 we experimented with extracting features and identifying candidates automatically. We now briefly describe the particular cues investigated.

3.3.1. Intensity dynamics

It was observed that many phonetic transitions coincide with changes in the signal intensity and initial experiments indicated

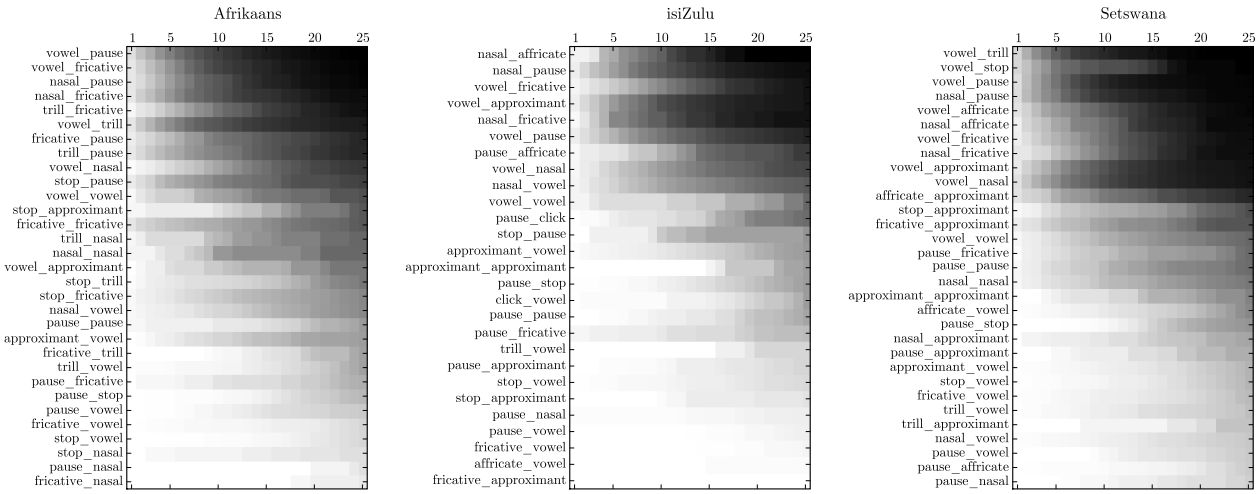


Figure 1: Detection rates: for each phonetic transition context we obtain detection rates for a range of time thresholds (in milliseconds), darker areas represent higher detection rates; this figure represents rates when using the intensity gradient minima cue for each of the languages.

that the slope of the intensity contour peaked near potential boundaries. We thus determine intensity values at 5ms intervals and subsequently obtain the derivative and flag the local minima and maxima of the resulting contour (we distinguish between candidates at minima and maxima).

3.3.2. Waveform envelope

Between neighbouring voiced regions such as vowels and nasals, "dips" in the waveform can indicate a phonetic transition. By obtaining the waveform envelope and flagging local minima, such events can be detected. The use of the intensity contour directly was considered, but in cases such as just mentioned, the envelope provides a more pronounced cue.

3.3.3. Voicing

By means of a pitch analysis in the frequency range 75Hz to 600Hz, one obtains regions that have a strong periodic component which can be identified as voiced regions. By distinguishing between periodic and aperiodic regions one can place boundary candidates between neighbouring regions in the hope of detecting transitions between voiced and unvoiced segments.

3.3.4. Fundamental frequency dynamics

It has been noted that there exists structure within the f0 contour which can be used to identify phonemic events [9]. We attempt to detect these events by employing the *Praat* pitch detection algorithm [13] in the 75Hz to 600Hz range and analysing the slope of the resulting contour.

3.3.5. Cepstral distance

As a measure of spectral difference, which is often used directly via observing the spectrogram or more specifically the changes in formants in order to identify boundary locations manually, we calculated 12 mel frequency cepstral coefficients in 20ms windows with a 2ms time shift. Using this observation sequence we consider windows of N observations, calculate the average of the first $N-1$ observations and simply calculate the euclidian distance between the last observation and the average calculated in order to obtain a contour representing a measure of difference between each observation and the prior $N-1$ observations. This contour exhibits peaks at points where the spectral properties change radically.

3.4. Evaluation metric

Because boundary candidates will not coincide exactly with reference boundary locations, we consider a reference boundary location to be *detected* when a candidate boundary is located within a certain time threshold of the reference (following a strategy similarly defined in [14]). Subsequently we define an *unambiguous detection* where only detections with at most one candidate within the defined window around the reference are considered. This discredits detections where false alarms are present. For a specific phonetic transition context we can thus define the *unambiguous detection rate* as the ratio between the number of unambiguous detections and the number of occurrences for each context.

4. Results

By analysing the detection rates for various cues and phonetic contexts over a range of time thresholds, it is possible to obtain a detailed picture of the success of each cue based on phonetic context (see Figure 1 for an example). To investigate the detection rates for individual phonetic contexts, we have to evaluate a range of time thresholds instead of one common threshold (such as 20ms, which is often used), because of the relative durations of phones (e.g. stop phones often have average durations of less than 20ms).

In the subsequent sections we present quantitative results obtained when applying the techniques described on the corpora mentioned in Section 3 (see Table 1).

4.1. Transition detection: coverage

To measure the utility of each cue, the number of detections as a percentage of the total number of transitions is determined. This is done by firstly distinguishing contexts which are deemed successfully detected in general (it was decided that any transition context with detection rates in excess of 70% would be considered), after which detections are summed for these contexts. The results of this process are presented in Table 2.

Cue	Afrikaans	isiZulu	Setswana
Intensity gradient maxima	39.8%	49.1%	38.1%
Intensity gradient minima	36.4%	28.9%	37.4%
Cepstral difference	32.3%	53.5%	35.2%
Waveform envelope minima	36.9%	33.0%	52.8%
Voicing	4.4%	5.8%	37.5%
F0 gradient extrema	3.6%	10.0%	17.9%

Table 2: Cue significance: the percentages reflect the fraction of all phonetic transitions which are successfully detected by each of the listed cues; only transition contexts for which at least 70% detection is achieved are included in these counts.

By using the same notion of successfully detected context, it is also interesting to note the combined transition coverage by the complete set of cues. Figure 2 shows the cumulative coverage when the total occurrences for successfully detected phonetic contexts by each cue are added in turn.

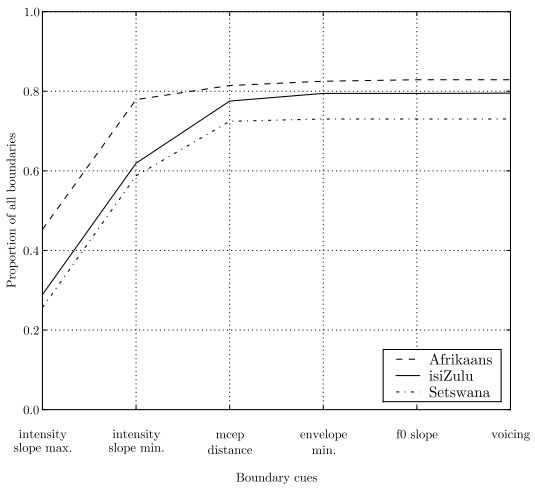


Figure 2: Coverage: the graphs represent the fraction of all phonetic transitions when the number of occurences of successfully detected transition contexts are accumulated for each language.

4.2. Problematic contexts

By differencing the set of contexts that are successfully detected with the complete set, the set of contexts which are least successfully detected is obtained (listed in Table 3). The sets obtained are not surprising considering most of the contexts listed are generally found to be relatively ambiguous (e.g. approximant-vowel transitions) and difficult to distinguish even by manual transcribers. Some of the contexts listed here are also relatively short in duration which suggests that the candidate generation methods used might not be well suited to these conditions.

Afrikaans:
stop-fricative, stop-trill, stop-pause, vowel-nasal, trill-approximant, fricative-pause, approximant-vowel, trill-stop, nasal-nasal, vowel-approximant, fricative-fricative
isiZulu:
pause-affricate, stop-approximant, approximant-pause, affricate-pause, approximant-vowel, stop-vowel, vowel-vowel
Setswana:
pause-affricate, stop-approximant, trill-pause, trill-approximant, approximant-pause, nasal-trill, trill-trill, stop-stop, affricate-pause, approximant-vowel, affricate-affricate, stop-vowel, fricative-nasal, vowel-vowel, pause-trill, fricative-fricative

Table 3: Problematic transition contexts: the contexts listed here were not successfully detected by any of the cues investigated.

5. Conclusion

In this paper we demonstrated the possibility of generating phonetic boundary candidates based on specific acoustic cues that were extracted for three different languages. We showed that it is possible to detect actual boundary positions to a large degree (especially in contexts where the specific cue is relevant from the perspective of speech production).

Although each cue had specific contexts where it outperformed others, the most significant cues were based on the intensity contour and cepstral distance. The fundamental frequency proved to be less successful than expected (based on [9]), but this can probably be attributed to the nature of the reference TTS corpora where the tone is kept more constant than in purely natural speech. Another interesting observation is that the voicing cue worked reasonably well for the female voice but poorly for the male voices, based on these results one should probably carefully consider the exact pitch range of the specific voice before attempting to use this cue.

The problematic contexts remaining seem to be either acoustically ambiguous (e.g. approximant-vowel boundaries cannot be easily distinguished by spectral properties or by observing the waveform) or present cases where our method of candidate generation fails. Segments with very short durations can cause the peak detection method or averaging process set up for the average case to miss detections and particularly the cepstral distance measure proposed would also be more effective for longer segments. Future work in detecting the remaining transitions might involve more sophisticated candidate generation or the application of more appropriate features (formant contours might prove successful).

The identification of boundary candidates presented here will allow us to improve the quality of the alignment process automatically. This can be done by defining a protocol similar to protocols designed to allow consistency between multiple human transcribers and using this directly or integrating candidates into training procedures in order to refine models with

respect to precise boundary placements. Another useful application would be to flag potentially misaligned boundaries during quality control of manually or automatically segmented corpora.

An important observation is that boundary refinement based on these candidates can be done automatically and with the target use in mind. This presents opportunity for further research questions relating to text-to-speech synthesis quality when relying on certain acoustic cues to define boundaries. Important acoustic properties relating to speech parametrisation used for speech synthesis should also be explored, e.g. when employing the Harmonics Plus Noise Model, the maximum voiced frequency contour might prove relevant when performing segmentation.

6. References

[1] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[2] M. Makashay, C. Wightman, A. Syrdal, and A. Conkie, "Perceptual evaluation of automatic segmentation in text-to-speech synthesis," in *Proceedings of ICSLP*, Beijing, China, October 2000, vol. 2, pp. 431–434.

[3] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The Buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability," *Speech Communication*, vol. 45, no. 1, pp. 89–95, 2005.

[4] D.R. van Niekerk and E. Barnard, "Important factors in HMM-based phonetic segmentation," in *Proceedings of PRASA*, Pietermaritzburg, South Africa, November 2007, pp. 25–30.

[5] D.T. Toledano, L.A. Hernández Gómez, and L.V. Grande, "Automatic Phonetic Segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, 2003.

[6] A. Sethy and S. Narayanan, "Refined Speech Segmentation for Concatenative Speech Synthesis," in *Proceedings of ICSLP*, Denver, Colorado, USA, September 2002, pp. 149–152.

[7] Y. Kim and A. Conkie, "Automatic Segmentation Combining an HMM-Based Approach and Spectral Boundary Correction," in *Proceedings of ICSLP*, Denver, Colorado, USA, September 2002, pp. 145–148.

[8] J. Adell, A. Bonafonte, L.A. Hernndez Gmez, and M. J. Castro, "Comparative study of automatic phone segmentation methods for tts," in *Proceedings of ICASSP*, Philadelphia, Pennsylvania, USA, Mar. 2005, vol. 1, pp. 309–312.

[9] T. Saito, "On the use of F0 features in automatic segmentation for speech synthesis," in *Proceedings of ICSLP*, Sydney, Australia, December 1998, vol. 7, pp. 2839–2842.

[10] R. Cole, M. Noel, and V. Noel, "The CSLU Speaker Recognition Corpus," in *Proceedings of ICSLP*, Sydney, Australia, December 1998, pp. 3167–3170.

[11] P. Boersma, *Praat, a system for doing phonetics by computer*, Amsterdam: Glott International, 2001.

[12] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Veltchev, and P. Woodland, *The HTK Book (for HTK Version 3.3)*, Cambridge University Engineering Department, http://htk.eng.cam.ac.uk/, 2005.

[13] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the Insitute of Phonetic Sciences, University of Amsterdam*, Amsterdam, The Netherlands, 1993, vol. 17, pp. 97–110.

[14] Y. P. Estevan, V. Wan, and O. Scharenborg, "Finding maximum margin segments in speech," in *Proceedings of ICASSP*, Honolulu, Hawai'i, USA, April 2007, vol. 4, pp. 937–940.

Photometric modelling of real-world objects

John Morkel and Fred Nicolls

Department of Electrical Engineering
University of Cape Town, South Africa
{jmorkel, nicolls}@dip.ee.uct.ac.za

Abstract

This paper presents a method to model the photometric properties of real-world objects from single-view and calibrated multi-view image sets. Lights are modelled as point sources and reflection properties are modelled using the isotropic Ward reflectance model. Lighting and reflectance are simultaneously recovered using known geometry. Measured reflectance data and model results are presented along with rendered scenes generated using the photometric models. The rendered images compare closely to the original images with the colours, positions of shadows and highlights accurately reproduced.

1. Introduction

In computer graphics, an image of an object is rendered by calculating how light sources interact with objects, given the shape, the position of the lights, the reflectivity of the objects and the viewpoint of the observer. Rendering, or more specifically *forward rendering*, is widely used to create special effects and animation in television shows and films and is perhaps a more familiar concept than inverse rendering. Forward rendering involves calculating the appearance of an object when geometry, reflectance properties and lighting are known.

Inverse rendering is the logical opposite of forward rendering. It is the process of decomposing a scene with unknown geometry, reflectivity and lighting into its constituent elements such that the same scene could be synthetically recreated through forward rendering. Recovering geometry, reflectance and lighting through inverse rendering when all three properties are unknown is theoretically an ill-posed problem [15], since in order to recover the lighting distribution for a scene, geometry and reflectance data for the objects in the scene are required. Similarly, to measure the reflectance data, geometry and the lighting distribution are required.

In this paper, geometry is known and is represented by a triangular mesh model consisting of vertices defined in world coordinates and a connectivity matrix. Reflectivity is represented by a *bidirectional reflectance distribution function* (BRDF) model [4, p. 61], which describes how the object's material absorbs and reflects light as the incoming light angle and viewpoint vary. The lighting distribution is modelled by an ambient light intensity and point light sources defined by a position in world coordinates and intensity.

The appearance of an object in a scene can be modelled by a geometric and a photometric model. The geometric model describes the position of the object in space and the orientations of the facets that comprise the model. The geometry of objects is assumed to be known in this paper. Geometry data is obtained using a 3D scanner. The photometric model consists of two models: a lighting model and a reflection model. The lighting model describes the light distribution of the scene in which the object is imaged and the reflection model describes how the light interacts with the surface of the object. Given these models, the object can be rendered from a novel viewpoint and under novel lighting.

This paper is organised as follows: Section 2 briefly discusses work related to this paper; Section 3 describes the process of fitting lighting and reflectance models to the data; Section 4 details the equipment, methods and calibration involved in the data acquisition process; Section 5 explains the preprocessing steps performed on the data, including aligning the 3D data with the image data, that are necessary to ensure that the different data are consistent with each other; Section 6 contains results of the reflectance and lighting measurements for the rock data. Sections 7 and 8 contain concluding remarks and possible future work.

2. Related work

A device known as a *gonioreflectometer* is traditionally used to measure the reflectance properties of an object. Recent research has led to image-based methods [11, 12, 18] that measure reflectance properties of an object without the need for specialised equipment. BRDF measurements are inherently noisy [19] and a complete estimated BRDF for an object requires many data points. BRDF models are either empirical or physics based. Model parameters are optimised so that predictions closely match BRDF measurements. BRDF models are convenient for applications in computer vision and computer graphics because an entire BRDF data set can be substituted by a few parameters. Noise is also averaged out when fitting a restricted model to the data.

Physics based models, such as the Torrance and Sparrow [17] and He-Torrance [8] models, are preferred in some literature [5, 16], but implementation is complicated because of dependence on wavelength. Low dimension models, such as the isotropic Ward model [19], are simpler to implement and provide adequate accuracy [6, 18]. Other models by Phong [14] and Lafortune et al. [9] are also widely used [1, 11, 12, 20].

3. Photometric modelling

To model the photometric properties of an object, a model of the light sources in a scene is required. This model provides

information about the intensity of the light and incidence angles for light that arrives at the object's surface. Only once a model for the scene lighting distribution is available can the reflectance properties of the object be measured.

3.1. Modelling light sources

Most reflection models describe the two fundamental types of reflection, namely diffuse and specular reflection. A simple assumption to make is that the objects in the scene exhibit only one type of reflection and not a combination, as is usually the case. The reflection is hence modelled by a single-parameter BRDF model which describes the albedo or ratio of incident to reflected light. This parameter cannot be calculated without a lighting distribution so it is assumed arbitrarily to be unity, thus the object is assumed to exhibit either perfect specular or perfect diffuse reflection.

The reflectance model used as an initial estimate should be chosen based on the typical reflectance properties of the objects to be modelled. Most objects exhibit Lambertian (or diffuse) reflection, with some exhibiting both diffuse and specular reflection. The exception is highly reflective surfaces such as mirrors, which are mainly specularly reflective. As such, the Lambertian assumption is used to infer an initial estimate for the lighting distribution.

For an object with Lambertian reflection, the intensity of a point p for a given light distribution is given by

$$I(p) = L_0 + \sum_{i=1}^{n} \mathbf{N}_p \cdot (\mathbf{V}_i - \mathbf{P}_p)\Gamma(p,i)\frac{1}{||\mathbf{V}_i - \mathbf{P}_p||^2}L_i, \quad (1)$$

where L_0 is the ambient light term that accounts for ambient light and inter-reflections, n is the number of point light sources, $\Gamma(p,i)$ is a function that is 1 when point p is visible to light source i and 0 otherwise, $\mathbf{N}_p$ is the normal vector at point p, $\mathbf{V}_i$ is the position vector for light source i, $\mathbf{P}_p$ is the position vector for the point p, and L_i is the intensity for light source i.

The vector from the point p on the surface to the light source i is $\mathbf{V}_i - \mathbf{P}_p$. The $\frac{1}{||\mathbf{V}_i - \mathbf{P}_p||^2}$ factor is the falloff in intensity that occurs as a result of the light energy being distributed over an increasingly larger area as distance from the light source increases. The falloff is inversely proportional to the square of the distance from the light source.

Light sources are modelled as points with a position in 3D space and an intensity for each of the RGB channels. There is an additional ambient light term that accounts for any background lighting that is present in the room, as well as any inter-reflection that might occur. The number of light sources is a user-defined input. Face orientation is extracted from the geometric model of the object. Selecting the correct number of light sources for the model is not critical because light sources will converge to the same point in space if there are more light sources in the model than in reality.

The position and intensity for each light source is optimised using the MATLAB non-linear optimisation routine *lsqnonlin*. The cost function employed is defined as the squared error between the intensity obtained using the rendering equation (Equation 1) and the observed intensity from image data. The point light sources are initially optimised without an ambient light term. Once good estimates for the point light sources are available, their positions and intensities are optimised in a second step along with the ambient light intensity. This lighting distribution is used as an initial estimate and is further refined when reflectance and lighting parameters are simultaneously estimated.

3.2. Modelling reflectance

The isotropic Ward model [19], used for modelling reflectivity in this research, is defined as

$$\beta(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{\rho_d}{\pi} + \frac{\rho_s \exp\left(-\tan^2(\delta)/\alpha^2\right)}{4\pi\alpha^2}, \quad (2)$$

where ρ_d is the diffuse reflectance, ρ_s is the specular reflectance, α is the standard deviation of the surface slope, and δ is the angle between the half vector, $\hat{h}$, and the surface normal, $\hat{n}$. It offers a good compromise between complexity and accuracy. The Ward model does not explicitly depend on wavelength, but diffuse and specular reflectivity parameters can be calculated for each RGB channel so that colour can be modelled.

Calculating the ρ_d, ρ_s and α parameters is done by regression in an optimisation framework using the MATLAB *lsqnonlin* nonlinear optimiser. Each pixel in every image of an object provides a data point that is used to calculate optimal BRDF model parameters to fit the the observed BRDF data. Angle information is deduced from the mesh model of the object, the camera positions, and the light source positions. The viewing ray is obtained by backprojecting each pixel onto the 3D model of the object. The cost used in optimising is the squared error between the intensity of each observed pixel and the intensity calculated from the rendering equation

$$I(p) = \frac{\rho_d}{\pi}L_0 + \sum_{i=1}^{n} \beta(\theta_i, \phi_i, \theta_o, \phi_o)\Gamma(p,i)\cos(\theta_i)\frac{1}{||\mathbf{V}_i - \mathbf{P}_p||^2}L_i, \quad (3)$$

where ρ_d is the diffuse reflectance parameter of the Ward model, L_0 is the intensity of the ambient light, n is the number of light sources, $\beta(\theta_i, \phi_i, \theta_o, \phi_o)$ is the Ward BRDF model, $\Gamma(\cdot)$ is a function that is 1 when point p is visible to light source i and 0 otherwise, $\mathbf{V}_i$ is the position vector for light source i, $\mathbf{P}_p$ is the position vector for the point p, and L_i is the intensity for light source i.

Once a BRDF model for an object has been calculated, the lighting distribution can be refined using the BRDF model. Vogiatzis et al. [18] choose to alternately optimise the lighting distribution and the BRDF model until both converge. In this paper however, the BRDF model parameters and lighting distribution are optimised simultaneously, which leads to faster convergence and has a lower likelihood of converging on a local minimum.

4. Data acquisition

Four data sets are used to generate the results in this paper: two single-view data sets of marbles and two multi-view data sets of rocks. The geometry for the marbles is approximated by a sphere and therefore data does not need to be captured in these cases. Geometry data is captured for the two rock data sets.

Image data is needed for all four data sets to make reflectance measurements. A single distant point light source is used.

4.1. Capturing geometry data

The geometry of an object is represented by a 3D model that closely approximates its structure. A 3D mesh model of an object consists of vertices connected together in a mesh, with each facet of the model forming a triangle. A mesh model not only yields shape information but also contains the normal vectors for each facet. This is important for measuring the angles at which the light hits the object surface relative to the viewing angle. Measuring these two angles is fundamental in the measurement of the reflectance of an object.

Geometry data is captured using the *NextEngine Desktop 3D Scanner*. It is a multi-stripe laser triangulation 3D scanner that interfaces with its own proprietary software to produce 3D models of real world objects. The 3D models it produces are accurate up to 0.125 mm in *macro* mode and 0.375 mm in *wide angle* mode [13]. The high level of accuracy of the model means it can be used as a ground truth or baseline description of the geometry of the object.

The scanner captures colour information for each view of the object and then texture maps these images onto the 3D model. Regions of overlapping colour data are blended. The colour information captured is adequate for visualisation purposes, but qualitative experiments have shown that the colour data can contain significant errors and are not suitable for photometric modelling.

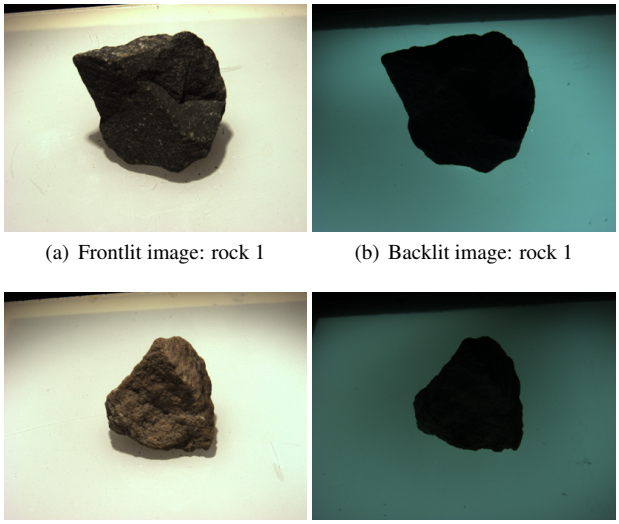
It is important to have control over the lighting conditions of the room for reflectance and lighting recovery so that ambient light and the positions and types of light sources used are well suited to capturing data for photometric modelling.

4.2. Capturing image data

Colour data is captured using a digital camera because the data acquired by the 3D scanner is not suitable for photometric modelling. A 1024×768 colour *Point Grey Flea* camera mounted on a tripod is used to capture frontlit and backlit images of the objects. The camera's gamma and gain parameters are fixed at unity so that the intensity response of the camera is as close to linear as possible and noise is minimal. The camera aperture is chosen to be just large enough so that the brightest regions of the image are almost saturated when the shutter speed is at a maximum, but small enough to maintain a large depth of field. A large depth of field ensures that all parts of the sample are in focus.

The object is positioned on top of a fluorescent lightbox to simplify silhouette extraction from the backlit images. A programmable and accurate turntable is used to change the position of the object and light source relative to the camera. The halogen light source mounted on a stand is positioned on the turntable approximately 50 cm above the object. To minimise ambient light, the room is darkened so that the halogen light source is dominant.

The turntable is used to position the object at 20 different orientations, making up a complete 360° revolution. Each orientation is rotated 18° from the previous one. The frontlit image is captured with the halogen light source on and the fluorescent lightbox off. The backlit image is captured with the opposite configuration. A frontlit and backlit image is captured for each orientation.



(a) Frontlit image: rock 1 (b) Backlit image: rock 1

(c) Frontlit image: rock 2 (d) Backlit image: rock 2

Figure 1: Frontlit and backlit images of objects. The frontlit image is used to extract colour information and the backlit image is used to extract the silhouette of the object.

4.3. Calibration

The images in the rock data sets are captured as a turntable sequence. A set of calibration images is captured of a checkerboard pattern positioned on the turntable, which is used to calculate the extrinsic camera parameters for each turntable position. This calibration step is performed by means of the *Camera Calibration Toolbox* for MATLAB [2].

The grid points extracted from the the turntable image sequence of the checkerboard calibration pattern are coplanar because the calibration pattern is rotating about a fixed axis. Calibrating intrinsic camera parameters requires non-coplanar grid points. As a result, the intrinsic camera parameters cannot be determined from these images. A separate set of calibration images is required with non-coplanar data points spread over the entirety of the image plane at varying depths. Intrinsic parameters, including focal length, distortion and principal point, can be determined from this calibration image sequence.

The Camera Calibration Toolbox gives a good estimate of the intrinsic and extrinsic camera parameters. However, there is a chance that the calibration results contain small errors due to slight variations in the position of the object. These variations can be brought on by vibrations from the turntable or small errors in turntable position.

A final calibration step similar to a *bundle adjustment* [7] optimises the camera parameters to minimise these errors. The silhouettes extracted from the backlit images are used to calculate the *epipolar tangency* (ET) error. ET error is defined as the

squared perpendicular distance between the tangent point on the silhouette for a particular epipole and the epipolar line. A bundle adjustment involves optimising the camera parameters for all cameras such that the ET error is a minimum. This optimisation is performed using the Levenberg-Marquardt non-linear minimisation method as implemented in the MATLAB *lsqnonlin* function.

5. Data pre-processing

Before reflectance and lighting estimates can be made from input 3D and image data, the raw data resulting from the data capture stage need to be processed to account for limitations and inaccuracies in the data capture process. The pre-processing steps include undistorting image data to remove radial and tangential distortion introduced by the camera lens, and aligning image and 3D data so that the two forms of data can be processed within the same reference frame.

5.1. Undistorting image data

When calibrating the intrinsic properties of the cameras, a 5-parameter combined radial and tangential distortion model of the camera lens is calculated. The distortion model produced by the Camera Calibration Toolbox is non-linear and hence cannot be modelled by the camera matrix. As a result, the image data are undistorted directly by generating new images with the distortion removed. Both frontlit and backlit images are undistorted. To reduce the noise in the undistorted silhouette, the silhouette is first extracted from the unprocessed image and then the extracted silhouette coordinates are undistorted. The undistortion is performed using functions from the Camera Calibration Toolbox.

5.2. Aligning image and 3D data

For measuring reflectance and lighting data from images, the colour data at points in the images must correspond to the correct points on the surface of the 3D model. This means that the coordinate system in which the cameras are specified must be aligned to the coordinate system in which the 3D model is specified. This can be achieved by transforming the vertices of the 3D model with a transformation matrix $\mathbf{T} = [\lambda\mathbf{R}\ \ \mathbf{t}]$ where λ is a scaling factor, $\mathbf{R}$ is the rotation matrix that aligns the orthogonal vectors, and $\mathbf{t}$ is the translation vector between the origins.

The transformation matrix $\mathbf{T}$ is optimised iteratively by minimising the ET error between pairs of silhouettes generated from the transformed 3D model and backlit image data. The visual hull [10], or volume of intersection of the silhouettes from each camera, is used as an approximate 3D model that lies in the same coordinate system as the cameras. A good initialisation is required for the transformation matrix to ensure convergence, especially due to the rotational degrees of freedom of the matrix.

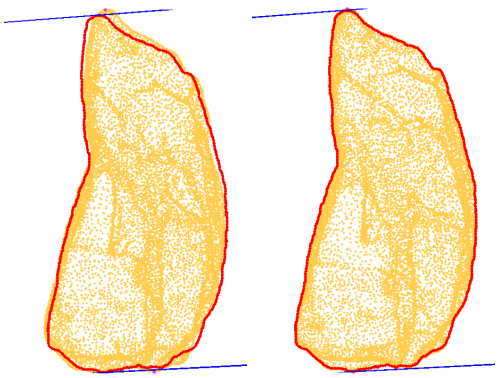
The eigenvectors of the vertices of the 3D data form reliable orthogonal bases that have approximately the same orientation as the vertex data for both sets of 3D data. The eigenvectors are used along with the centroids of vertices of the visual hull and ground truth. The ground truth data is translated so that the origin coincides with the centroid of the vertices. The vertices are rotated so that the eigenvectors of the ground truth data are aligned with the eigenvectors of the visual hull. The vertices of the ground truth data are then translated so that the centroid coincides with that of the visual hull. The scaling factor is initialised to be the average ratio of the caliper diameter measurements [3, p. 12] along the directions defined by the orthogonal bases of the visual hull and ground truth data.

The cost function that is minimised to find the optimum transformation matrix is defined as

$$e(\mathcal{C}, \mathcal{S}, \mathbf{T}, \mathbf{V}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[\Delta(\mathbf{C}_i, \mathbf{C}_j, \mathbf{S}_j, \beta(\mathbf{TV}, \mathbf{C}_i))\right]^2 \quad (4)$$

where $e(\cdot)$ is a function that returns the sum of squared distances between the epipolar lines and tangent points for all camera pairs, $\mathcal{C}$ is the set of cameras, $\mathcal{S}$ is the set of silhouette boundary coordinates, N is the number of views, $\mathbf{T}$ is the transformation matrix that is applied to mesh vertices $\mathbf{V}$, $\Delta(\cdot)$ is a function that returns the epipolar tangency error for a pair of cameras $\mathbf{C}_i$ and $\mathbf{C}_j$ with boundary points $\beta(\mathbf{TV}, \mathbf{C}_i)$ and $\mathbf{S}_j$, and $\beta(\cdot)$ is a function that returns the projection of vertices $\mathbf{V}$ into camera $\mathbf{C}_i$.

Figure 2 shows the initial starting point and also the result of the optimisation process of aligning the 3D model to the image data. The starting point obtained using the initial guess for the transformation matrix is close to the optimum solution. After the final iteration, the tangent points on the 3D model and epipolar lines approximately coincide, as is the expected outcome when minimising Equation 4.



(a) Before optimisation (b) After optimisation

Figure 2: Results of aligning 3D data to image data as seen from a single viewpoint. Figures 2(a) and 2(b) show the alignment of the image silhouette (bold red outline) with the 3D model (orange points) before and after of the optimisation process respectively. The initial estimate for alignment can be seen to be a good starting point, since the silhouette and 3D model are only slightly misaligned. The image silhouette and 3D model are well aligned after the final iteration. The epipolar tangent lines can be seen in blue at the top and bottom of both images. The tangent points on the 3D model are shown in magenta and can also be seen at the top and bottom of each image.

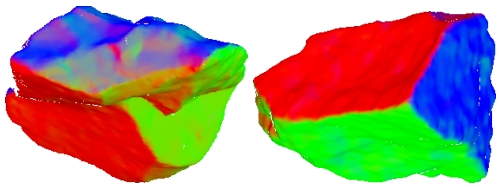
6. Results

A representation of the positions of the point light sources is shown in Figure 3 for the rock data sets. Each image shows the objects as viewed from the positions of the point light sources in each scene, this indicates the portion of the surface that is illuminated by each light.

Figure 4 shows a spherical plot of the measured reflectance data and the model that fits the data. The data come from the first marble data set. The diffuse reflection (constant radius) and the specular highlight (radial spike) can be seen in the plot.

Figure 5 shows the results of the lighting and reflectance recovery process for four data sets comprising of two single-view data sets of an opaque glass marble and two multi-view data sets of different rocks. Figures 5(a), 5(d), 5(g) and 5(j) show the original image data with background information removed. Figures 5(b), 5(e), 5(h) and 5(k) show a rendered image of each object that is generated using only the recovered lighting distribution and reflectance parameters for each data set. The positions of the highlights and shadows in the rendered images correspond to those in the original images. Figures 5(c), 5(f), 5(i) and 5(l) are difference images that show the difference in intensity of the green colour band between the original image and the rendered image. The red and blue colour bands exhibit similar behaviour. The green colour band is used because there are twice the number of green pixels in the Bayer pattern of the colour image than red or blue. Fewer interpolation operations are required on the green data making it more accurate.

The rendered objects can be seen to closely resemble the original images. The rendered image of the first rock data set (Figure 5(h)) does not capture the spatial variation of the material present in the original image (Figure5(g)) due to the limitation that the material is assumed to be homogeneous, i.e. the appearance is modelled by one set of reflectance parameters. As a result, the reflectance parameters model the average appearance of the objects, which is especially obvious in the grey appearance of the rendered image of the first rock data set.



(a) Rock 1 (b) Rock 2

Figure 3: Each of the rock data sets as seen from each light source. Each view is from the position of the recovered light source and indicates which faces are lit by each light source. The colour represents an RGB encoding of the surface normals. In these scenes, lighting is represented by a single point light source and an ambient light source. Ray casting is used to determine which light sources illuminate each face.

7. Conclusion

This paper details the data capture process for measuring the reflectance properties of objects from images and highlights considerations that need to be taken into account. The geometry information is captured using a NextEngine Desktop 3D Scanner, which provides more accurate data than image-based methods. Colour information is captured separately from turntable sequences and aligned with the geometry information using the epipolar tangency constraint. Ward reflectance model parameters are estimated through a regression process that matches the predicted appearance with the original image data.

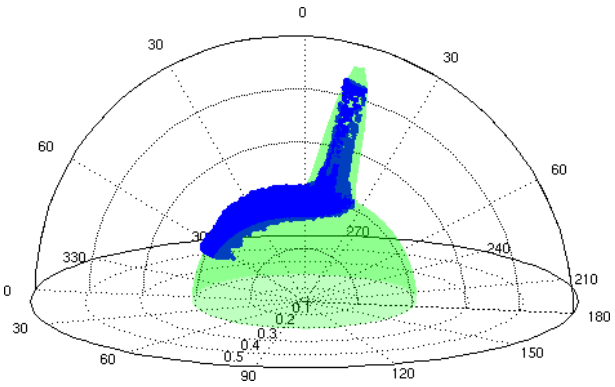


Figure 4: BRDF data and model plotted against the azimuthal and zenith angles of observation. The BRDF data are plotted as blue points with the Ward model predication plotted as a green surface. The radial spike corresponds to a specular highlight and the regions of constant radius correspond to the diffuse colour. The parameters of the Ward model shown here are $\rho_d = 0.647$, $\rho_s = 0.0127$ and $\alpha = 0.0629$.

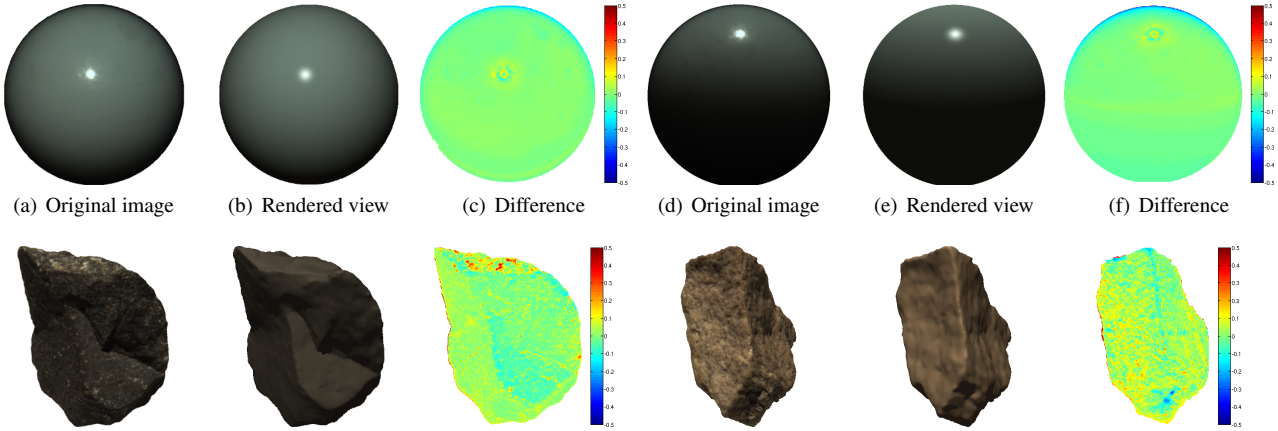
Qualitative results show promise, with renderings comparing closely to original images. These results indicate that the reflectance and lighting modelling succeeds in modelling the appearance of the objects, with discrepancies only appearing when more than one material is present in an image. This and other limitations are to be addressed in future work.

8. Future work

The following avenues are envisioned as future work: a quantitative analysis of the accuracy of surface normals obtained from the visual hull as compared to 3D scanner data; an analysis of the effect on accuracy of reducing the number of triangles in the geometry model to find a balance between processing speed and accuracy; an analysis of the effect on accuracy of reducing the number of data points in the sample for reflectance and lighting recovery to find a balance between processing speed and accuracy; extending the reflectance model to account for objects made of more than one material and spatial variation in material on the surface of the object in a similar manner to Lensch et al. [11]; and colour calibration to ensure linearity in colour measurements;

9. Acknowledgements

The authors would like to thank De Beers and the National Research Foundation for their financial assistance.



(a) Original image (b) Rendered view (c) Difference (d) Original image (e) Rendered view (f) Difference

(g) Original image (h) Rendered view (i) Difference (j) Original image (k) Rendered view (l) Difference

Figure 5: Results of lighting and reflectance recovery. Ground truth images, rendered views and difference images for the two marble data sets and the two rock data sets. The shadows and highlights in the rendered images can be seen to correspond with those in the original images. The difference images show the difference in intensity of the green colour band between the original image and the rendered image. Positive values occur when the original image is greater in intensity than the rendered image and vice versa for negative values.

10. References

[1] N. Birkbeck, D. Cobzas, P. Sturm, and M. Jägersand. Variational shape and reflectance estimation under changing light and viewpoints. In *ECCV '06, Graz, Austria*, volume 1, pages 536–549. Springer, May 2006.

[2] J.-Y. Bouguet. Camera calibration toolbox for MATLAB. URL `http://www.vision.caltech.edu/bouguetj/calibdoc/`. Last accessed 14/08/08.

[3] K. Forbes. *Calibration, Recognition, and Shape from Silhouettes of Stones*. PhD thesis, University of Cape Town, June 2007.

[4] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003. ISBN 0130851981.

[5] A. S. Georghiades. Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In *ICCV '03*, page 816, Washington, DC, USA, 2003. IEEE Computer Society.

[6] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. In *ICCV '05*, pages 341–348, Washington, DC, USA, 2005. IEEE Computer Society.

[7] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[8] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. In *SIGGRAPH '91*, pages 175–186, New York, NY, USA, 1991. ACM Press.

[9] E. P. F. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *SIGGRAPH '97*, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[10] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, 1994.

[11] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. Graph.*, 22 (2):234–257, 2003.

[12] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, K. E. Torrance, and D. P. Greenberg. Image-based BRDF measurement including human skin. In *Eurographics Workshop on Rendering*, Granada, Spain, June 1999.

[13] NextEngine. NextEngine FAQ. URL `http://www.nextengine.com/`. Last accessed 14/08/08.

[14] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.

[15] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH '01*, pages 117–128, New York, NY, USA, 2001. ACM Press.

[16] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *SIGGRAPH '97*, pages 379–387, New York, NY, USA, 1997. ACM Press.

[17] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of Optical Society of America*, 57(9):1105–1114, 1967.

[18] G. Vogiatzis, P. Favaro, and R. Cipolla. Using frontier points to recover shape, reflectance and illumunation. In *ICCV '05*, pages 228–235, Washington, DC, USA, 2005. IEEE Computer Society.

[19] G. J. Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH '92*, pages 265–272, New York, NY, USA, 1992. ACM Press.

[20] T. Yu, N. Xu, and N. Ahuja. Recovering shape and reflectance model of non-lambertian objects from multiple views. In *CVPR '04*, volume 2, pages 226–233, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

Experiments in automatic assessment of oral proficiency and listening comprehension for bilingual South African speakers of English

Febe de Wet[1], *Pieter Müller*[3], *Christa van der Walt*[2] *& Thomas Niesler*[3]

[1]Centre for Language and Speech Technology (SU-CLaST), [2]Department of Curriculum Studies,
[3]Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa.
{fdw,pfdevmuller,cvdwalt,trn}@sun.ac.za

Abstract

We describe ongoing research into the automatic assessment of listening comprehension and oral language proficiency of South African L2 English speakers. Proficiency indicators are extracted from the speech signals by means of an automatic speech recognition system, and compared with assessments of the same speech by human experts. By means of carefully designed assessment scales, we are able to achieve high intra-rater correlations for the human scores. We show that, in accordance with the findings of other authors, rate of speech (ROS) is the most successful among the automatically derived measures that were evaluated. We also determine the effect of including context dependency in the speech recogniser's acoustic models, and investigate the effect which reciprocal transformations have on the correlations with human scores. Our results provide no evidence to support the hypothesis that context independent acoustic models yield better proficiency indicators when dealing with non-native speech. We also find that the use of the reciprocal of ROS does not lead to consistently better correlations.

1. Introduction

Assessment of a student's entrance level language skills for the purpose of placement into appropriate language programmes is often restricted to reading and writing proficiency tests. Listening and speaking skills are frequently not assessed because they either require specialised equipment or labour intensive procedures. In addition, the assessment of oral skills is generally highly subjective, and efforts that enhance inter-rater reliability further increase the labour intensiveness of the assessment process. The assessment of reading and writing comprehension skills, on the other hand, can be automated by means of computerised multiple choice tests, which reduce the time and manpower requirements for their administration. However, research has shown that good results in written tests are not necessarily good predictors of corresponding results in an oral test [1].

This study describes progress in an ongoing effort to develop an automated system to assess the listening comprehension and oral language proficiency of large numbers of students. The system will operate within the specific context of the Education Faculty at Stellenbosch University, where new students are required to obtain a language endorsement on their teaching qualification. For English, this means that students must enrol for a language module appropriate to their level of proficiency, and that their progress must be monitored regularly thereafter. With a current ratio of between 100 and 200 students per university staff member, this is only feasible when placing a major emphasis on computerised multiple-choice reading and writing tests. However, students regard oral proficiency as an important component of their teaching abilities. Consequently, they object to an exclusive focus on writing and reading skills, and regard the infrequency with which their oral skills are assessed with much suspicion. A technological solution may not only lighten the heavy workload of staff, but also provide a more transparent and objective metric with greater acceptance among students.

A factor which sets this study apart from others is that the L2 proficiency of the test population is always high and varies from intermediate to advanced. In contrast, the proficiency of the subjects in other studies varies to a much greater degree [2, 3, 4, 5]. Our research therefore focuses on students who speak English as a *second* language rather than a *foreign* language.

2. Computerised test development

The goal of the computerised test was to assess listening and speaking skills limited to the specific context of secondary school education. The test was therefore designed to evaluate language behaviour that is specifically relevant to this domain. There was no attempt to mimic natural human dialogue except in the sense that the test content relates specifically to teaching and learning in a school environment.

A telephone-based test was implemented because it requires a minimum of specialised equipment and allows flexibility in terms of the location from which the test may be taken. Past experience at the Faculty of Education has indicated that on-line telephone assessments using human judges give a fair indication of oral and aural proficiency.

2.1. Test design

The test was designed to include instructions and tasks that require comprehension of spoken English and elicit spoken responses from students. In this paper we will focus on two of the seven tasks that comprise the test, namely the *reading* and the *repeating* tasks. For a detailed description of the complete test, the reader is referred to [6].

- **Reading task:** Students are provided with a list of 12 sentences on a printed test sheet. The system randomly chooses six of these sentences, and instructs students to read each one in turn. For example, *"School governing boards struggle to make ends meet."*
- **Repeating task:** Students are asked to listen to sentences uttered by the system and to repeat the same sentence. For example, *"Student teachers do not get enough exposure to teaching practise."*

The first task is a familiar one to students since reading aloud is a task they had to complete successfully for their fi-

nal school examinations. Moreover, the students could rely on the printed test sheet, which helped nervous candidates to relax.

The construction of the repeating task is based on the hypothesis that phonological working memory capacity influences oral production in first language users [7, 8] and even more so in second language learners [9, 10]. In terms of this hypothesis, second language learners will struggle to produce the target language in face-to-face communication because of time pressure in conjunction with limited access to vocabulary and the L2 sound system.

The sentences in the repeat task were designed with the context of students' experiences as teacher trainees in mind, and ranged from fairly simple (e.g. *It is boring to sit and watch teachers all day.*) to longer and more complex sentences where the subject is a separate clause (e.g. *How parents' interests and hopes are accommodated is crucial to the success of a school.*). In the case of advanced learners, it was assumed that their working memory capacity in the second language would make it possible for them to repeat the sentences accurately.

2.2. Test implementation

A spoken dialogue system (SDS) was developed to guide students through the test and to capture their answers. To make the test easy to follow, the system's spoken prompts were recorded using different voices for test guidelines, for instructions and for examples of appropriate responses. The SDS plays the test instructions, records the students' answers, and controls the interface between the computer and the telephone line. In a fully operational system, the SDS would also control the flow of data to and from the ASR system, but in our set-up the students' answers were simply recorded for later, off-line processing.

2.3. Test administration

A number of students volunteered to test the SDS in a pilot experiment. 120 students subsequently took the test as part of their oral proficiency assessment. The majority of the students speak Afrikaans as a first language and their proficiency in English varies from intermediate to advanced. Calls to the SDS were made from a telephone located in a private office reserved for this purpose. Oral instructions were given to the students before the test. In addition to the instructions given by the SDS, a printed copy of the test instructions was provided. No staff were present while the students were taking the test.

3. Human assessments

Teachers of English as a second or foreign language were asked to rate speech samples from the read and repeat tasks in the test. The raters were not personally acquainted with the students they rated.

A subset of 90 students was selected from the group of 120 who took the test. Students were chosen to represent male and female as well as Afrikaans and English mother tongue speakers in accordance with the composition of the student population at the Faculty of Education. Students were chosen to ensure a balanced test population with regard to mother tongue and gender. Given the large number of students, it was not feasible to have each utterance of every student rated. Three examples of each student's read and repeat responses were randomly chosen to be judged by the raters.

Six raters each assessed 45 students and each student was assessed by three human raters. In order to measure intra-rater consistency, five students were presented twice to each rater. Each rater therefore performed 50 ratings: 45 unique and 5 repeats.

Before judging the students, the raters attended a training session on the use of the rating scales. Example utterances and their respective ratings were also presented.

For the reading task, each sentence was assessed on three separate scales in terms of degree of hesitation, pronunciation (including accent) and intonation, as shown in Figure 1. The scales were conceptualised as a continuum on which certain points are described, with the possibility to mark points between two descriptions.

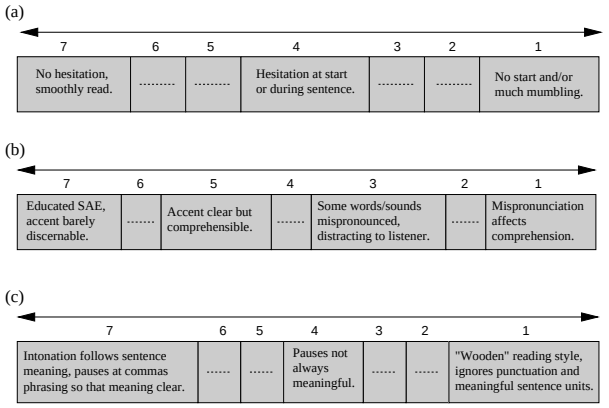


Figure 1: Scales used to assess (a) degree of hesitation (b) pronunciation and (c) intonation in the reading test.

The numbers above the scales are meant to guide the eventual mark allocation, providing numerical information that can be used to grade students. Scores below three on the scale would indicate students who need additional language support. However, numerical scores were not included on the scales supplied to the raters in order to avoid pre-conceptions about student grades.

For the repeating task, a different set of scales was designed in order to measure the success with which a repetition was formulated and the accuracy of the repetition, as shown in Figure 2. In this case the scales contained precise descriptions for all the categories.

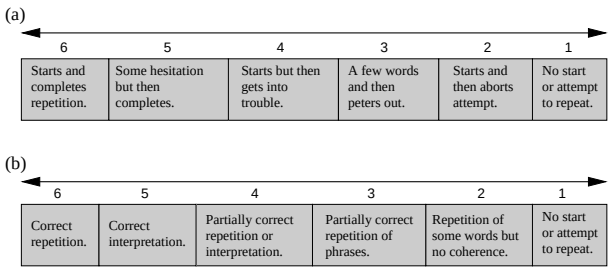


Figure 2: Scales used to assess (a) degree of success and (b) accuracy in the repeating test.

During the pilot test it had become clear that students did not necessarily repeat each sentence accurately, but were nevertheless able to comprehend it and could repeat a rendition that reflected the meaning of the original sentence. Since the

test was also intended to measure listening comprehension, it seemed fair to distinguish correct *repetitions* from correct *interpretations*, since the latter would indicate that the students responded by interpreting what they heard. This kind of behaviour offers a glimpse into a speaker's working memory, which seems to reduce information into meaningful chunks in order to make sense of an incoming message.

3.1. Results: Human assessments

Table 1 shows the intra-rater correlations[1] that were obtained using these scales. These values are much higher than those obtained in our previous study based on global assessments [6]. This seems to indicate that the more detailed assessment guidelines introduced in this study assist the human raters in allocating marks more consistently.

Rater	Intra-rater correlation
1	0.83
2	0.94
3	0.81
4	0.96
5	0.67
6	0.91

Table 1: Intra-rater correlations for human raters.

Figures 3(a) and 3(b) illustrate the inter-rater agreement for the read and repeat tasks, respectively.

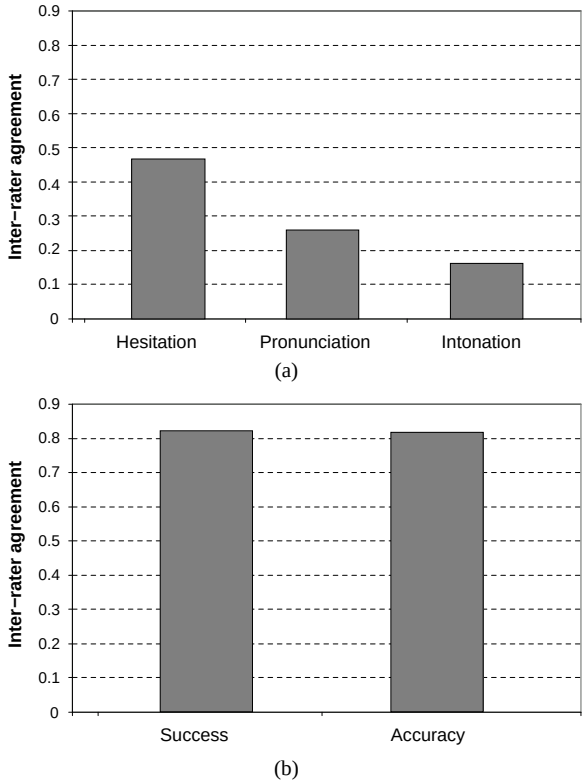


Figure 3: Inter-rater agreement for the read (a) and repeat (b) tasks.

[1]The correlations are two-way random, intra-class correlation coefficients and were calculated using Statistica [11].

The three bars in Figure 3(a) indicate the values for the three scales shown in Figure 1, and the two bars in Figure 3(b) the values for the scales in Figure 2. By comparing Figures 3(a) and 3(b) we see that the inter-rater agreement was higher for the repeat than for the read task. The raters clearly disagree in their assessment strategies for the pronunciation and intonation aspects of the reading task.

The values shown in Figure 3(a) for the read speech are lower than those reported in [3] and [4], but are similar to those reported in [5]. The fact that our test population is fairly homogeneous in terms of proficiency could explain this observation. The raters appear to be less consistent in their assessments when there is little variation in proficiency. In studies where higher inter-rater agreement was measured, the speaker populations were more diverse in terms of L2 proficiency. Furthermore, in our experiments the human judges also rated fewer utterances per speaker, i.e. two or three as opposed to 10 in [3] and 30 in [4].

The average score (percentages calculated across all raters) for the read and repeat tasks are shown in Figures 4(a) and 4(b). The standard deviation around the mean values is indicated by the vertical lines in the figures. Figure 4 shows that students performed better in the reading task than in the repeating task and that, on average, they were given good marks. In previous studies we observed that only the top part of the assessment scales were used by the judges, especially for the reading task [6]. Despite our efforts to 'broaden' the assessment scales in this experiment, the lower extremes of the scales were again rarely chosen.

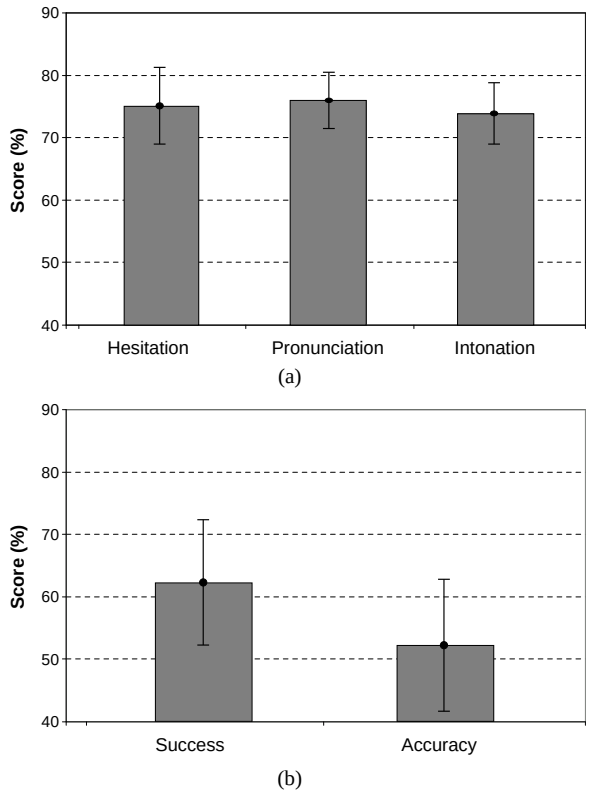


Figure 4: Average scores for the read (a) and repeat (b) tasks. The standard deviation around the mean is indicated by the vertical line in each bar.

4. ASR-based assessment

Numerous studies on the role of ASR in language learning applications have been published in the last decade e.g. [2, 3, 4, 12]. However, the investigations reported on in the literature differ in terms of several aspects of experimental design. As a result, it is difficult to make direct comparisons between studies. Nevertheless, a common aim of most studies in this field is the identification of parameters that can be automatically derived from speech data and that correlate well with human judgements of oral proficiency.

4.1. ASR system

ASR is a relatively new research field in South Africa and the resources that are required to develop applications are limited. During the *African Speech Technology* project, telephone speech databases were compiled for South African English, isiZulu, isiXhosa, Sesotho and Afrikaans [13]. Prototype speech recognisers were subsequently developed for each language, and this study makes use of the standard South African English ASR system.

A training set consisting of approximately six hours of phonetically-annotated telephone speech data was parametrised as Mel-frequency cepstral coefficients (MFCCs) and their first and second differentials. Cepstral mean normalisation (CMN) was applied on a per-utterance basis.

A set of 52 speaker-independent monophone hidden Markov models (HMMs) with three states per model and 64 mixtures per state was trained on this data by embedded Baum-Welsh re-estimation using the HTK tools [14]. A set of speaker-independent cross-word triphone HMMs was then obtained using decision-tree state clustering, resulting in a total of 4797 clustered states. Each triphone model employed eight Gaussian mixtures per state and diagonal covariance matrices. The phone recognition accuracies of the monophone and the triphone models on a separate test set and using a bigram language model are shown in Table 2.

Model	Phone Recognition Accuracy (%)
Monophone	58.4
Triphone	73.0

Table 2: Phone recognition accuracies measured for the monophone and triphone acoustic models.

The students' responses to the test were transcribed orthographically by human annotators. The data that was assessed by the human raters was used as an independent test set (90 speakers). The remainder of the data (30 speakers) was used as a development test set.

For each sentence in the the reading task, a finite-state grammar was constructed allowing two options: the target utterance and "I don't know". Students were instructed to say "I don't know" if they were unsure about how to respond to a test item. Filled pauses, silences and speaker noises were permitted between words by the grammar. The recogniser's word insertion penalty was chosen to ensure optimal correlation between the ROS values derived from the manual and automatic transcriptions of the development test set.

For the repeat task, a unigram language model with equal probabilities for all words was derived from the manual transcriptions of the development test set. A separate language model was constructed for each sentence of the repeat task. The recogniser's word insertion penalty and language model factor were chosen to maximise the correlation between recognition accuracy as well as the ROS values derived from the manual and automatic transcriptions of the development test set.

4.2. Automatically derived proficiency indicators

Many indicators of oral proficiency that can be automatically derived from speech data have been proposed in the literature. We have chosen three that have been reported to perform best by several authors, namely rate of speech, goodness of pronunciation, and transcription accuracy.

4.2.1. Rate of speech

Previous studies have found that, for read speech, rate of speech (ROS) is one of the best indicators of fluency [3, 12]. In our experiments ROS was calculated according to Equation (1), as proposed in [15].

$$ROS = \frac{N_p}{T_{sp}} \tag{1}$$

The quantity N_p denotes the number of speech phones in the utterance, while T_{sp} is the total duration of speech in the utterance, including pauses.

The correlation between the ROS values derived from the manual and automatic transcriptions of the test data were 0.98 and 0.94 for the read and repeat data, respectively. These values indicate that the automatic system's ability to segment the speech into phones compares very well with its human counterpart.

4.2.2. Goodness of pronunciation

As an example of the general class of posterior HMM likelihood scores [4, 12], we used the "goodness of pronunciation" (GOP) proposed in [2]. The GOP score of phone q_i is defined as the frame-normalised logarithm of the posterior probability $P(q_i|O)$, where O refers to the acoustic segment uttered by the speaker.

$$GOP(q_i) = \frac{|log(P(q_i|O))|}{NF(O)} \tag{2}$$

In equation 2, $NF(O)$ corresponds to the number of frames in acoustic segment O. A GOP score was determined for each phone in an utterance and utterance level scores were subsequently obtained by taking the average of all the phone scores in the utterance.

Some authors claim that less detailed native models, like monophone HMMs, perform better for non-native speakers than detailed native models like triphone HMMs [16, 17]. Others report very small differences between the results obtained with monophone and triphone models for non-native speakers [18]. In this study we investigate the influence of model complexity on automatically derived proficiency indicators by deriving GOP scores from monophone (GOPmono) as well as cross-word triphone (GOPxword) HMMs.

4.2.3. Transcription accuracy

Because highly restrictive finite-state grammars were used for the reading task, the recognition accuracy obtained for the read responses was in all cases very high and therefore not used as a proficiency indicator. Repeat accuracy, on the other hand, was considered as a proficiency indicator, as is also proposed in [12].

This accuracy was determined by comparing the ASR output for the repeated utterances to the orthographic transcriptions of the sentences students were prompted to repeat during the test. Accuracy was subsequently calculated according to Equation 3, as proposed in [14]:

$$Accuracy = \frac{H - I}{N} \times 100\% \quad (3)$$

In Equation 3, H is the number of correctly recognised words, I is the number of insertion errors and N is the total number of words in an utterance.

4.2.4. Nonlinear transformations

Research has shown that it is possible to improve the correlation between automatically derived indicators and human ratings by using a non-linear combination of several machine scores. However, it was found that these improvements are often due mainly to the non-linear transformation of a single indicator that already correlates well with human ratings [19]. One such non-linear transformation is the reciprocal, and experiments have suggested that using $\frac{1}{\text{ROS}}$ instead of ROS leads to a slightly higher correlation with human ratings [12]. We will establish whether this is true for our experimental conditions in the following section.

4.3. Results: ASR-based assessment

The ASR system judged the same material previously evaluated by the human raters. The average ROS, accuracy and GOP values that were measured for the read and repeat tasks are shown in Table 3.

	ROS	Accuracy	GOP
Read	11.94	-	3.88
Repeat	9.81	50.16	4.05

Table 3: Average ROS, accuracy and GOP scores for the test data.

The observation that average ROS is higher for the reading task than for the repeat task is in agreement with what one would intuitively expect, given the level of difficulty of the tasks. The correlations measured between the ROS, accuracy and GOP scores are listed in Table 4.

Task	Score pair	Correlation
Read	ROS & GOP	0.01
Repeat	ROS & Accuracy	0.75
Repeat	ROS & GOP	-0.44
Repeat	Accuracy & GOP	-0.40

Table 4: Correlation between ROS, accuracy and GOP scores for the read and repeat tasks.

Table 4 shows that repeat accuracy correlates strongly with ROS and to a lesser extent with the GOP scores. In contrast, there is no correlation between ROS and the GOP scores for the read data and only a weak correlation for the repeat data. This observation seems to indicate that ROS is not related to the acoustic properties of the data. ROS and GOP scores could therefore be used to evaluate different aspects of speech.

5. Correlation between human and ASR-based assessment

Table 5 gives the correlation[2] between the scores given by the human raters and the automatically derived proficiency indicators for the reading task. The highest correlation in Table 5 is observed between degree of hesitation and ROS. To the extent that degree of hesitation is an indicator of fluency, this result is consistent with what has been reported in the literature [15].

Indicator	Hesitation	Pronunciation	Intonation
ROS	0.53	0.46	0.49
1/ROS	0.55	0.45	0.51
GOPmono	0.03	0.18	0.01
GOPmono/ROS	0.37	0.19	0.39
GOPxword	0.11	0.13	0.05
GOPxword/ROS	0.43	0.19	0.36

Table 5: Correlation between human and automatic scores for the reading task.

The GOP scores show almost no correlation with the human judgements of the read material. This result is similar to the observation made in [3], where the weakest correlation between human and automatic scores was measured for likelihood ratios. This trend seems to indicate that posterior scores derived at the utterance level do not provide meaningful information on pronunciation. Discriminating between GOP scores for vowels and consonants or deriving phone-specific GOP scores for a number of "problematic" phones may improve the GOP scores' correlation with the human data.

Table 6 shows the correlation between the scores the human raters assigned for the repeat task and those derived automatically using the ASR system.

Indicator	Success	Accuracy
Accuracy	0.68	0.69
ROS	0.71	0.68
1/ROS	0.71	0.66
GOPmono	0.31	0.32
GOPmono/ROS	0.60	0.57
GOPxword	0.40	0.40
GOPxword/ROS	0.67	0.63

Table 6: Correlation between human and automatic scores for the repeat task.

ROS as well as accuracy correlate well with the human scores. However, it should be kept in mind that these variables are also strongly correlated with each other for the repeat task (Table 4). The GOP scores are only poorly correlated to the human ratings of the repeat data, but the correlations are consistently higher than those in Table 5.

The results in Tables 5 and 6 show that there is no consistent improvement in correlation when using the reciprocal of ROS as a proficiency indicator. This is in contrast to the results reported in [12], where small improvements were observed. The two tables also show that the assertion made in [16, 17] that monophone acoustic models are more appropriate than triphone models when dealing with non-native speech is not borne out by our experiments. The small improvement observed for using

[2] Spearman rank correlation coefficients were derived (using Statistica [11]) because the data in question is ordinal.

context sensitive models (Table 6) is consistent with the results reported in [18].

6. Discussion and conclusion

We have described progress made in our effort to develop an automated system to assess the listening comprehension and oral language proficiency of South African L2 English speakers. Despite our revised and more specific rating scales, we found that the scores allocated by the human raters for the reading task still fall within a narrow range of high marks. We believe that this narrow range led to the associated relatively poor inter-rater agreement, and possibly also the low correlations with the automatically derived indicators. To improve this, we will attempt to increase the difficulty of the read sentences in future implementations of the test, in order to achieve a greater spread of human scores. For the repeat task, the spread of the scores was considerably greater as were their correlation between the automatically derived indicators, especially ROS.

Using the reciprocal of ROS instead of ROS as an indicator showed no consistent improvement in the correlation with the human scores, in contrast with other published research. This probably indicates that the relationship between the scores' distributions in our study is different to the relationships observed in other studies. Other non-linear transformations, such as neural networks and distribution estimation, have also been reported to improve the correlation with human ratings to a greater degree [20]. The effect of these alternative and more flexible transformations on our data will be investigated in future research.

When comparing the effectiveness of context independent (monophone) and context dependent (triphone) acoustic models, we found that the triphones performed slightly better in the repeat task, and there was no consistent difference for the read task. Thus, the finding that context independent models show superior performance for the automatic assessment of non-native speech does not hold for our experimental situation.

7. Acknowledgements

This research was supported by the Fund for Innovation and Research into Teaching and Learning at Stellenbosch University, an NRF Focus Area Grant for research on *English Language Teaching in Multilingual Settings* and the "Development of Resources for Intelligent Computer-Assisted Language Learning" project sponsored by the NHN.

8. References

[1] S. Sundh, *Swedish school leavers' oral proficiency in English*, Ph.D. thesis, Uppsala University, Uppsala, 2003.

[2] S. M. Witt, *Use of speech recognition in computer-assisted language learning*, Ph.D. thesis, Department of Engineering, University of Cambridge, Cambridge, UK, November 1999.

[3] C. Cucchiarini, H. Strik, and L. Boves, "Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithms," *Speech Communication*, vol. 30, pp. 109–119, 2000.

[4] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic scoring of pronunciation quality," *Speech Communication*, vol. 30, pp. 83–93, 2000.

[5] T. Cincarek, "Pronunciation scoring for non-native speech," Master's thesis, Institut für Informatik, Friedrich-Alexander Universität, Erlangen-Nürnberg, Germany, 2004.

[6] C. Van der Walt, F. De Wet, and T. R. Niesler, "Oral proficiency assessment: the use of automatic speech recognition systems," *Southern African Linguistics and Applied Language Studies*, vol. 26, no. 1, pp. 135–146, 2008.

[7] M. Daneman, "Working memory as a predictor of verbal fluency," *Journal of Psycholinguistic Research*, vol. 20, no. 6, pp. 445–464, 1991.

[8] G. Wigglesworth, "An investigation of planning time and proficiency level on oral test discourse," *Language Testing*, vol. 14, no. 1, pp. 85–106, 1997.

[9] N. C. Ellis and S. Sinclair, "Working memory in the acquisition of vocabulary and syntax: Putting language in good order," *Quarterly Journal of Experimental Psychology*, vol. 49, no. A, pp. 234250, 1996.

[10] J. S. Payne and B. M. Scott, "Synchronous CMC, working memory, and L2 oral proficiency development," *Language Learning & Technology*, vol. 9, no. 3, pp. 35–54, 2005.

[11] StatSoft Inc., Ed., *STATISTICA 8.0*, www.statsoft.com, 2008.

[12] C. Hacker, T. Cincarek, R. Gruhn, S. Steidl, E. Nöth, and H. Niemann, "Pronunciation feature extraction," in *Proceedings of 27th DAGM Symposium*, August 2005, pp. 141–148.

[13] J. C. Roux, P. H. Louw, and T. R. Niesler, "The African Speech Technology project: An assessment," in *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004, pp. I:93–96.

[14] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK book, version 3.2.1*, Cambridge University Engineering Department, 2002.

[15] C. Cucchiarini, H. Strik, and L. Boves, "Quantitative assessment of second language learners' fluency by means of automatic speech recognition technology," *Journal of the Acoustical Society of America*, vol. 107, no. 2, pp. 989–999, 2000.

[16] T. Schultz and K. Kirchhoff, *Multilingual speech processing*, chapter Other challenges: non-native speech, dialects, accents and local interfaces, Academic Press, 2006.

[17] X. He and Y. Zhao, "Model complexity optimisation for non-native English speakers," in *Proceedings of Eurospeech*, Aalborg, Denmark, 2001, pp. 1461–1463.

[18] O. Ronen, L. Neumeyer, and H. Franco, "Automatic detection of mispronunciation for language instruction," in *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[19] H. Franco, L. Neumeyer, V. Digalakis, and O. Ronen, "Combination of machine scores for automatic grading of pronunciation quality," *Speech Communication*, vol. 30, pp. 121–130, 2000.

[20] H. Franco, V. Abrash, K. Precoda, H. Bratt, R. Rao, J. Butzberger, R. Rossier, and F. J. Cesari, "The SRI EduSpeak^{TM} system: Recognition and pronunciation scoring for language learning," in *Proceedings of InSTILL 2000*, Dundee, 2000, University of Abertay, pp. 123–128.

Rapid 3D Measurement and Influences on Precision Using Digital Video Cameras

Willie van der Merwe, Kristiaan Schreve

Department of Mechanical and Mechatronic Engineering
University of Stellenbosch, South Africa
kschreve@sun.ac.za

Abstract

Quality assurance and reverse engineering have become an almost inseparable part of the mass production industry. Non-contact measurement methods are playing an ever more important role. This study implements a rapid measurement system using two digital video cameras. Three different methods, using either laser tracking or structured light patterns, were developed and employed to solve the coordinate extraction and correspondence matching problems. The system achieves calibration in less than a minute and accumulates point correspondences at 12 frames per second. Accuracies of better than 0.4 mm are achieved using a single pair of images with 640 x 480 pixel resolution each.

1. Introduction

Optical measurement techniques have traditionally been bound to specific applications requiring expensive and specialised equipment. With the rapidly developing digital technologies in the market, computers and off-the-shelf digital cameras are continually improving in both speed and capability while also becoming less expensive.

Using digital cameras and video cameras as the main data receiver component, the literature reports quite a few techniques that lend themselves to accurate vision metrology [2] [6] [11].

This study shows what can be achieved with the effective combination of simple techniques, readily available software and hardware. The ultimate goal is to build a working vision metrology system capable of rapid measurements, however currently the accuracy is not yet sufficient for this purpose. It is shown that measurement accuracies better than 0.4 mm (for a 235 x 190 x 95 mm volume) can be reached. It also shows that data-sets of thousands of measurements can be made within minutes using the automated and semi-automated processes of calibration, coordinate extraction and stereo-matching developed for the system. Three different methods of correspondence matching are explored and results on measurement precision presented.

2. System Design

As covered in sections 3 and 4, the parameters that mathematically describe the camera model will, to a certain degree, influence the precision of the measurement system. There are however also other factors influencing the precision of calibration and measurement that are mostly independent of the camera model. With the practical implications in mind, some important factors influencing precision, speed and cost are considered here.

2.1. Sub-pixel Target Extraction

In general, the greater the precision with which a feature is extracted, the greater the precision of the calibration.

Before the location of a feature can be determined, the other important consideration is the initial recognition of the features in an image. From an image processing point of view, the simplest way in which to aid automatic detection is by using high contrast features [8]. Examples of this are markers made of reflective material [6] or high contrasted black and white patterns. Using simple geometric shapes for the features, such as circles, rectangles and checkerboard patterns, can then further aid in the recognition phase.

For each of these shapes a different image processing method is used to extract precise target locations. For the rectangles or checkerboard patterns, corners can be initially detected using, for instance, Harris corner-detection. Sub-pixel refinement of the corner locations can then be made using interpolation between pixels [4]. Another method of refining the corner coordinates in these two cases is by using edge information to calculate line intersections [5] [10]. For circular features a number of locating methods are discussed and evaluated by [8].

The precision with which the coordinates of each of these shapes can be extracted using their corresponding methods is influenced differently by lens distortion and perspective effects of an optical system. Mallon and Whelan [5] found that circular patterns yield the least precise target location, being influenced by the lens distortion as well as the perspective effects. The best results were found for the line-intersection method which is invariant under perspective transformation, but is still influenced by lens distortion.

2.2. Angles of Convergence

With an increase in angles between rays formed by the same point, the precision of the calibration network will also increase. The practical implication is that the "base-to-depth"ratio should be as large as possible, up to 1:1, i.e. when the angle of convergence is 90 degrees. The base refers to the distance between camera centres and the depth refers to the perpendicular distance from the base-line to the point being measured. This effect of the converging angles is mentioned throughout the literature [7] [3], but no results were found to indicate the increase of calibration or measurement precision with an increased convergence angle.

2.3. Projective Coupling

Projective coupling refers to the correlation between the internal and external camera parameters. An example given by [9] is

the typical coupling between the principal point location, decentring distortion and the tip or tilt of the camera. Small changes in any of these parameters will still yield the same overall calibration result. For the case where there is a strong projective coupling, [7] as well as [10] makes a similar observation: there is a negligible difference in the final 3D precision if the principal point offset parameters are given different values (within a reasonable range). Remondino and Fraser [7] note this is also true for the decentring distortion terms.

3. Camera Model

The mathematical description has been well established throughout the literature. For this reason a brief description of the final camera model will be given here and the reader is referred to [4] or [3] for further information. In this paper the pinhole camera model is used [3] with a distortion model.

3.1. Linear Camera Parameters

In order to calculate the 3D coordinate of a corresponding point in a stereo-pair of images, the camera matrix, **P**, is needed. This 3x4 matrix describes the projection (or mapping) of a 3D coordinate onto the image plane of the camera for which P was determined using only the linear description of the camera. Equation 1 is the compact notation for this mapping from the world coordinates **X**, to the image coordinates **x**.

$$\mathbf{x=KR[I|-C]X} \tag{1}$$

Starting from the left in Equation 1, the 3x3 matrix **K** describes the camera's internal parameters in terms of pixels. These parameters describe the focal length, the image centre as well as the width to height ratio of the pixels. The following 3x3 matrix, **R**, describes the rotation of the camera and the 3x4 matrix **[I|-C]** is constructed using the position of the camera centre. The final camera matrix containing all these elements is given in Equation 2 and the final mapping using **P** is given in Equation 3.

$$\mathbf{P=KR[I|-C]} \tag{2}$$

$$\mathbf{x=PX} \tag{3}$$

In practice, **P** will be determined directly from the calibration process and the separated internal and external parameter components will not be needed. In this application, **P** is used directly for triangulation (along with the added non-linear parameters), which takes care of the projective coupling problem mentioned in section 2.3.

3.2. Distortion Model

Different mathematical models can be used for radial distortion, but they are most commonly described in the form of some polynomial expansion as a function of the distance from the radial centre, r. The radial distortion model used here was taken from [4] and its vector form is shown in Equation 4.

$$\mathbf{x}_u = \mathbf{c} + f(r)(\mathbf{x}_d - \mathbf{c}) \tag{4}$$

The undistorted image coordinate, $\mathbf{x}_u$, is computed by adding the coordinates of the centre of radial distortion, **c**, to the coordinates of the corrected x- and y-distances. These corrected distances are calculated by multiplying the x- and y-distances from **c** to the distorted point, $\mathbf{x}_d$, by the correction function, $f(r)$ in Equation 5.

$$f(r) = 1 + k_1 r + k_2 r^3 \tag{5}$$

4. Calibration

A very simple two-step method has been developed and implemented here. In the first step, the camera parameters are approximated using a linear method which ignores non-linear effects such as lens distortion. For this method, a 3D calibration object with known feature coordinates was designed and manufactured (Figure 1). The coordinates of the grid corners were determined on a Coordinate Measurement Machine (CMM). The repeatability of the CMM measurements of the grid was determined to be well below 0.1mm (within 95% confidence). The second step introduces non-linear effects of lens distortion with the model described in section 3.2. These parameters are determined through an optimisation function which minimises the back-projection error of the known 3D coordinates using the initial values from the first step.

4.1. Initialisation of Camera Parameters

If non-linear effects can be ignored, the camera matrix, **P**, can be determined using a simple linear method if the image coordinates and their corresponding world coordinates are known. Used here is the direct linear transform (DLT) method as described by [3], but without the minimisation of geometric error.

For practical implementation of the solution, the linear system is first properly pre-conditioned. This is done by scaling and shifting both the image and world coordinates [3]. After normalisation the DLT algorithm calculates a normalised camera matrix. This matrix is de-normalised to retrieve the final camera matrix.

4.2. Refinement of the Camera Parameters

The values of the camera matrix from the DLT algorithm are now used as initial values for a robust and quickly converging minimisation function. This function must introduce the non-linear lens distortion into the thus far linear camera model.

With the camera matrix and a set of known world-coordinates available, there is an almost intuitive error to minimise: the difference between the calibration-feature coordinates initially extracted from the image and the back-projection of the world-coordinates onto the image plane.

There are different ways in which this error-set can be used to calculate an output for minimisation. Here it has been decided that the mean and standard deviation (SD) of the error-set will be added together and used as the value to be minimised. This has been established through trial and error as the best combination of values. Using the sum of these values gives a low mean value with a higher certainty in the error distribution. Using only the mean usually causes the standard deviation to be slightly higher and vice versa if only the standard deviation is used.

5. Image Processing

A number of image processing techniques are used and combined for the different stages of the process in order to speed it up. The first step is to automate the calibration phase and secondly the measurement phase in terms of automatic target extraction and correspondence matching between image pairs.

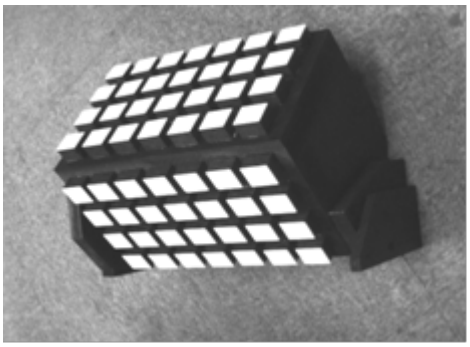


Figure 1: *Calibration Object.*

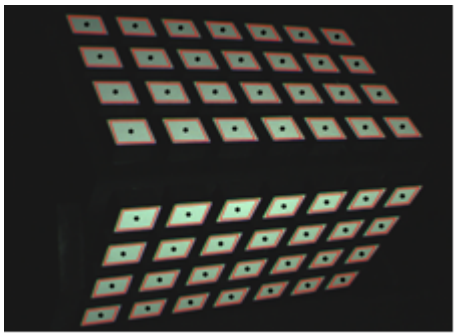


Figure 2: *Square Finding on Calibration Object.*

5.1. Automated Detection of the Calibration Grid

Figure 1 shows the calibration object used in the processing. Two well contrasted views are taken through a number of image processing techniques in order to detect each block on the object and find the matching blocks in the two views.

Figure 2 shows the initial image of the grid from one view. After a number of processing steps each block is detected, its approximate centre found and then fitted with an approximate square.

The final operation for each square is to precisely determine the corner positions as shown in Figure 3. To achieve this, a segment of each edge of a block is extracted using the positions of the approximated squares. Each edge-segment is then processed to detect the position of each edge pixel with high precision using the intensity peaks of the edge segment's derivative image. The derivative image is in turn calculated by convolving the edge segment with a 1D derivative kernel. The edge positions are then used to apply a least-squares line-fitting to that edge. The corners are calculated as the intersection of the lines.

5.2. Rapid Correspondence Matching

Once the calibration stage is complete, the camera matrix and distortion coefficients can be used to determine the 3D coordinates of any two corresponding image coordinates. In order to solve the correspondence matching problem during the measurement stage, both a Digital Light Processing (DLP) projector and a moving laser dot is used to scan objects for measurement. Three methods are proposed for the correspondence matching. Only the first, using a moving laser dot, can be used for practical measurements on non-planar objects. The other two are only for testing and comparing the achievable measurement precision. These last two methods use the DLP projector to project known patterns onto an object, currently only a planar surface.

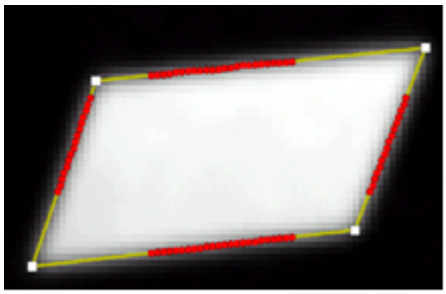


Figure 3: *Block with sub-pixel corners detected.*

5.2.1. Tracking a Moving Laser Dot

For tracking the laser, it is assumed that it is the highest intensity moving object in the image. Two consecutive images from the video stream are subtracted from one another. The position of values that are above a given threshold in the difference image is taken as the approximate position of the laser. A small region of interest (ROI) in one of the original images is taken around the approximate position determined from the difference image. Using a binary threshold on the ROI, the centroid of the black-and-white image is calculated.

5.2.2. Corner Detection Using Square Projections

This method is semi-automated and requires some user-input. Three squares in a single column are projected onto a flat surface using the highest possible intensity of the projector. This is done to get the best contrast between the white squares and the darker surroundings. The algorithm then finds the corners of each square and calculates the correspondences accordingly.

5.2.3. Projected Line-Crossings

This method uses horizontally and vertically projected strips of light. It is also only functional for a flat surface. The derivative images of the vertical and horizontal projections are added together. This last image gives the higher intensity areas where the vertical and horizontal lines cross. A small ROI is now extracted around each of the high intensity spots and the greyscale centroid is calculated. In order to find a corresponding coordinate in the stereo pair, the epipolar geometry of the cameras are used [3]. With only two images, this is an unstable way of searching for correspondences, with many erroneous correspondences being found, especially if there is a large number of crossing points positioned close together.

6. Experiments

6.1. System Description

The system was developed with the Python programming language using OpenCV (www.intel.com/technology/computing/opencv) for many of the image processing functions. Two Firefly MV IEEE 1394 cameras distributed by Point Grey Research are used for image capture. Each camera has a 640x480 resolution, with a frame-rate of 40 frames per second. One camera is greyscale, the other is colour, using a BGR Beyer-pattern

and a edge-sensing colour processing method to create a three-channel colour image.

6.2. Definition of Errors

There are three errors that will be used as outputs for evaluation during the calibration and measurement experiments: back-projection and triangulation error for the calibration stage and deviation from a fitted plane for the measurement stage.

The back-projection error is the same error defined in section 4.2. The triangulation-error is calculated in much the same way as the back-projection error. The distances between the triangulated coordinates of the calibration grid corners and their known coordinates are also accumulated in an error set. Note that these errors can only be computed during the calibration stage because of the known world coordinates. It is one of the main reasons for the use of this type of calibration method: the achievable metric precision for the system can be established directly from calibration.

For the measurement phase, a flat surface will be scanned using all three methods: laser dot, projected squares and projected line-crossings. To evaluate the error, the triangulated surface coordinates will be fitted with a plane using a least-squares method. Each perpendicular distance from the plane (planar deviation) to a triangulated coordinate is accumulated in an error set and evaluated statistically using the standard deviation and visually using the histogram. The mean value is not used, because it is usually very close to zero due to the way the plane is fitted to the triangulated coordinates. Both [1] and [12] use the deviation from a flat surface as an estimate of the noise in the measurement system. The principle is that if a flat surface is reconstructed, any planar deviation indicates the basic measurement error that can be expected in the system.

6.3. Experimental Variables

The variables that will be used as inputs for the calibration experiments are the camera model complexity and the base-to-depth ratio. From the literature it is known that the optimum results should be achieved using a base-to-depth ratio of one and the most complex camera model. In this case it is a model containing two radial distortion coefficients and a drifting centre coordinate. Even though the optimum case is predictable, it will be tested in order to verify results already presented in the literature as well as evaluate the effect on precision for this unique system.

Because all code was custom-developed for this study, the camera model can be adjusted to contain different coefficients for distortion, allowing for the complexity to be increased systematically by adding more of the distortion model coefficients.

It is assumed that the effect of the two variable parameters are independent of one-another. An experimental design testing the interdependence of the variables, such as a full-factorial experimental design, is therefore not used. For each variable, the other parameters are held fixed at their optimum values as given above.

6.4. Results

6.4.1. Model Complexity

The first test uses the DLT method directly with no distortion parameters. The first radial distortion coefficient, k_1, is then introduced, followed by the second, k_2, and finally the drifting radial centre, **c**, is also added. Table 1 and Table 2 give the back-projection and triangulation results of calibration respectively.

The very small difference in the triangulation error between the last two columns of Table 2 indicates that the tangential distortion has a much smaller effect on precision than the radial distortion. To clarify: when adding the drifting centre to the distortion model, the improvement in precision is two orders of magnitude smaller than the improvement gained for adding radial distortion.

When using only one distortion coefficient, the precision is still comparably close to the cases of greater precision. Using only the linear model, however, yields significantly less precise results, even with the iterative improvement that gets rid of statistical outliers.

Table 1: *Back-projection error for different camera model complexities.*

Camera model	Pinhole model	k_1	k_1, k_2	$k_1, k_2, \mathbf{c}$
Colour Camera				
Mean (pixels)	0.353	0.223	0.216	0.206
SD (pixels)	0.191	0.122	0.116	0.114
Mono Camera				
Mean (pixels)	0.431	0.248	0.235	0.231
SD (pixels)	0.237	0.142	0.133	0.129

Table 2: *Triangulation error for different camera model complexities.*

Camera model	Pinhole model	k_1	k_1, k_2	$k_1, k_2, \mathbf{c}$
Mean (mm)	0.266	0.163	0.156	0.153
SD (mm)	0.122	0.079	0.073	0.073
Precision, 95% conf	0.632	0.400	0.375	0.371

6.4.2. Base-to-depth Ratio

For the different test runs, the calibration object remains in the same position while the cameras are moved further from or nearer to one another across the baseline (the line along which the base distance is measured). Table 3 shows the results of the back-projection error for the two approximate base-to-depth ratios, while Table 4 shows the triangulation results.

Even though the 0.5 ratio yields better back-projection results for the colour camera (Table 3), this does not mean it will give better triangulation results. As expected, after five consecutive runs to get the average values presented in the tables, it is clear that for a greater base-to-depth ratio the triangulation is more precise.

Table 3: *Back-projection errors for varying base-to-depth ratios.*

Base/depth ratio	1	0.5
Colour Camera		
Mean (pixels)	0.219	0.214
SD (pixels)	0.116	0.114
Mono Camera		
Mean (pixels)	0.225	0.272
SD (pixels)	0.123	0.138

Table 4: *Triangulation error for varying base-to-depth ratios.*

Base/depth ratio	1	0.5
Mean (mm)	0.157	0.214
SD (mm)	0.077	0.118
Precision, 95% conf (mm)	0.388	0.569

6.4.3. Planar Deviation

The results for the laser tracking method, the square corner matching and the projected line crossings are all presented in Table 5. Four times standard deviation (4 SD) of the error is used as the final output value to evaluate these measurements.

Table 5: *Comparison of matching method precision.*

Matching method	Square corners	Laser	Line crossings
SD (mm)	0.105	0.235	0.263
4 SD (mm)	0.419	0.940	1.052

The best results by far are given by the square corner method. This is understandable, because it extracts the matching coordinates much more precisely than the laser or line-crossing method. The laser-dot's form is not very stable from frame to frame, making the calculation of its centre quite unpredictable. Lastly, the line-crossing method performs worst. Different methods than the weighted centroid calculation might have to be used to achieve greater precision with the line-crossing method.

Figure 4 compares the histograms of the error-sets for each of the methods using the same x-axis scale for comparison. Note that for every method an area of about 210 x 240 mm is used. The spread of the histograms illustrate how the precision differs from method to method.

6.4.4. A Practical Measurement

The laser tracking method is used here to scan the profile of the bottle seen in Figure 5(a), along with different presentations of the 3D data. Even though not the most precise, this matching method is currently the only one capable of measuring more complex surfaces. This measurement is used for a qualitative evaluation only.

The point-cloud of the scanned profile consists of 15790 coordinates accumulated at about 12 fps. Points can of course only be constructed if the laser is visible in both images, which explains the loss of data around sharp bends. Note that the base-to-depth ratio used here is approximately 0.5 in order to increase the field of view common to both cameras.

7. Conclusions

A rapid optical measurement system has been developed and implemented for this project. It is capable of accumulating feature correspondences at 12 points per second with sub-millimetre precision. The precision achieved by calibration is better than 0.4 mm (in the case of the square corner method or better then 1 mm for the laser tracking method) for a 235 x 190 x 95 mm volume, using only one image pair and an image resolution of 640 x 480 pixels.

Most of the processes usually requiring time intensive user interaction in such a system has been automated using different image processing techniques in combination with the right hardware components. This includes the calibration phase as well as three different semi-automatic methods for solving the problem of rapid and precise correspondence matching.

8. References

[1] Chi-Fang, L. and Chih-Yang, L., "A New Approach to High Precision 3-D Measuring System", Image and Vision Computing, 17(11):805–814, 1999.

[2] Fraser, C. S., Shortis, M. R. and Ganci, G., "Multi-sensor System Self-calibration", Videometrics IV, SPIE 2598:2-18, 1995.

[3] Hartley, R. and Zisserman, A., "Multiple View Geometry in Computer Vision" Second Edition, Cambridge University Press, Cambridge, 2003.

[4] Ma, Y., Soatto, S., Košecká, J. and Shankar Shastry, S., "An Invitation to 3-D Vision: From Images to Geometric Models", Springer-Verlag, New York, 2004.

[5] Mallon, J., and Whelan, P. F. "Which Pattern? Biasing Aspects of Planar Calibration Patterns and Detection Methods", Pattern Recognition Letters, 28(8):921-930, 2007.

[6] Muller, N., De Kock E., Van Rooyen, R., Trauernicht, C., "A Stereophotogrammic System to Position Patients for Proton Therapy", 2^{nd} International Conference on Computer Vision Theory and Applications, VISAPP (2):538-541, 2007.

[7] Remondino, F. and Fraser, C, "Digital Camera Calibration Methods: Considerations and Comparisons", International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, Vol. XXXVI:266-272, 2006.

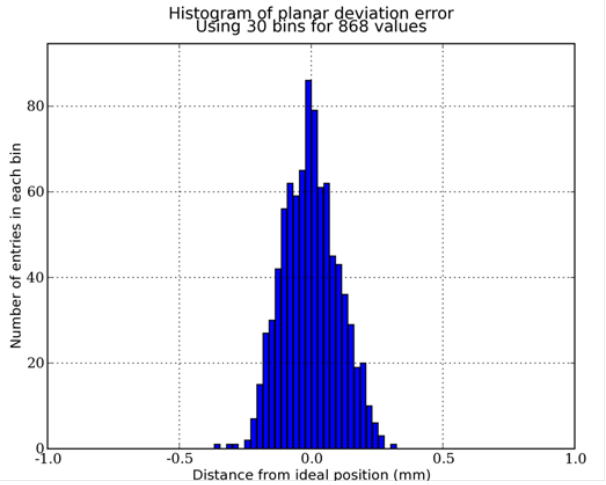
[8] Shortis, M. R., Clarke, T. A. and Short, T., "A Comparison of Some Techniques for the Subpixel Location of Discrete Target Images", Videometrics II, Proc. SPIE 2350:239-250, 1994.

[9] Shortis, M. R., Snow, W. L. and Goad, W. K., "Comparative Geometric Tests of Industrial and Scientific CCD Cameras Using Plumb Line and Test Range Calibrations", International Archives of Photogrammetry and Remote Sensing, 30(5W1):53-59, 1995.

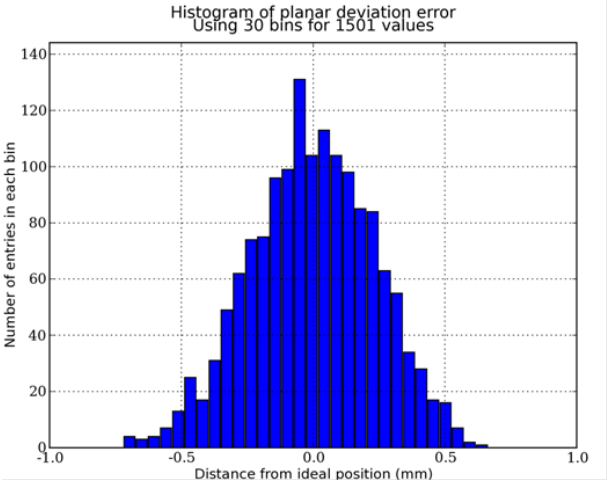
[10] Tsai, R. "A Versatile Camera Calibration Technique for High-accuracy 3-D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses", IEEE Journal of Robotics and Automation, 3(4):323-344, 1987.

[11] Valkenburg, R. J. and McIvor, A. M. "Accurate 3D Measurement Using a Structured Light System", Image and Vision Computing, 16:99-110, 1998.

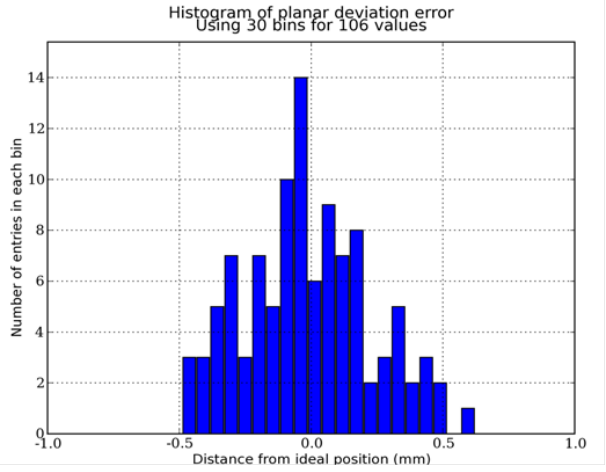
[12] Zhang, S. and Huang, P. S. "High-resolution, Real-time Three-dimensional Shape measurement", Optical Engineering, 45(12):123601, 2006.



(a) Square corner method.



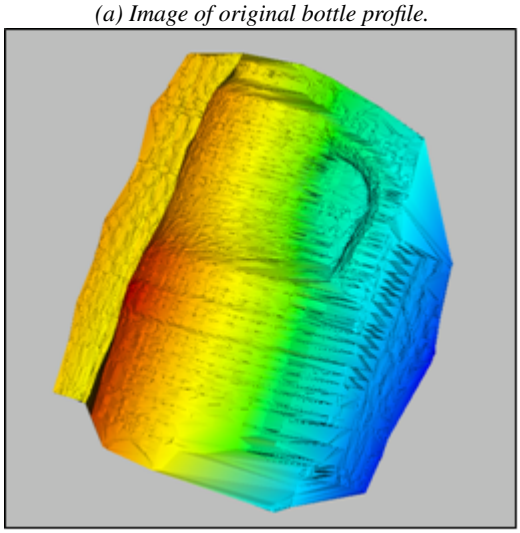
(b) Laser tracking method.



(c) Line crossing method.

Figure 4: *Error histograms for matching methods.*

(a) Image of original bottle profile.



(b) 3D visualisation with a 2D Delaunay filter for surface approximation.

Figure 5: *3D Visualisation of a scanned bottle profile.*

Evaluating Topic Models with Stability

Alta de Waal and Etienne Barnard

Human Language Technologies, Meraka Institute
Faculty of Engineering, North West University
adewaal@csir.co.za, ebarnard@csir.co.za

Abstract

Topic models are unsupervised techniques that extract likely topics from text corpora, by creating probabilistic word-topic and topic-document associations. Evaluation of topic models is a challenge because (a) topic models are often employed on unlabelled data, so that a ground truth does not exist and (b) "soft" (probabilistic) document clusters are created by state-of-the-art topic models, which complicates comparisons even when ground truth labels are available. Perplexity has often been used as a performance measure, but can only be used for fixed vocabularies and feature sets. We turn to an alternative performance measure for topic models – topic stability – and compare its behaviour with perplexity when the vocabulary size is varied. We then evaluate two topic models, LDA and GaP, using topic stability. We also use labelled data to test topic stability on these two models, and show that topic stability has significant potential to evaluate topic models on both labelled and unlabelled corpora.

1. Introduction

The vast amount of electronic text available has stimulated the development of novel processing techniques in order to extract, summarise and understand the information contained therein. Topic modelling is a technique for extracting topics from a text collection by creating probabilistic word-topic and topic-document associations [1]. The most successful topic models are generative models,using the assumption that documents are generated from a mixture of latent topics. A variety of topic models with different generative assumptions about how the documents are generated have been proposed. The documents do not need labels, implying that topic modelling is an unsupervised technique [2]. Unsupervised techniques do not allow for comparison of predicted outcomes with ground truth outcomes; therefore, traditional classification performance metrics cannot be used. Hence, indirect measures of generalization, such as perplexity, are commonly employed as performance measures for topic models. However, current measures suffer from a number of shortcomings. Perplexity, for example, depends on the size of the vocabulary modelled – it can therefore not be used to compare models which use different input feature sets or across different languages. In this paper, we investigate an alternative, namely topic stability, which overcomes some of these deficiencies.

The objective of this study is threefold. First, we compare the behaviour of perplexity and topic stability as two alternative performance metrics for topic models. Secondly, we compare the performance of two topic models, namely Latent Dirichlet Allocation (LDA) and Gamma-Poisson (GaP), using topic stability. Finally, we investigate the relationship between stability and classification accuracy when labels are available. The rest of the paper is outlined as follows. First we put our work in context with the literature. Two topic models, LDA and GaP are described in section 3. Then, we give an overview of perplexity as well as the process to derive topic stability in sections 4 and 5. Two text corpora that we use in experimentation and data preprocessing are described in section 6. The experimental setup and results follow in section 6.1

2. Related Work

In this study we focus on evaluation techniques for unsupervised methods, specifically topic models. In the field of topic modelling, the majority of studies use perplexity as an evaluation method [1, 3, 4]. Rigouste further suggests [1] a document co-occurrence score that is not dependent on feature dimensionality reduction in the way that perplexity is. The document co-occurrence method demands an equal number of topics in two independent sets. The use of this method to evaluate unsupervised algorithms is described in detail in [5]. Information-based measures, such as relative information gain are also used to evaluate topic models, but are difficult to interpret [1, 6].

The concept of topic stability was introduced by Steyvers and Griffiths [2], where stability between aligned topics for two independent topic solutions is measured using the symmetrized Kullback Leibler (KL) distance between the two topic distributions. Classification of documents is another way to test the performance of topic models [3, 7]: the *document* $\times$ *topic* matrix is used as the feature matrix to classify the documents of a labelled corpus using a classifier such as a support vector machine. The topic model is thus measured in terms of the quality of features that it produces.

We focus on comparing perplexity and topic stability as evaluation methods for topic models. Our approach to measuring topic stability is a hybrid between the document co-occurrence of Rigouste and the topic stability of Steyvers and Griffiths. Instead of using the Kullback Leibler divergence between two topic distributions over words (Steyvers and Griffiths), or the document co-occurrence score (Rigouste), we calculate the document correlation between two aligned topics. This allows us to compute a stability measure which is somewhat insensitive to the specific words chosen to describe each topic.

3. Topic Models

For the purpose of topic modelling, a large matrix is constructed from a text corpus (consisting of a number of distinct documents), with rows representing the documents and columns representing the word frequencies (for words in the corpus vocabulary – see figure 1).

In this view, a document is represented as a high-

	word1	word2	word3	. . .	word*n*
doc1	11	5	1		1
doc2	0	1	2		8
⋮					
doc*n*	3	2	0	. . .	9

Figure 1: Document × Word Matrix

dimensional vector, containing the counts of each word in the document. This representation of a text corpus is widely used by a number of clustering techniques, where documents are associated based on their semantic or 'thematic' similarity [1]. 'Thematic' similarity or meaning is extracted by applying statistical computations on the large *document* × *word* matrix [8]. Many approaches to text clustering exist [1, 3, 7, 9], using different sets of assumptions on how the documents in a text corpus are generated. We focus on probabilistic approaches that result in probabilistic topic-document associations [1] by assuming a probabilistic generative process for documents. This section describes two popular topic models with different generative assumptions, namely Latent Dirichlet Allocation (LDA) and Gamma-Poisson (GaP).

3.1. Terminology and notation

We define the following terms and their associated notation:

- A *corpus* is a collection of *M* documents denoted by $\mathcal{C}$ = $\{ \mathbf{w}_1, \mathbf{w}_2 , \ldots, \mathbf{w}_M \}$. The first dimension of the *document* × *word* matrix in figure 1 is of size *M*.
- A *word* w is the basic unit of discrete data.
- A *document* is a sequence or passage of *N* words denoted by $\mathbf{w}_d = \{ w_1, w_2 , \ldots, w_N \}$.
- A *vocabulary* is subset of unique words (denoted by w_l) in the text corpus and indexed by $\{1, \ldots, V\}$. The second dimension of the *document* × *word* matrix is of size *V*.
- We define *T* latent semantic components or *topics* to approximate the *document* × *word* matrix with $T \ll V$.
- The *bag-of-words* representation of a document is the matrix representation illustrated in Fig.1; it neglects word order and only stores the word counts in each document. The quantity $C_{w_i d}$ is the word count of word w_i in document d.

When relating this terminology to machine learning theory, a word is a feature, a bag is a data vector and a document is a sample [7].

3.2. Latent Dirichlet Allocation (LDA)

The basic idea of LDA is that a document is represented as a random mixture over latent topics and a topic is a distribution over words in the vocabulary. LDA assumes that the mixture of topics for a document originates from a Dirichlet distribution and assigns a Dirichlet prior to the mixture of topics for a document. The Dirichlet prior is chosen because of its conjugacy to the multinomial distribution, a property which is crucial in simplifying the statistical inference problem [1, 3]. LDA assumes the following generative process for documents in a corpus $\mathcal{C}$ [3]:

For each document $\mathbf{w} = 1, \ldots, M$

1. Choose $\theta \sim$ Dirichlet(α), θ and α are of dimension *T*.
2. For each word w_i in the document,
   (a) Choose a topic $z_i \sim$ Multinomial(θ).
   (b) Choose a word $w_i \sim$ Multinomial($\boldsymbol{\beta}_{z_i}$). $\boldsymbol{\beta}$ is a *V* × *T* matrix.

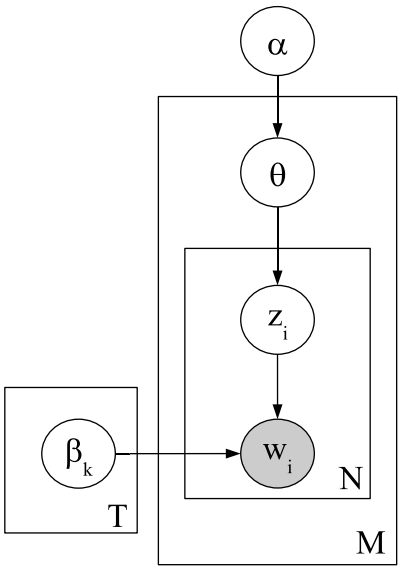


Figure 2: LDA graphical model

Topic models can be described graphically using directed graphs. In such a graphical model, variables are represented by *nodes*, dependencies between variables by *edges* and replications by *plates* [3]. Plates can be nested within one another. Observable nodes are shaded whereas latent variables are unshaded. In figure 2 the plate surrounding θ indicates that θ is a document level variable (with *M* replications) and the plate surrounding z and w indicates that they are word-level variables (with *N* replications). The plate surrounding β indicates that one topic must be chosen from *T* topics. The parameter β indicates which words are important for which topic and θ indicates which topics are important for a particular document [2].

3.3. Gamma-Poisson (GaP)

In [4], Canny introduces the Gamma-Poisson model (GaP), which uses a combination of Gamma and Poisson distributions to infer latent topics. It presents an approximate factorisation of the document × word matrix with matrices β and *X* (see figure 3). The word × topic matrix β represents the global topic information of the corpus $\mathcal{C}$ and each column β_k can be thought of as a probability distribution over the corpus vocabulary for a specific theme *k*. Each column $\mathbf{x}_d$ in the topic × document matrix *X* represents the topic weights for the document *d*. The Gamma distribution generates the topic weights vector $\mathbf{x}_d$ in each document independently. The Poisson distribution generates the vector of observed word

counts **n** from expected counts **y**. The relation between $\mathbf{x}_d$ and **y** is a linear matrix $\mathbf{y} = \beta\mathbf{x}_d$. The topic weights $\mathbf{x}_d$ represent the topic content for each document and encodes the total length of passages about topic *k* in the document. GaP differs from LDA in this regard: LDA chooses topics independently per word in a document, according to the Dirichlet distribution [3], whereas GaP chooses words according to this topic weighting. GaP assumes the following generative process:

For each document $\mathbf{w}_d = 1, \ldots, M$

1. Choose $\mathbf{x}_d \sim$ Gamma(*a*,*b*)
2. For each word $w_i = 1, \ldots, N$

   (a) Generate $n_{w_i} \sim$ Poisson($\beta\mathbf{x}_d$)

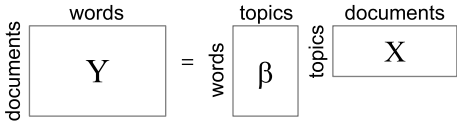


Figure 3: Matrix factorisation of *GaP*

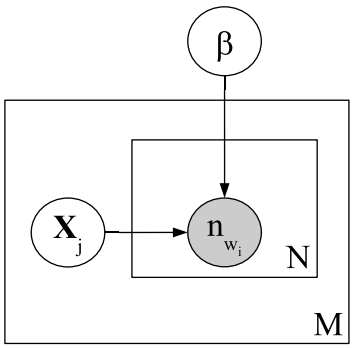


Figure 4: GaP graphical model

The Gamma distribution has two parameters: The first parameter *a* is called the shape parameter and the second parameter *b* is called the scale parameter. The mean value of $\mathbf{x}_k$ is $c_k = a_k b_k$ [4]. The plates in figure 4 further illustrate topics as passages of text in a document, as the $\mathbf{x}_d$ parameter does not reside in the *N*-plate.

4. Perplexity

Perplexity is a standard performance measure used to evaluate models of text data. It measures a model's ability to generalise and predict new documents: the perplexity is an indication of the number of equally likely words that can occur at an arbitrary position in a document. A lower perplexity therefore indicates better generalisation. We calculate perplexity on the test corpus $\mathcal{C}^*$ containing M^* documents as follows:

$$p(\mathcal{C}^*) = \exp\left\{ -\frac{\sum_{d=1}^{M^*} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M^*} N_d} \right\} \qquad (1)$$

Perplexity is therefore the exponent of the mean log-likelihood of words in the test corpus. Consequently, it exhibits similar behaviour to log-likelihood: a reduction in feature dimensionality (in our case, vocabulary) reduces the perplexity, regardless of whether an improved fit to the data has been achieved [1]. This argument will be extended below.

5. Topic Stability

One of the key attributes of a useful topic model is that it should model corpus contents in a stable fashion. That is, useful topics are those that persist despite changes in input representation, model parametrization, etc. We therefore propose topic stability under such perturbations as an alternative performance indicator.

For probabilitstic models such as LDA and GaP, a natural perturbation method presents itself: since these models rely on the iterative optimization of a likelihood function from a random initial condition, they invariably converge to different local solutions from different starting points. We therefore measure stability as the document correlation between two topics that were generated in two independent algorithmic runs from different initial conditions.

In unsupervised learning, there is no way to order or lable topics prior to model estimation [2]. Thus, topics will in general be assigned to unrelated lables in separate runs. When the numbers of topics in the two algorithmic runs are the same, the Hungarian method (also known as Kuhn's method [10], [11]) can be used to align the topics. The Hungarian method is an algorithm for determining a complete weighted bipartite matching that minimises the distance between the two sets in the graph [11], [12]. First, a weight matrix must be set up to indicate the similarities of all pairs resulting from different runs; the algorithm then calculates the optimal overall matching between the two runs.
Two algorithm runs of a topic model can be represented in a bipartite graph (figure 5), where each set represents a run. Once a weight matrix is calculated for the graph, the best matched pairs can be calculated using the Hungarian method. Greedy matching is an alternative method that does not guarantee optimal matching [12].

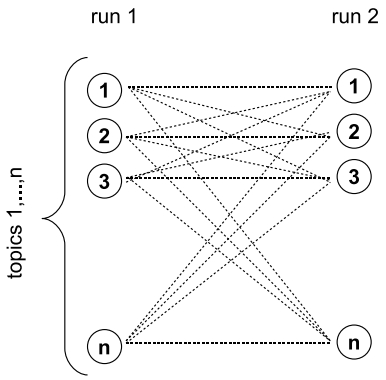


Figure 5: Bipartite graph

The topic stability score is defined as the mean document correlation over all topics, after topics have been aligned with the Hungarian method. The process of obtaining the topic stability scores is described in more detail in the following subsections.

5.1. Weighting

The first step in matching the bipartite graph is to obtain a weighting matrix that represents the weighting of all possible edges between topics in the two runs. Topic models result in two outputs, namely a *topic* $\times$ *document* matrix and a *word* $\times$ *topic* matrix. We use the *topic* $\times$ *document* matrix to calculate the weighting *l*.
Given two algorithmic runs, represented as sets *A* and *B* in a bipartite graph with an equal number of topics *k* in both sets, the weighting between two topics in the respective sets is calculated as follows:

$$l_{ab} = \sum_{d=1}^{M} P(\tau_a|\mathbf{w}_d)P(\Gamma_b|\mathbf{w}_d), \tag{2}$$

where *M* is the number of documents in the corpus, $\mathbf{w}_d$ represents document *d*, and τ_a and Γ_b are the topic distributions from the respective sets *A* and *B*, over document *d*. In order to find the best matched pairs between *A* and *B*, the quantity $\sum_{i=1}^{T} l_{a_i b_i}$ is maximised.
Alternatively, the Kullback-Leibler divergence can be used as a weighting scheme [13].

5.2. Topic Alignment

The Hungarian method searches for the match with maximum weight, i.e., the set of edges that touches each topic in the two sets exactly once, so that $\sum_{i=1}^{T} l_{a_i b_i}$ is maximised [13].
Let $\mathcal{G}$ = (*A*,*B*;*E*) be a bipartite graph, with sets *A* and *B* as in figure 5.The algorithm starts with an empty matched set $\mathcal{M}$. Given the current matching $\mathcal{M}$, $D_{\mathcal{M}}$ is a directed graph where each edge *e* in $\mathcal{M}$ is oriented from *B* to *A* with length $\lambda_e = w_e$. Each edge *e* not in $\mathcal{M}$ is oriented from *A* to *B*, with length $\lambda_e = -w_e$. Let $A_{\mathcal{M}}$ and $B_{\mathcal{M}}$ be the set of topics in *A* and *B*, missed by $\mathcal{M}$. If there is an alternating path from $A_{\mathcal{M}}$ to $B_{\mathcal{M}}$, find the shortest one *P*, and replace $\mathcal{M}$ with the set difference of $\mathcal{M}$ and the edges of *P*. We iterate this process until no alternating path from $A_{\mathcal{M}}$ to $B_{\mathcal{M}}$ can be found.

5.3. Document Correlation

Once the topic alignment is completed, the correlation of documents between matching topics in the respective sets gives a good indication of the model stability. The document correlation is calculated using the *topic* $\times$ *document* matrix where each row represent the topic assignment to documents. Figures 7 and 8 are graphical representations of the document correlation between the topics from the first run and matching topics from the second run, for two different topic models. The dark diagonal line in figure 7 indicates a strong correlation between documents in matching topics.

6. Data Description and Experimental Setup

We used two text collections for the purpose of this research:

- The Cranfield collection [14] of aerodynamic abstracts has 1397 documents. The Cranfield (CRAN) collection is not labelled.
- The *20 Newsgroup* (NEWS) corpus, a large collection of approximately 20,000 newsgroup documents from 20 different newsgroups, collected by Kevin Lang [15]. Each document in this corpus is labelled according to its newsgroup. Cross-posts (duplicates) were removed from the corpus. Some of the newsgroups are closely related, whereas others cover completely unrelated domains.

As part of the data pre-processing step, all non-alphabetic characters were removed as well as words containing only consonants, or words with a sequence of three and more of the same alphabetic character. All words occurring only once were removed, and lastly, documents containing fewer than five words were also removed. From the NEWS corpus, email headings and group information were also removed. After the preprocessing step, the NEWS corpus contained 18705 documents with 52416 unique words and the CRAN corpus 1397 documents with a vocabulary of size 4437.
Both datasets were split into a 80% - 20% training and test set and words occurring only in the test set were ignored.

6.1. Experiments

6.1.1. Perplexity vs Document Correlation

As mentioned in section 4, perplexity as a performance metric is influenced by the feature dimensionality: it invariably improves with a reduction in input dimensionality, regardless of the quality of the fit obtained. To demonstrate this behaviour, we compare perplexity and document correlation against feature dimensionality. Using the CRAN corpus, we gradually reduce the vocabulary by randomly removing columns from the *word* $\times$ *topic* matrix. Thus, the number of vocabulary words is systematically reduced from 100% to 30%, keeping the number of documents the same. The document correlation was calculated on both the training and test set and perplexity was calculated on the test set.

Figure 6 displays the results. The lower graph represents the perplexity scores on the y-axis against the vocabulary dimension on the x-axis. The perplexity scores decrease (i.e. improve) every time dimensionality is reduced, even though there is no reason to believe that the random deletion of words will improve the topic model. The document correlation (upper graph) on the training and test set changes less dramatically, and the correlation on the test set becomes somewhat worse (lower) when words are removed, as would be expected.

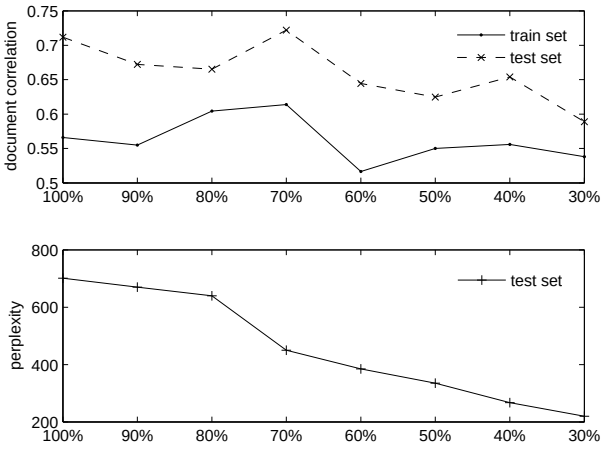


Figure 6: Perplexity vs Document Correlation

6.1.2. LDA vs GaP

In this set of experiments, we compare the performance of the two topic models, LDA and GaP, using document correlation.
In the first experiment, we conduct the straightforward document correlation method as described in section 5 on LDA and GaP, using the full CRAN and NEWS corpora. The resulting document correlation is displayed in table 1. It is clear from the results that LDA has a somewhat more stable topic assignment (indicated by the document correlation) than GaP. Figures 7 and 8 are graphical representations of the topic stability of the two respective models. The dark diagonal line in figure 7 indicates that the aligned topics generally have high document correlation. On the other hand, figure 8 has a less pronounced diagonal line, indicating more instability in topic assignment for the GaP model.

Table 1: *Document correlation for two topic solutions*

	CRAN	NEWS
LDA	0.591	0.757
GAP	0.488	0.527

In the second experiment, instead of performing two independent executions of the algorithm, we run each algorithm once on the labelled NEWS data. We then use the document labels to populate the second set in the bipartite graph. Table 2 displays the results. Although neither LDA nor GaP result in a very good correlation between inferred topics and document labels, LDA has a slightly better correlation than GaP. The relatively low correlation values are not surprising, given that these algorithms make continuous-valued "soft" assignments between documents and topics, whereas the NEWS lables consist of binary assignments. It is encouraging to see that the stability and correlation results nevertheless agree in their preference for the LDA algorithm in this instance.

Table 2: *Document correlation for a topic solution and labelled data*

	NEWS
LDA	0.246
GAP	0.197

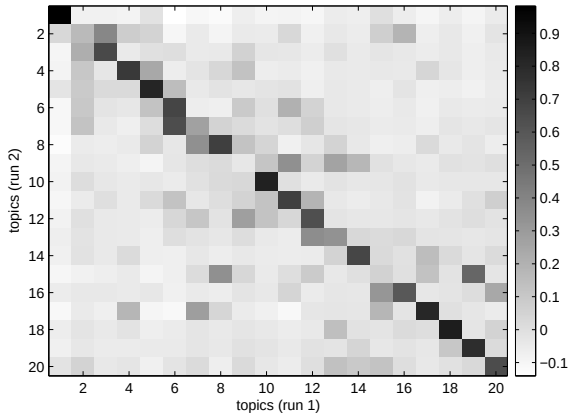


Figure 7: Document correlation matrix for 2 LDA topic solutions

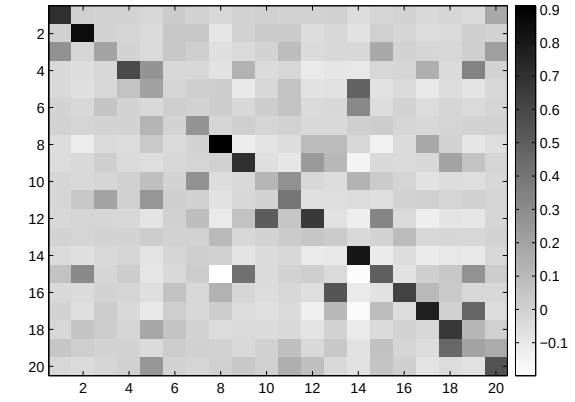


Figure 8: Document correlation matrix for 2 GaP topic solutions

7. Conclusions

The two biggest challenges when measuring the performance of a topic model, are the unsupervised nature of the data and the creation of probabilistic 'soft' document clusters, rather than 'hard' clusters. The most common measure used to evaluate topic models, perplexity, solves these problems by using a word-predictability criterion. However, perplexity values computed with different feature sets are not comparable. We have shown that a modified version of topic stability is a useful alternative performance measure for topic models. At the core of topic stability is the ability to align topics from two independent topic assignments. For this purpose, the Hungarian method guarantees an optimal one-on-one alignment of topics.
We present a topic stability method that uses the average document correlation between topics as the performance metric. Our method does not suffer from the vocabulary dependency of perplexity. We also tested two topic models, LDA and GaP using this method and found that LDA performs better than GaP in terms of topic stability; this agrees with the assessment that arises from the use of document lables when those are available. In future work, we would like to confirm that stability is a useful comparative measure, by studying other forms of perturbation, other corpora, and additional modelling algorithms. We also plan to perform a systematic comparison of our document correlation technique for topic stability with other techniques, such as the document co-occurrence scores used by Rigouste *et al.* [1]. Furthermore, we used the topic $\times$ document matrix to align the topics and indicate the topic stability. This is in contrast with Steyvers and Griffiths [2], who used the topic $\times$ word matrix for the same tasks. More work is needed to understand the respective properties of these two matrices in evaluating the performance of the topic model. (Our preliminary results suggest that word correlation is less reliable than document correlation, since closely related words may take on widely varying weights without affecting document classification.) Finally, we are in the process of implementing a suite of evaluation methods that address different aspects of topic models in order to describe the properties of these models more comprehensively.

8. References

[1] Rigouste, L., Cappé, O., and Yvon, F., "Inference and evaluation of the multinomial mixture model for text clustering." Inf. Process. Manage. 43(5), 1260-1280, 2007.

[2] Steyvers, M. and Griffiths, T., "Probabilistic topic models", In T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning, pp 427-448. Laurence Erlbaum New Jersey, 2007.

[3] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation", Journal of Machine Learning Research. 3:993-1022, 2003.

[4] Canny, J., "GAP: A Factor Model for Discrete Data", ACM Conference on Information Retrieval (SIGIR) Sheffield, England, July 2004.

[5] Lange, T., Roth, V., Braun, M.L. and Buhmann, J.M., "Stability-based validation of clustering solutions.", Neural Computation, 16(6):1299-1323, 2004.

[6] Rigouste, L., Cappé, O., and Yvon, F., "Evaluation of a probabilistic method for unsupervised text clustering" In Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA), Brest, France, 2005.

[7] Buntine, W. and Jakulin, A., "Discrete component analysis", Tech. Rep., Helsinki Institute for Information Technology, July 2005.

[8] Deerwester S., Dumais, S. T Landauer T K., Furnas, G. W and Harshman, R. A., "Indexing by latent semantic analysis", Journal of the Society for Information Science, 41(6), 391-407, 1990.

[9] Hofmann, "Probabilistic latent semantic indexing" in SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp 50-57, ACM Press, 1999.

[10] Kuhn, H., "The Hungarian method for the assignment problem.", Naval Research Logistic Quarterly, 2:83-97, 1955.

[11] Frank, A., "On Kuhn's Hungarian method - a tribute from Hungary.", Tech. Rep. TR-2004-14, Egerváry Research Group, Budapest, 2004.

[12] Rosenbaum, P.R., "Optimal matching for observational studies.". Journal of the American Statistical Association, Vol.84, No.408, pp.1024-1032, 1989.

[13] Rigouste, L., "Méthodes probabilistes pour l'analyse exploratoire de données textuelles.", PhD Thesis, 2007.

[14] Glasgow IDOM - Cranfield collection `http://ir.dcs.gla.ac.uk/resources/test_collections/cran`

[15] Lang, K., "News Weeder: Learning to filter Netnews", Proc. 12th Intl Conf. Machine Learning, San Francisco, 1995

Action Classification using the Average of Pose Changes

Janto F. Dreijer and Ben M. Herbst

Applied Mathematics
Stellenbosch University
Private Bag X1, 7602 Matieland, South Africa
{janto, herbst}@dip.sun.ac.za

Abstract

This article briefly discusses some of our ongoing work on the problem of human action recognition. We evaluate a simple and intuitive technique, based on the changes in human pose, against publicly available behaviour datasets. We achieve results comparable to many other state of the art techniques, while also being much simpler and potentially faster.

1. Introduction

1.1. Problem Statement

Action recognition has interesting applications such as detecting falls [1] and indexing movies [2], and has received increased attention in recent years.

Our goal is to create a system capable of classifying actions in a live video stream, using lightweight techniques. We evaluate a simple and intuitive technique, based on the changes in human pose, against publicly available behavior datasets. We achieve results comparable to many other state of the art techniques, while also being much simpler and potentially faster.

We find that the average of pose changes are surprisingly discriminative for these datasets and conclude that this simple approach is sufficient for action types that have stereotypical poses, at least while the library of poses remain small.

1.2. Related Work

State of the art approaches to action recognition can roughly be grouped into three: Pose transition models, collections of quantized space-time interest points ("bag of features") and template images. Also of interest is the motion of key points, which is often used in gesture recognition applications.

1.2.1. Pose transitions

In this approach actions are regarded as transitions over a sequence of observations of body pose. Individual poses are usually represented as a location in a feature space and a model constructed of the motion through this space. Actions are then classified based on how well they fit the learnt model.

Pose observations have been encoded in terms of their contours [3, 4], optical flow [5, 6], geometric moments [7], and various others. These transitions are then represented in graphical models such as hidden Markov models [6] and Monte Carlo random walks through graphs [4].

It should be noted that separation of the pose from the background is often not ideal, and therefore may introduce significant noise in the pose encodings.

1.2.2. Bag of features

This approach is inspired by recent advances made in recognising generic objects and textual understanding. Actions are seen as collections of specific space-time interest points or cubelets. These techniques involve extracting interesting features from the space-time volume [8, 9, 10]. These discrete feature points are usually summarized in the form of multidimensional histograms. Segments of videos are then compared via a comparison of their histograms [8, 9].

For example, Laptev *et al.* [11] represent interesting points, found with a Harris corner detector, by a Histogram of Gradients descriptor. These features are quantized into words using k-means clustering. Video segments are then classified based on their histogram of words using support vector machines.

Others also consider the space-time volume, but instead try to characterize its properties by using, for example, the solution to the Poisson equation [2].

1.2.3. Motion of points

Lange *et al.* [12] have investigated the human ability to recognise a moving human figure from no more than a few key points. They found a high correlation between their simulation results and psychophysical data. This news might be promising to those that believe the path of various body parts such as hand, head and feet, may be a major component in interpreting human behaviour.

Much work has been done on hand-gesture recogni-

Figure 1: Average of poses (AME) for Weizmann dataset (source: [16])

tion in this regard. Bobick and Wilson [13] represent the trajectory of a hand as a sequence of fuzzy states in a configuration space to capture the repeatability and time-scale variability of a gesture. Nam and Wohn [14] described a hidden Markov based method for recognising the space-time hand movement pattern of various basic gestures by first projecting 3D motion data onto a 2D plane. They then employ a chain encoding scheme and construct a HMM network from simpler left-to-right discrete HMMs.

Song *et al.* [15] have addressed the problem of detecting humans from their motion pattern. They model the joint position and velocity probability density function of triplets of moving features.

1.2.4. Template based

Techniques such as Average Motion Estimates (AMEs, [16]) represent the average of a subject's poses as a single image. Although this is much simpler than the above methods, Lu *et al.* [16] reported surprisingly high performance on the Weizmann dataset. AMEs have, however, only been tested on this relatively simple dataset, partly because poses need to be made translation invariant first.

AMEs emphasize body parts that do not vary (see Figure 1). Indeed, although AMEs represent the motion with regard to the image background, it does not represent the changes in the pose itself.

Davis and Bobick [17] have examined motion-energy images (MEI) and motion-history images (MHI). MEIs are binary images which represent where motion has occurred spatially and MHIs are grayscale images where intensity indicates recent motion. Examples are shown in Figures 2 and 3. MEIs and MHIs are made scale and translation invariant by comparing their Hu moments [18] when classifying actions.

An attractive property of template techniques is that motion can be represented by a single intuitive image. They do, however, also rely on tracking and segmentation of a subject from its background.

1.3. Practical considerations

There are two important factors that have to be taken into account when designing a system capable of performing action recognition on live video streams: amount of processing resources and type of background information

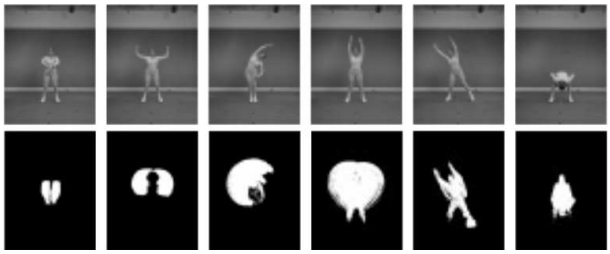


Figure 2: Examples of MEIs for aerobic exercises (source: [17])

Figure 3: Examples of MHIs for waving and crouching (source: [17])

available.

If realtime performance is required, a lightweight strategy has to be used, especially when multiple cameras are involved. Lightweight algorithms allow one to process input from multiple cameras with a single server or push the action recognition algorithm onto smart cameras that typically employ weaker processors.

There are, however, few reports that provide the computational costs involved with existing techniques that would make them applicable to realtime action classification. Those that do report their costs are, in the best cases, in the order of a frame per second for low resolutions on modern consumer hardware [2, 5, 6]. We assume that those that do not report on their efficiency are much slower.

A sophisticated background model is also not always available, depending on the application. It might be good enough to separate subjects, but not to provide error-free body silhouettes. We assume some degree of segmentation of a subject from its background and sufficient inter-subject separation can be obtained.

2. Our Approach

We have investigated various background models, but have decided to use a naive technique to demonstrate our action classifier. Because the datasets (discussed later) contain only one subject we do not require a tracker or inter-subject separation that may be needed in real world applications such as surveillance.

By assuming any motion within the video is primarily of the subject we can use a simple technique to determine the *changes in the subject's pose*, i.e. consecutive frames are subtracted and the difference thresholded:

$$\Delta'_{pose}(n) = |I(n+1) - I(n)| > k \qquad (1)$$

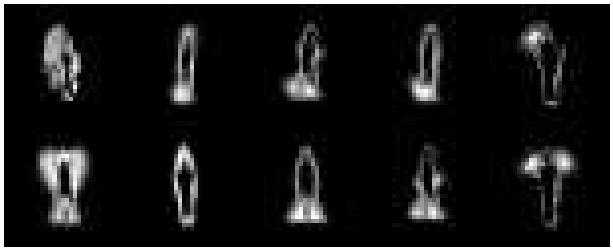


Figure 4: Average of pose changes for Weizmann dataset.

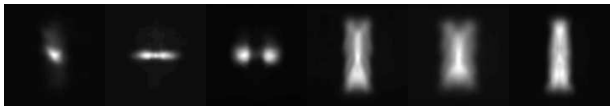


Figure 5: Average of pose changes for KTH dataset.

where $I(n)$ is a specific frame in the sequence and $k = 20/255$ in pixel intensity. We also apply median filtering to remove minor noise.

To obtain a translation and scale invariant representation of the change in pose, we shift and scale the contents of $\Delta'_{pose}(n)$ so its immediate bounding box is centered and encompasses the entire image. We call this new image $\Delta_{pose}(n)$.

By taking the average of *changes* in pose in a video, i.e.

$$T_{video} = \frac{1}{N} \sum_{n=0}^{N-1} \Delta pose(n), \qquad (2)$$

we can obtain Average Pose Changes as shown in Figures 4 and 5.

For classification we determine a template for each video in the testing set. We represent each template as a vector by concatenating its rows. We then estimate a query video's associated action through either a k-nearest neighbour lookup or using a Support Vector Machine (SVM).

Our approach is related to AMEs that represent the average pose (and indirectly including some motion information) and MEIs that are a binary indication of motion. However it is our opinion that it pays to emphasize exactly those body parts that vary and how often they vary.

Note the difference between Figures 1 and 4. In our technique changing body parts are emphasised instead of the static body. We believe that this is an important distinction for two reasons:

- the pose *change* is obtained through simple subtraction and thresholding between frames and is thus, unlike the pose itself, readily available,
- the AME cannot adequately address actions where the average pose may be the same, but the amount of activity of body parts are important.

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	100	0	0	0	0	0	0	0	0	0
jack	0	100	0	0	0	0	0	0	0	0
jump	0	0	89	0	0	0	11	0	0	0
pjump	0	0	0	100	0	0	0	0	0	0
run	0	0	0	0	90	0	10	0	0	0
side	0	0	0	0	0	100	0	0	0	0
skip	0	0	0	0	10	0	90	0	0	0
walk	0	0	0	0	0	0	0	100	0	0
wave1	0	0	0	0	0	0	0	0	100	0
wave2	0	0	0	0	0	0	0	0	0	100

96.9% class average

Table 1: Confusion matrix for Weizmann dataset using pose change templates and nearest neighbour for classification. Provided silhouettes used as pose images.

3. Evaluation

3.1. Datasets

We test the average pose change templates against the Weizmann and KTH datasets.

The Weizmann dataset [2] contains examples of 10 actions performed by 9 subjects giving a total of just more than 90 videos. Segmented translation invariant silhouettes are provided with this dataset. As many have achieved near perfect results on this dataset, we only use it as a demonstration of acceptable results, rather than a measure of relative accuracy.

The KTH dataset [19] contains examples of 6 actions performed by 25 subjects, totaling 593 videos. These videos where designed to contain significant camera motion and zooming effects. Since the backgrounds are relatively uniform, it is easy to isolate the subject from the background.

We used similar cross-validation techniques as used in other studies: leave-one-person-out cross validation (LOOCV) for the Weizmann dataset and three-way cross validation for the KTH videos.

3.2. Discussion

We used the differences in provided foreground as pose changes in one test (Table 1), and immediate frame subtraction in another (Table 2). The near perfect results that were achieved on the Weizmann dataset, are similar to those of the AMEs [16]. Table 2 shows that even without a sophisticated background model, significant performance can still be achieved with an immediate foreground detection scheme.

Tables 3 and 4 show the performance against the KTH dataset using a nearest neighbour classifier and linear SVM. Actions with similar poses (jogging and walking, jogging and running) account for most of the loss in performance. It should be reiterated that no foreground mask was provided with the KTH dataset and hence is to be compared to Table 2 and not 1.

The results of some related studies are reported in Table 5. Note that many of these use different cross vali-

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	89	0	0	0	0	0	0	0	11	0
jack	0	100	0	0	0	0	0	0	0	0
jump	0	0	78	0	0	0	22	0	0	0
pjump	0	0	0	89	0	0	0	0	11	0
run	0	0	0	0	90	0	10	0	0	0
side	0	0	0	0	0	89	11	0	0	0
skip	0	0	10	0	30	10	50	0	0	0
walk	0	0	0	0	0	10	0	90	0	0
wave1	11	0	0	0	0	0	0	11	78	0
wave2	0	0	0	0	0	0	0	0	0	100

85.2% class average

Table 2: Confusion matrix for Weizmann dataset using pose change templates and nearest neighbour for classification. Pose changes extracted from videos.

	boxing	handclapping	handwaving	jogging	running	walking
boxing	85	8	1	2	1	4
handclapping	6	90	4	0	0	0
handwaving	0	4	95	0	0	1
jogging	0	0	0	77	13	9
running	1	0	0	20	78	1
walking	0	0	0	6	2	91

86.0% class average

Table 3: Confusion matrix for KTH dataset using pose change templates and nearest neighbour for classification.

dation techniques and are strictly not comparable. E.g. LOOCV allows one to use approximately three times more videos for training than 3-way split. Still, we can say with reasonable confidence that the accuracy of our approach is comparable to many state-of-the-art algorithms.

A few remarks are in order:

- These datasets, specifically, contain actions mostly differentiable through pose analysis alone. i.e. these actions have stereotypical poses. This is in line with the recent analysis by Weinland and Boyer [23] of the Weizmann dataset.
- Some interesting real world actions are distinguishable through pose analysis alone.
- The datasets do not adequately represent interesting actions that are different primarily due to the speed at which they are executed. Jogging vs running, falling down vs sitting/bending, handing over an item vs punching another person in the stomach, are actions that contain similar poses, but should be treated as different actions due to their speed. This is especially important for applications such as fall detection, as with higher speeds comes higher risk of injury.

3.3. Efficiency

Relatively little attention has been given by others to make existing algorithms work on live video streams. Be-

	boxing	handclapping	handwaving	jogging	running	walking
boxing	88	8	1	0	0	2
handclapping	2	94	4	0	0	0
handwaving	0	9	91	0	0	0
jogging	1	0	0	76	10	13
running	2	0	0	15	79	4
walking	1	1	0	1	0	97

87.3% class average

Table 4: Confusion matrix for KTH dataset using pose change templates and SVM for classification.

method	accuracy
Our method	87.3%
Laptev *et al.* [11]	91.8%
Rodriguez *et al.* [5]	88.7%
Ahmed and Lee [6]	88.3%
Wong [20]	86.6%
Dollar *et al.* [21]	85.9%
Niebles [10]	81.5%
Schuldt [19]	71.7%
Ke *et al.* [22]	63.0%

Table 5: Reported accuracies of related studies on the KTH dataset. Note that many of these use different cross validation techniques and, strictly speaking, are not comparable.

cause we compare the templates directly without extracting any features or moments, we gain a significant advantage in runtime speed. Ahmad and Lee [6], for example require calculating Zernike moments on 160x120 images, which take 0.69-0.82 seconds a frame (approximately 1.4fps) on their 1.7GHz machine.

Our implementation of pose change templates (using a SVM for classification), can currently run at approximately 16fps for a 400x400 video stream on a 3GHz processor.

The effects of tracking and framerate also need to be analysed. Higher frame rates will improve the detection of small motions, but will adversely affect our bounding box model. We therefore plan on using a more complex background model to determine the bounding box.

4. Conclusion

We have investigated a simple method of classifying human actions from a sequence of images.

Even though we have used a very simple approach, our performance is comparable to other existing techniques. At the same time, our approach holds the promise for action recognition requiring few computer resources. Several improvements can still be made to pose change templates, such as a temporal multiscale to detect actions that differ due to their speeds (e.g. running and jogging).

5. References

[1] H. Nait-Charif and S. J. McKenna, “Activity summarisation and fall detection in a supportive home

environment," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, (Washington, DC, USA), pp. 323–326, IEEE Computer Society, 2004.

[2] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, December 2007.

[3] J. Hsieh and Y. Hsu, "Boosted string representation and its application to video surveillance," *Pattern Recognition*, vol. 41, pp. 3078–3091, October 2008.

[4] S. Xiang, F. Nie, Y. Song, and C. Zhang, "Contour graph based human tracking and action sequence recognition," *Pattern Recognition*, vol. 41, no. 12, pp. 3653–3664, 2008.

[5] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *CVPR08*, pp. 1–8, 2008.

[6] M. Ahmad and S.-W. Lee, "Human action recognition using shape and clg-motion flow from multi-view image sequences," *Pattern Recognition*, vol. 41, no. 7, pp. 2237–2252, 2008.

[7] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[8] L. Zelnik-Manor and M. Irani, "Statistical analysis of dynamic actions," *PAMI*, vol. 28, pp. 1530–1535, September 2006.

[9] I. Laptev, B. Caputo, C. Schuldt, and T. Lindeberg, "Local velocity-adapted motion events for spatio-temporal recognition," *Computer Vision and Image Understanding,*, vol. 108, pp. 207–229, 12 2007.

[10] J. Niebles, H. Wang, and F. Li, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, vol. 79, no. 3, 2008.

[11] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[12] J. Lange, K. Georg, and M. Lappe, "Visual perception of biological motion by form: A template-matching analysis," *Journal of Vision*, vol. 6, pp. 836–849, 7 2006.

[13] A. Bobick and A. Wilson, "Configuration states for the representation and recognition of gesture," in *In International Workshop on Automatic Face and Gesture Recognition*, pp. 129–134, 1995.

[14] Y. Nam and K. Wohn, "Recognition of space-time hand gestures using hidden markov models," *In ACM Symposium on Virtual Reality Software and Technology*, 1996.

[15] Y. Song, X. Feng, and P. Perona, "Towards detection of human motion," in *In CVPR*, pp. 810–817, 2000.

[16] L. Wang and D. Suter, "Informative shape representations for human action recognition," *ICPR*, vol. 2, pp. 1266–1269, 2006.

[17] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, (Washington, DC, USA), p. 928, 1997.

[18] M.-K. Hu, "Pattern recognition by moment invariants," in *IRE*, vol. 49, p. 1428, Sep. 1961.

[19] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*, (Washington, DC, USA), pp. 32–36, IEEE Computer Society, 2004.

[20] S. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *ICCV07*, pp. 1–8, 2007.

[21] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, (Washington, DC, USA), pp. 65–72, IEEE Computer Society, 2005.

[22] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *ICCV05*, pp. I: 166–173, 2005.

[23] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Anchorage), pp. 1–7, 2008.

Real-time surface tracking with uncoded structured light

Willie Brink

Council for Scientific and Industrial Research, South Africa
wbrink@csir.co.za

Abstract

A technique for tracking the orientation and position of a near-planar surface in real-time is presented. It uses the principle of structured light to determine 3D surface shape, by the projection of a dense pattern of uniform (uncoded) stripes. In order to estimate pose a weighted least-squares plane is fitted to the 3D data. As an example the technique is applied in a gaming environment where the user moves and rotates his/her open hand in the field of view of the structured light system to control the yaw, pitch and speed of a model aircraft in real-time.

1. Introduction

High-speed 3D shape measurement and tracking has immense potential in a wide variety of application fields [1], ranging from medical imaging and industrial testing to mobile robot navigation and the gaming industry. Such a system may involve two steps: (1) reconstructing or estimating the shape of some dynamic target surface, and (2) approximating and tracking its orientation and position over time.

The first step is usually regarded as a computer vision problem, and addressed through the utilization of a 3D imaging method such as stereo vision [1, 2] or structured light [3, 4]. The latter has the advantage over the former that its output resolution and accuracy can be controlled to a far greater extent. On the other hand, unlike stereo systems that require only two cameras, structured light is an active vision system that has to project a pattern onto a target surface, that has to be opaque and non-specular, in order to measure it. If the specific application permits, however, structured light is ideal for high-speed, low-cost and high-quality 3D surface acquisition.

A structured light system consists typically of a projector (for casting some pattern onto the target surface) and a camera (for capturing the reflecting pattern). The pattern is found in the image and its deformation reveals the shape of the target. The concept evolved from single laser-spot projection, requiring several hours to complete a scan, to the projection of complex patterns that can measure large surface areas in a few milliseconds. For real-time systems it is imperative that sufficient information for full surface reconstruction is gathered within the timebase of a single video frame. A dense pattern of parallel stripes is often chosen as a projection pattern because it can cover a large surface area in a single shot and can produce high-quality 3D models.

The problem associated with multiple stripe patterns, here called the *indexing problem*, amounts to establishing correct correspondences between the projected stripes and those observed in the recorded image. Various different approaches have been proposed, including coding stripes by colour [5], width [6] and time-dependent sequences [4]. These have limitations however: colour cannot be applied consistently to surfaces with weak or ambiguous reflectance, for width coding the resolution is less than for uniform narrow stripes, and time-coded sequences require multiple images over time and are thus not suitable for real-time applications. In light of this we are biased to the use of *uncoded* (i.e. homogeneous) stripes. The indexing problem becomes more involved but can still be solved with a high degree of accuracy [7].

Once a 3D model is constructed the target surface can be tracked over time as it moves, rotates and deforms. We accomplish this by fitting a weighted least-squares plane to the data and tracking its orientation and position in 3D space. As an example for application the technique is implemented in a human-computer interaction (HCI) system where a user controls the yaw, pitch and speed of a model aircraft in real-time, by moving and turning his/her open hand in the field of view of the structured light system.

The remainder of the paper is structured as follows. Section 2 discusses the structured light system we use for 3D reconstruction, a calibration technique and how a surface model can be built. Section 3 explains the method for pose estimation, provides some simplifications in order to speed up the system for real-time applications, and describes the HCI system developed for controlling a model aircraft. The paper is concluded in section 4.

2. Structured light

A schematic diagram of a typical structured light system is shown in Fig. 1. A light stripe is projected onto some target surface, and captured by a camera. The stripe is extracted from the image and, together with the spatial relationship between the projector and camera, reveals the 3D shape of the surface along that stripe. This system uses a single stripe to measure one "slice" at a time, and the target object can now be moved or rotated repeatedly in order to measure other parts that can finally be stitched together into a surface model.

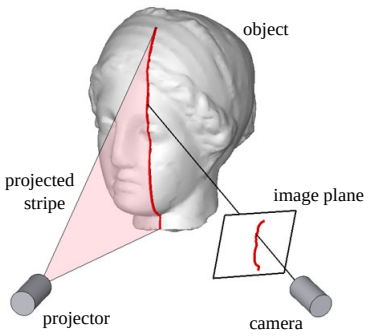


Figure 1: *Schematic diagram of a structured light system where a light source projects a stripe onto a target object's surface and a camera captures the scene.*

The above procedure can be very effective in generating accurate and high-resolution surface models of stationary objects. It is slow however, and does not lend itself to real-time applications where complete surface information has to be acquired at frame rate (i.e. within a few milliseconds).

The current availability of controlled light sources, such as DLP projectors, allows for more complex patterns that increase the surface area measurable per single scan. A dense pattern of parallel stripes is a popular choice [8], but its reconstruction accuracy depends upon establishing correct correspondence between projected and recorded stripes (so that, as shown in Fig. 1, surface points can be determined as intersections with the correct stripe planes). Different methods have been devised to discriminate between captured stripes by, for example, varying colour or width. However we are interested in patterns of homogeneous stripes as they can be applied consistently on surfaces with weak or ambiguous reflection and produce maximal resolution [7].

2.1. Calibration

Calibration of the structured light system is necessary in order to extract accurate metric information. Without calibration a surface can be reconstructed only up to some projective transformation [9] which does not preserve distance.

The calibration procedure should produce the intrinsic parameters (such as the focal lengths, principal points and distortion coefficients) of the camera and projector, as well as the relative pose between the camera and projector (often referred to as the extrinsic parameters). The parameters can be determined once prior to surface acquisition under the assumption that they remain fixed throughout operation.

Figure 2 depicts the layout of the structured light system and our defined coordinate system. The camera is located at the system origin and the projector at point $\mathbf{p}$. The camera ray associated with pixel (r, c) in the image will be denoted by $\mathbf{r}(r, c)$, and the projected plane that produces stripe n will have normal vector $\mathbf{t}(n)$.

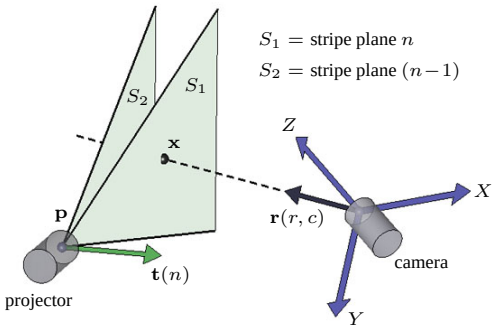


Figure 2: *We let the system coordinate frame (X, Y, Z) coincide with the camera's. The projector is located at point $\mathbf{p}$ and stripe plane n has normal $\mathbf{t}(n)$. The camera ray associated with pixel (r, c) is denoted by $\mathbf{r}(r, c)$.*

The camera's intrinsic parameters can be found through Zhang's method [10] (for example). A printed grid pattern is attached to a planar surface and captured under different orientations. Figure 3a shows one such image. The grid is extracted from the images by a corner detector (e.g. [11]) and the camera's parameters are then estimated using a closed-form solution. The parameters can be refined further through the minimization of a non-linear functional. We let the camera's principal point coincide with the system origin. After calibration any pixel (r, c), i.e. at row r and column c, in a captured image can be associated with a ray $\mathbf{r}(r, c) \in \mathbb{R}^3$. The surface point $\mathbf{x}$ captured at (r, c) is then located at $\lambda\,\mathbf{r}(r, c)$, as can be seen in Fig. 2, for some unknown $\lambda > 0$.

The projector may be calibrated as follows. A grid pattern is projected onto a planar surface within the camera's field of view and an image is recorded. Such an image is shown in Fig. 3b. The location and orientation, in system coordinates, of the planar surface is determined by means of physical markers at known distances apart (the four outer corner markers in Fig. 3b). The locations of the projected grid points are found by computing intersections of this plane with the camera rays corresponding to their pixel coordinates. Repeating the process for different orientations of the plane yields a collection of points in space that can be used to trace back the position $\mathbf{p}$ of the projector. A row (or column) of the projected grid can then be used to determine the relative orientation of the projector. Here it is assumed that the projector counters the effects of radial distortion internally, which is true for most DLP projectors and can be verified easily by observing that the projection of any straight line onto a planar surface remains straight.

The stripe pattern is typically an image consisting of alternating black and white bands of pixel rows, displayed at full-screen resolution. A single stripe can be associated with some index n, directly related to a row (or column) in the pattern, and the calibration parameters can be used to determine a normal vector $\mathbf{t}(n)$ of the plane containing that stripe (see Fig. 2).

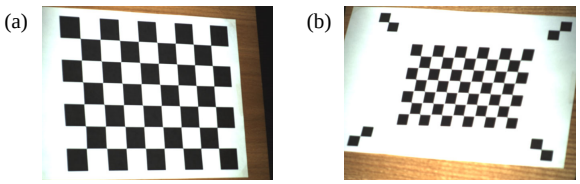


Figure 3: *Examples of captured images for calibrating (a) the camera and (b) the projector. In (b) the four outer corners are physically printed on the plane, and the inner grid is projected by the projector.*

2.2. Indexing uncoded stripes

Before 3D surface points can be determined (as Fig. 2 illustrates, by intersections of camera rays with projection planes) the stripes in the recorded image need to be (1) located and (2) indexed. The latter refers to the assignment of indices to located stripe pixels, which correspond to the individual projected stripes. For example, we may choose to assign an index 0 to the centre stripe and increase (decrease) the index as the row in the projection pattern increases (decreases). An index n can then be mapped to its corresponding stripe plane.

To locate the stripes in the image a simple and quick linear search for local maxima in luminance values across every column can be sufficient, where additional local thresholding can be applied to reduce the effects of noise.

The method we developed for indexing dense patterns of uncoded stripes is described in detail in [7] and a brief summary is given here. The method relies on some geometric constraints derived from the epipolar geometry [12] of the system. Located stripe pixels are grouped according to "left–right" adjacencies

(meant here in an 8-connected sense). Certain smoothness assumptions on the surface are made which then permit the assignment of a single index to all pixels in a connected group. "Up–down" adjacencies between the groups are evaluated in order to build a weighted directed graph. Here edge weights favour clear (and likely to be correct) index transitions between groups. A traversal of a maximum spanning tree of the graph yields an indexing of all the groups relative to an arbitrary starting point. These relative indices differ from their true values by some constant K found, in our implementation, through locating a single reference stripe (noticeable in the image as being slightly darker than the rest) with known index.

The abovementioned smoothness assumption regards elements of an apparently continuous stripe in the image as the same projected stripe. Discontinuities in surface depth of a certain critical size can nullify this assumption, introduce indexing errors and necessitate stripe coding [7]. But we aim to reconstruct near-planar surfaces that should not contain such discontinuities and, because the indexing method seeks a solution that agrees with stripe connections and transitions correctly on a global scale, errors tend to be small, localized and have little influence on the pose of the reconstructed model.

Figure 4 shows an example image of a statue's head, and a close-up in which the dense stripe pattern can be seen. The located stripe pixels are shown on the right, coloured according to assigned indices (note that the colour sequence repeats).

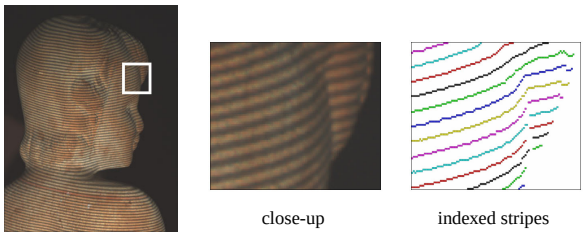


Figure 4: *A captured image and a close-up, in which the dense stripe pattern can be seen. The stripes need to be located and indexed, as shown on the right, for reconstruction.*

2.3. 3D reconstruction

The above procedure yields a collection of indexed stripe pixels, each having pixel coordinates (r_i, c_i) and an index n_i. We determine each one's associated surface point $\mathbf{x}_i$ as the intersection of the camera ray $\mathbf{r}(r_i, c_i)$ with stripe plane n_i having normal $\mathbf{t}(n_i)$. Hence

$$\mathbf{x}_i = \left(\frac{\mathbf{t}(n_i) \cdot \mathbf{p}}{\mathbf{t}(n_i) \cdot \mathbf{r}(r_i, c_i)} \right) \mathbf{r}(r_i, c_i), \qquad (1)$$

where $\mathbf{p}$ indicates the position of the projector (see Fig. 2).

3D points can be calculated for all indexed stripe pixels and a piecewise linear surface can then be constructed in the following way. Suppose the recorded image has N columns, let P indicate the total number of indexed stripes and $n_{\min}$ the smallest index found. Let $\mathbf{x}(i, j)$ indicate the surface point on stripe $n_i = i + n_{\min} - 1$ captured on image column j (we know from the epipolar constraints that an index can exist at most once in each image column). If such a point does not exist, i.e. if stripe n_i is not present in column j, $\mathbf{x}(i, j)$ is flagged as "missing". $\mathcal{S}$ is then defined to be the quadrilateral mesh consisting of all 4-sided polygons with vertices $\mathbf{x}(i, j)$, $\mathbf{x}(i, j+1)$, $\mathbf{x}(i+1, j+1)$ and $\mathbf{x}(i+1, j)$, with $i = 1, \ldots, P-1$ and $j = 1, \ldots, N-1$. Quadrilaterals containing missing points as vertices are removed from $\mathcal{S}$.

An example of a 3D surface reconstructed by this method is shown in Fig. 5, from the source image shown on the left. The model is shown from different viewpoints and consists of roughly 100,000 vertices.

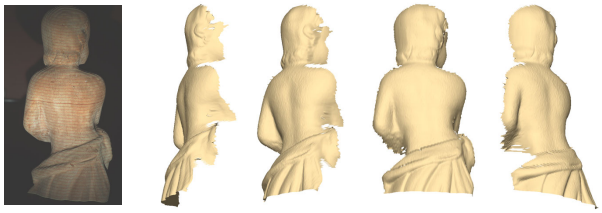


Figure 5: *The surface reconstructed from the single image on the left, shown here from 4 different viewpoints.*

Note that the resolution (or smallest measurable interval) of the system depends on the resolution of the image, as stripes are found only to pixel accuracy. A so-called sub-pixel estimator can be applied [13] which has been shown to greatly improve the resolution. Further surface processing such as hole-filling, smoothing, subdivision, etc., can also be performed to enhance the visual appearance of the 3D model.

3. Real-time surface tracking

The aim of this work is to track the position and orientation, which we collectively refer to as *pose*, of a near-planar surface such as an open hand. Figure 6 shows an image captured of a hand on the left and its reconstruction on the right.

3.1. Pose estimation

We opt to approximate the pose of the hand by a weighted least-squares plane. The palm should influence the orientation of such a plane more than the fingertips, hence weights are assigned according to the distance from the mean of the collection of points (assumed to be close to the centre of the palm). Suppose $\mathbf{x}_i = (x_i, y_i, z_i)$, $i = 1, \ldots, n$ are the 3D surface points, and let $\mathbf{m}$ denote their mean. The Euclidean distance between $\mathbf{x}_i$ and $\mathbf{m}$ is denoted by d_i. We scale these distances to the interval $(0, 1)$ and subtract from 1 to arrive at weights

$$w_i = 1 - \frac{d_i - d_{\min}}{\max_j\{d_j - d_{\min}\}}, \quad \text{with } d_{\min} = \min_j\{d_j\}. \qquad (2)$$

The equation of a plane is $ax + by + cz + d = 0$ which, assuming that $c \neq 0$, may be rewritten as $\alpha x + \beta y + \gamma = z$. The unknowns α, β and γ can be found through a least-squares solution of the overdetermined system

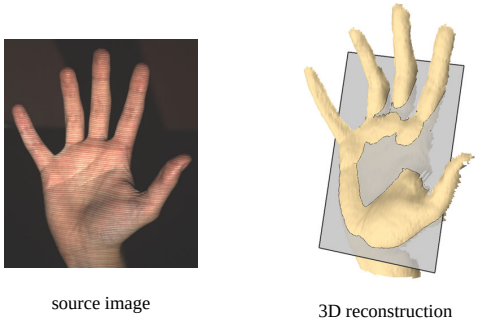
$$W \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = W \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \qquad (3)$$

with W the $n \times n$ diagonal matrix containing w_i, $i = 1, \ldots, n$ on the diagonal. Note that the contribution of each point is actually weighed by w_i^2 in the sum minimized by the least-squares solution to the above system.

The plane can now be defined completely by its normal vector $\mathbf{n}$ and a point $\mathbf{q}$ on it, where

$$\mathbf{n} = \begin{bmatrix} \alpha \\ \beta \\ -1 \end{bmatrix} \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} m_x \\ m_y \\ \alpha m_x + \beta m_y + \gamma \end{bmatrix}. \qquad (4)$$

Here m_x and m_y indicate the x- and y-coordinates of $\mathbf{m}$ respectively and, as it stands, $\mathbf{n}$ is not normalized. Note that the assumption that the z-component of $\mathbf{n}$ will never be zero is valid because the structured light system cannot reconstruct surfaces parallel to the Z-axis. An example of a weighted least-squares plane is shown in Fig. 6 (right).



source image 3D reconstruction

Figure 6: *The pose of the hand is approximated by a plane in a weighted least-squares sense.*

The plane estimates one translational and two rotational degrees of freedom (DOF) of the pose of the hand. We aim to extract this information and apply it to some computerized model in real-time. The structured light system used for the examples in Fig. 5 and Fig. 6 utilizes a colour camera with resolution 1024×768, and projects roughly 120 stripes across the surface of the hand (depending, of course, on the hand's distance from the projector). It yields a 3D model consisting on average of about $50,000$ points. The time required for the entire process by our implementation in C, executed on a Pentium 1.79 GHz dual-core processor with 2GB RAM, is given in Table 1 (the process of capturing an image involves sending a request to the camera, waiting and receiving camera output).

process	time (s)	Hz
capture image	0.018	56
locate stripes	0.053	19
index stripes	0.087	11
calculate 3D points	0.028	36
determine pose	0.019	53
TOTAL	0.205	5

Table 1: *The execution times required by the various processes in estimating the pose of an open hand.*

3.2. Simplifying the system for real-time

Although fast, particularly when considering the size of the generated models, 5 frames per second (fps) is hardly real-time. Since we are chiefly interested in the pose of the hand, and in an effort to speed up the process, the number of points in the output data can be reduced drastically. A simple test shows that a reduced version of the 3D model in Fig. 6, consisting of only about 2,500 points instead of 50,000, produces a least-squares plane very close to the one fitted to the original data. The normals differ by only about 0.01 degrees and the points determined through (4) are only 0.3*mm* apart.

Our simplified system captures 640×480 images, projects roughly 35 stripes across the surface of the hand and generates about $2,500$ points. Each image now contains almost 0.4 times as many pixels and the number of stripes that need to be located and indexed is about a third of the number in the original system. All the algorithms can be implemented in linear time complexity, except for solving (3) which is quadratic in n. The execution times achieved by the simplified system are given in Table 2.

process	time (s)	Hz
capture image	0.011	91
locate stripes	0.012	83
index stripes	0.016	63
calculate 3D points	0.006	167
determine pose	0.004	250
TOTAL	0.049	20

Table 2: *The execution times required for the simplified system to estimate the pose of an open hand. Experimental tests show that the result is insignificantly different from that of the original system.*

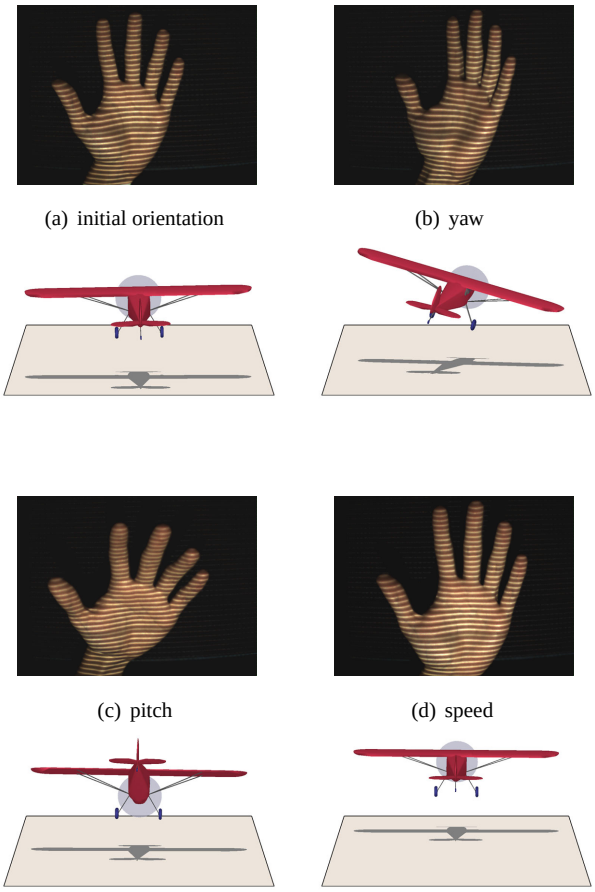
This technique for estimating the 3D pose of a hand runs at about 20 fps which can be sufficient for real-time systems. To demonstrate a possible application we applied the resulting poses to a model aircraft in a gaming environment; see Fig. 7. The camera and projector are positioned next to the computer screen and face towards the user. Note therefore that the hand in the images captured (and shown in the figure) are the left hand of the user, and the pictures of the model aircraft are shown as the user would view them on the screen.

A reference plane is defined for the aircraft which is parallel to the X–Y plane in system coordinates when the aircraft faces forward (away from the user, e.g. Fig. 7a). The differences in rotation and translation from the previous frame are determined, the plane is transformed at the current frame to coincide with the estimated pose of the hand, and the aircraft is rotated and moved accordingly. In this manner two rotational DOF can be controlled (Fig. 7b and c), and one translation DOF that may be used to control the forward speed (Fig. 7d).

The three DOF mentioned can be sufficient for this application, and provide the user with a immersive sense of control. The movements of the aircraft appear continuous and smooth, due to the system operating at approximately 20 fps (the time required to apply the calculated pose to the aircraft and render the result is short relative to those listed in Table 2).

The two remaining translational DOF (those that allow two-dimensional movement within the plane) could be incorporated as well by simply tracking the bounding box of the hand in the image. These could then be used to move the aircraft left, right, up and down. The remaining rotational DOF (allowing in-plane rotation) is slightly more difficult and would require, for example, some analysis of the actual 2D shape of the hand and how it rotates about the normal of the least-squares plane from one frame to the next.

Note that in the current system we do not attempt to first segment the hand from background regions of the image be-



(a) initial orientation (b) yaw

(c) pitch (d) speed

Figure 7: *The position and orientation of a hand is tracked in real-time and shown here to control the yaw, pitch and speed of a model aircraft. Note that the images of the hand are shown from the camera's point of view and those of the model aircraft from the user's.*

cause (it is assumed that) the background is far away. Stripes projected on background objects should therefore be out of focus so that they are not detected in the image and hence will not appear in the reconstructed model.

Also, stripes are currently located and indexed in every image frame completely independently. A further speed-up may be possibly if some temporal consistency constraint is incorporated, e.g. one that in some way bounds the spatial displacement of a particular stripe from one frame to the next.

4. Conclusions

We presented a system for the 3D reconstruction and pose tracking of a near-planar surface in real-time (around 20 frames per second). The system incorporates a simple structured light approach with a pattern of uncoded stripes to acquire 3D surface data, and fits a weighted least-squares plane in order to estimate position and orientation. A possible use was demonstrated in the form of an HCI system where the user controls 3 degrees of freedom of a model aircraft by moving and rotating an open hand in front of the structured light device.

The technique can be applied to various other problems such as the real-time capturing of non-rigid surfaces for deformation analysis or animation purposes, or the data generation and analysis for dynamic scene understanding.

5. Acknowledgments

The author thanks the Council for Scientific and Industrial Research (CSIR) and the Mobile Intelligent Autonomous Systems research group for their support of this work, and Alan Robinson for his useful contributions to the development of the structured light system.

6. References

[1] N.A. Ramey, J.J. Corso, W.W. Lau, D. Burschka and G.D. Hager, "Real-time 3D surface tracking and its applications", Computer Vision and Pattern Recognition Workshop, pp. 34–41, 2004.

[2] S. de Roeck, N. Cornelis and L.J. van Gool, "Augmenting fast stereo with silhouette constraints for dynamic 3D capture", International Conference on Pattern Recognition, pp. 131–134, 2006.

[3] S.Y. Chen, Y.F. Li and J. Zhang, "Vision processing for realtime 3-D data acquisition based on coded structured light", IEEE Transactions on Image Processing, 17(2):167–176, 2008.

[4] S. Zhang and P.S. Huang, "High-resolution, real-time 3-dimensional shape measurement", Optical Engineering, 45(12):123601, 2006.

[5] L. Zhang, B. Curless and S.M. Seitz, "Rapid shape acquisition using color structured light and multi-pass dynamic programming", International Symposium on 3D Data Processing, Visualization and Transmission, pp. 24–36, 2002.

[6] C. Beumier and M. Acheroy, "3D facial surface acquisition by structured light", International Workshop on Synthetic Natural Hybrid Coding and 3D Imaging, pp. 103–106, 1999.

[7] W. Brink, A. Robinson and M. Rodrigues, "Indexing uncoded stripe patterns in structured light systems by maximum spanning trees", British Machine Vision Conference, pp. 575–584, 2008.

[8] J. Salvi, J. Pagés and J. Batlle, "Pattern codification strategies in structured light systems", Pattern Recognition, 37(4):827–849, 2004.

[9] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", European Conference on Computer Vision, pp. 563–578, 1992.

[10] Z. Zhang, "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

[11] C. Harris and M.J. Stephens, "A combined corner and edge detector", Alvey Vision Conference, pp. 147–152, 1988.

[12] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"', 2nd ed., Cambridge University Press, Cambridge, 2003.

[13] R.B. Fisher and D.K. Naidu, "A comparison of algorithms for subpixel peak detection", Advances in Image Processing, Multimedia and Machine Vision, Springer-Verlag, Heidelberg, 1996.

Fiducial-based monocular 3D displacement measurement of breakwater armour unit models

R. Vieira[†], *F. van den Bergh*[†], *B.J. van Wyk*[‡]

[†]Meraka Institute
CSIR, PO Box 395, Pretoria
South Africa, 0001
{rvieira, fvdbergh}@csir.co.za

[‡]French South African Technical Institute of Electronics
Tshwane University of Technology, Pretoria
South Africa, 0001
vanwykb@tut.ac.za

Abstract

This paper presents a fiducial-based approach to monitoring the movement of breakwater armour units in a model hall environment. Target symbols with known dimensions are attached to the physical models, allowing the recovery of three-dimensional positional information using only a single camera. The before-change and after-change fiducial positions are matched optimally, allowing the recovery of three-dimensional movement vectors representing the shifts in the positions of the physical models. Experimental results show that sub-millimeter accuracies are possible using 6-megapixel images of an A4-scale scene.

1. Introduction

Most harbours are occasionally subjected to storms powerful enough to damage infrastructure and ships, unless some preventative measures are taken. To protect the harbour infrastructure, arrays of armour units are used to absorb wave energy and reduce overtopping. The armour unit arrays must dissipate as much energy as possible, without deforming or suffering damage to the armour units themselves. This can be achieved by using armour units with an interlocking structure, such as the *dolos*, invented in East London in the 1963 [1].

Currently, the most effective method of validating the design of armoured breakwater structures is by building and evaluating physical scale models. A scale model of an entire harbour is constructed, complete with a sea floor modelled from bathymetry data. Wave generators are used to simulate wave conditions corresponding to 1000-, 100-, and 50-year storms. A successful armoured breakwater design will suffer little or no damage, measured in the model hall by assessing the magnitude of shifts in the positions of the scale model armour units. The CSIR's model hall facility, located in Stellenbosch, routinely conducts tests of this nature. Owing to the time-consuming nature of the physical modelling process, efforts are under way to develop computer simulations to assist with the validation of harbour designs [2].

Although physical models are considered to be an effective method of determining the stability of an armoured breakwater structure, the method used to evaluate the impact of simulated storm conditions is often subjective. Current methods of assessing damage to a breakwater include a visual comparison of a pair of before-simulation and after-simulation images. By displaying the *before* and *after* images in rapid succession, the changed regions of the scene appear to flicker — this technique is often referred to as *flicker animation* [3]. An operator will manually draw lines, representing movement vectors, on top of the flicker animation. A final assessment of the degree of damage that a breakwater structure has suffered during a simulation can then be estimated from the number and magnitude of the displacement vectors.

In addition to the subjective nature of the flicker technique measurements, they are inherently restricted to two dimensions. One potential method of improving the accuracy of the measurement of the movement of armour unit models is to attach accelerometers to the physical models. This, however, may restrict the movement of the models, and could become prohibitively expensive for larger tests involving many hundreds of armour units.

This paper proposes a different, cost effective method of measuring the movement of armour units using monocular machine vision techniques. Printed fiducial patterns are attached to the physical scale models, enabling an automated system to track the three-dimensional displacement of the models with millimeter accuracy.

Section 2 briefly discusses some recent applications of fiducials, followed by a description of the proposed system in Section 3. An empirical analysis of the positional accuracy of the system is presented in Section 4. Section 5 discusses how the fiducial method has been applied to compute the displacement of armour unit models, followed by some suggestions for future research in Section 6.

2. Background

2.1. Fiducial patterns

Fiducials are special geometric patterns that are used as reference points in machine vision systems. They have long been used in applications such as printed circuit board alignment, but have recently gained popularity in *augmented reality* applications. In these applications, the fiducials are used to define navigation reference points in a three-dimensional space; for example, Naimark and Foxlin demonstrated the use of fiducial patterns to mark up entire buildings [4].

The intended application of a fiducial has a significant impact on its design: some fiducials patterns are optimised to have a very large number of codes, while others are designed to provide very high positional accuracy. Some of the earlier applications in circuit alignment relied on very simple fiducial patterns such as squares, diamonds or circles. Owing to their simplicity, these fiducials could not encode a large number of different codes, but they were simple to detect. Amongst these early fiducials, Bose and Amir showed that circular fiducials produced significantly smaller positional errors compared to squares or

diamonds [5].

Owen *et al.* proposed a square fiducial based on Discrete Cosine Transform (DCT) basis images [6]. The fiducial is identified by a square black border surrounding the DCT-coded interior. The interior of the fiducial is represented as a 16×16 block, meaning that under ideal conditions, the fiducial can still be identified when the sampled image of the fiducial is only around 16×16 pixels in size. The advantage of the DCT-coded interior is that it provides a medium-sized coding space of around 200 codes, while maintaining robustness to noise. Another augmented reality fiducial system built around square patterns was proposed by Rekimoto and Ayatsuka [7]. Their CyberCode fiducial pattern more closely resembles a 2D barcode, and can encode 24 bits of information, after error correction. Unfortunately, Rekimoto and Ayatsuka do not elaborate on the minimum image size required or maximum viewing angle allowed for successful identification.

Despite the success of such square-based fiducial patterns, the circular patterns remain popular. Recent examples include the code proposed by Naimark and Foxlin, which can encode $2^{15} = 32768$ different codes [4]. The minimum image size required for successful fiducial identification reported by Naimark and Foxlin was 16×16 pixels; no figures were reported on the maximum allowed viewing angle. Another circular fiducial was proposed by López de Ipiña *et al.* for use in their TRIP location system [8]. The TRIP code consists of a "bull's eye" pattern in the centre, which is used to identify potential fiducials in the image. Two concentric tracks surround the central bull's eye, in which a sector-based scheme with three discrete sector sizes is used to encode the code value of a fiducial. This design allows for up to $3^9 - 1 = 19682$ different code values. López de Ipiña *et al.* report that the fiducials can be successfully identified provided that the pattern is at least 35×35 pixels in size, and the angle between the viewing direction and the surface normal is less than 70°.

2.2. Correspondence problem

The correspondence problem can be defined as the problem of finding the optimal association between two sets of features, allowing for the possibility that either set may contain elements that have no corresponding element in the other set. To calculate the movement vectors of fiducial patterns from a before- and after-simulation image pair, a similar correspondence problem arises: Given a fiducial pattern in the *before*-simulation image, find the most likely matching fiducial pattern in the *after*-simulation image.

In the simplest case, where only a single fiducial code pattern is attached to all the armour unit models, this would reduce to the problem of finding the closest point P_j (corresponding to the centroid of a fiducial pattern) in the *after* image corresponding to the point P_i in the *before* image. If more than one fiducial code is used, then this problem is constrained so that points may only be matched if their codes agree.

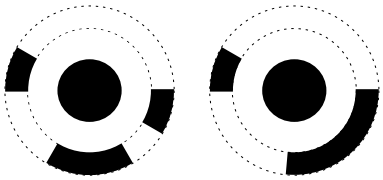
A simple algorithm that could be used to solve this type of correspondence problem is the Iterated Closest Point (ICP) method [9]. This algorithm computes the distances between all points, keeping only distances below a specified threshold. After rejecting outliers, a rigid motion transform is then computed on the remaining points. The algorithm iterates these steps until convergence. After the two sets have been aligned with the transform, the closest point pairs could be used as the correspondence map.

A more robust method was introduced by Maciel and Costeira [10]. Consider that the mapping of points in set $\mathbf{X}$ onto the points in set $\mathbf{Y}$ can be represented as a *partial permutation matrix* $\mathbf{P}$. This matrix resembles an identity matrix, with some of its rows exchanged, and potentially with some of the rows or columns set to zero. Finding the best mapping between $\mathbf{X}$ and $\mathbf{Y}$ can then be expressed as

$$\begin{aligned} \mathbf{P}^* &= \arg\min_{\mathbf{P}} J(\mathbf{X}, \mathbf{Y}, \mathbf{P}) \\ & \quad s.t.\ \mathbf{P} \in \mathcal{P}_p(p_1, p_2). \end{aligned}$$

where J represents a metric that compares elements from $\mathbf{X}$ and $\mathbf{Y}$, and $\mathcal{P}_p(p_1, p_2)$ represents the space of all partial permutation matrices, *i.e.*, matrices containing at most one "1" in each row or column.

Solving this integer optimisation problem is hard; Maciel and Costeira proposed a method that maps the integer optimisation problem to a dual problem on a continuous domain, where is can be solved efficiently using *concave programming* methods. If the metric J is linear, then this approach is guaranteed to find the globally optimal solution $\mathbf{P}^*$.



(a) (b)

Figure 1: Sample fiducial patterns.

Figure 2: Sample image of two dolos models with various fiducial patterns attached.

3. System overview

Based on the literature presented in Section 2.1, a simplified circular fiducial pattern, roughly similar to the one proposed by López de Ipiña *et al.* [8] was selected. Figure 1 presents some examples of this fiducial pattern. This particular fiducial has a fairly large white ring between the central dot and the outer coding track to reduce aliasing problems when viewing the fiducial from a direction with an angle of more than 70° with respect to the surface normal.

These fiducials were scaled so that the diameter of the outer track was 7.1 mm in size to match the scale of the physical models, printed at 600 DPI using a standard laser printer, and fixed to the physical models as illustrated in Figure 2.

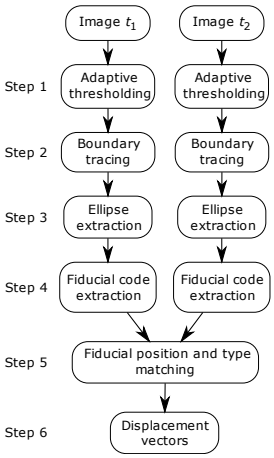


Figure 3: System overview

A system overview diagram is presented in Figure 3. The following algorithms were used to perform each of the steps:

1. The black regions in the image are identified by performing adaptive thresholding using the method of Bradley and Roth [11].
2. The pixel boundaries of objects are extracted using the component-labeling algorithm of Chang *et al.* [12]. Objects with very short boundaries (fewer than 10 pixels) or very long boundaries are discarded. This step produces all the boundaries of potential ellipses, corresponding to the central circle in the fiducial pattern.
3. Ellipse extraction is performed using the method of Ouellet and Hebert [13]. Note that the object boundaries extracted in step 2 are only used to seed the ellipse extraction algorithm; the algorithm derives ellipse parameters directly from the image gradient, producing significantly more accurate estimates of ellipse parameters compared to conventional algorithms. Objects that are unlikely to be ellipses are discarded by testing against conservative thresholds on various ellipse properties.
4. The fiducial code pattern is extracted by sampling the thresholded image along an elliptical path around the central dot of the candidate fiducial. The extracted signature is compared (using the Hamming distance metric) to a template library of known fiducial codes. Once a fiducial pattern is successfully identified, its fiducial code identifier and 3D coordinates are recorded. The 3D coordinates are determined directly from the ellipse parameters using the method proposed for the TRIP system [8].
5. The before-simulation (t_1) and after-simulation (t_2) images are processed with steps 1–5 to obtain the coordinates and identifiers of the fiducials in both images. The algorithm of Maciel and Costeira [10] is used to find the optimal association between fiducials from image t_1 and image t_2, producing as output the correspondence mapping.
6. Using the 3D coordinates of the fiducial patterns and the correspondence mapping, the displacement vectors of each of the matched fiducials is computed. For the purposes of this paper, the displacement vectors are merely visualised, but subsequent processing of the displacement vectors may be used to estimate the degree of damage to a breakwater armour unit array following a wave simulation.

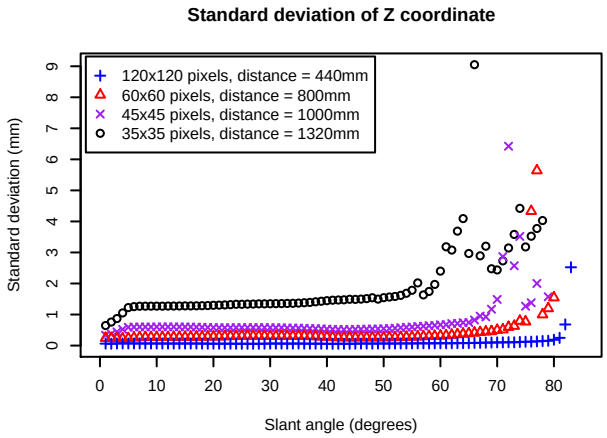


Figure 4: Standard deviation of z-coordinates, computed from degraded synthetic images (blur $\sigma = 0.5$, noise $\sigma = 1\%$).

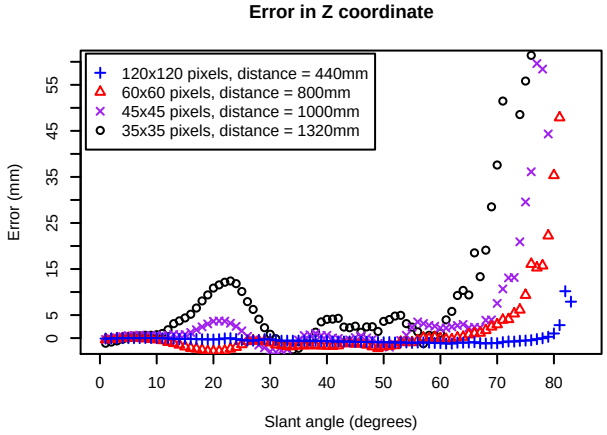


Figure 5: Z-coordinate error, computed from degraded synthetic images (blur $\sigma = 0.5$, noise $\sigma = 1\%$).

4. Performance evaluation of fiducials

The projection of a circle in world space onto the image plane is an ellipse, provided that a distortion-free pinhole camera model is assumed. A real lens will introduce some distortion, but because the lens distortion function typically varies slowly relative to the size of a fiducial, one can assume that the projection of a circle can be approximated with an ellipse.

A direct relationship exists between the imaged size of a fiducial, such as the one shown in Figure 1, and the accuracy with which its 3D position can be determined. The approximate ellipse formed by the boundary between the central black dot and the surrounding white ring is used to estimate the pose of the projected circle that it represents. Three factors directly influence the quality of this boundary ellipse: quantisation noise,

(a) 70° (b) 75° (c) 80°

Figure 6: Synthetic images, corresponding to the 35×35 pixel size experiment, magnified 500%. At this size, only (a) and (b) were successfully detected by the system.

Table 1: Maximum standard deviation (in mm), computed per slant angle, over slant angles $< 70^\circ$, derived from degraded synthetic images.

Degradation (σ)		**Fiducial size (pixels)**			
Blur	Noise	120×120	60×60	45×45	35×35
0.5	1%	0.101	0.497	1.487	9.051
0.5	2%	0.203	0.989	2.059	5.991
0.7	1%	0.122	0.623	1.565	4.292
0.7	2%	0.245	1.238	2.378	7.907

Table 2: Maximum z-coordinate error (in mm) over slant angles $< 70^\circ$, computed from degraded synthetic images.

Degradation (σ)		**Fiducial size (pixels)**			
Blur	Noise	120×120	60×60	45×45	35×35
0.5	1%	1.338	3.986	10.11	41.99
0.5	2%	1.534	5.042	11.42	46.08
0.7	1%	0.797	7.587	16.46	47.02
0.7	2%	1.037	8.902	18.42	52.55

sensor noise, and defocus blur.

Quantisation noise is effectively reduced by increasing the size of the ellipse, since more pixels now participate in its definition. Additive sensor noise is also effectively reduced by increasing the size of the ellipse, since the expected mean value of additive noise tends to zero as the number of pixels along the boundary of the ellipse increases. Defocus blur tends to spread the boundary over a larger area, ultimately leading to degradation owing to quantisation errors introduced by the limited bit depth of each pixel.

A monocular 3D pose approach is particularly sensitive to defocus blur, because this (together with quantisation) affects the apparent size of the ellipse, which in turn affects its estimated distance from the camera centre. The effect of the *slant angle*, that is, the angle between the surface normal of the fiducial and the viewing direction, should also be considered. Intuitively, as a circle turns away from the viewing direction, the eccentricity of its projection as an ellipse also increases, which effectively reduces the length of the boundary used to estimate the ellipse parameters, leading to larger errors in position estimates. In order to track displacements in the sub-millimeter range, the calculated position estimates must be repeatable, *i.e.*, their standard deviation over repeated measurements must be less than one millimeter.

4.1. Experiments using synthetic images

To evaluate the effect of these degradations on the proposed system, a number of experiments involving synthetic images were performed. For each viewing distance, a total of 90 base images are created using the POVRay ray tracer[1]. These base images correspond to fiducial patterns with slant angles from 0 to 90 degrees, in 1-degree increments. Each image was degraded first by blurring with a Gaussian kernel to simulate defocus, followed by the addition of zero-mean Gaussian noise to simulate sensor noise. For each viewing angle, distance and blur combination, a total of 30 additive noise images were instantiated. This process was repeated for several fiducial sizes, representing images captured at various distances from the target. Figure 6 illustrates some fiducial patterns viewed at large slant angles.

In a monocular 3D tracking system, it is expected that the extraction of the z-coordinate will be less reliable than the x- and y-coordinates. It is therefore important to measure the robustness of z-coordinate estimates on degraded images. Figure 4 illustrates the effects of slant angle and fiducial size on such degraded synthetic images. Observe how the z-coordinate standard deviation of the largest fiducial remains very small for slant angles less than 80°, whereas the smallest fiducial, at 35×35 pixels, produces significantly larger standard deviations, and degrades rapidly at slant angles greater than 55°. Table 1 lists the maximum standard deviation for a given target size at slant angles below 70° for various noise and blur combinations.

Similarly, Figure 5 and Table 2 illustrate the effective error in the z-coordinate under different slant angle and degradation combinations. The TRIP system [8] was reported to produce a z-coordinate error of 60mm at a slant angle of 60° at a distance of 1900mm, resulting in an error of 3.15% at the equivalent of a 35×35 pixel fiducial size. On the same size fiducial, our system achieves a maximum error of 14.7mm on slant angles below 60° at a distance of 1320mm, or 1.11% using degraded synthetic images[2] — see Figure 5. This indicates that the positional accuracy of the proposed system is comparable to that of the TRIP system.

The physical dolos models shown in Figure 2 measure around 38mm in length. From Table 2 one can see that the 60×60-pixel fiducial produces z-coordinate errors on the order of 10% of the size of the model at large slant angles. Fortunately, the x- and y-coordinate estimates are much more robust than the z-coordinate estimates. For comparison, the maximum Euclidean error (after discarding the z-coordinate) over all slant angles is only 0.0659mm for the values corresponding to row one of Table 2. This would suggest that a weighted Euclidean distance should be used when computing tracking the movement of a fiducial over time.

4.2. Experiments using captured images

In order to relate the synthetic results to real images captured with a digital camera, an experiment was set up to compare relative distances in both the real and synthetic images. Real images were captured using a 6-megapixel Nikon D40 camera at a focal length of 45mm. The images were captured in raw mode, and all the standard processing steps (such as sharpening) were disabled. The images were not corrected for lens distortion, since these effects are negligible in the central area of the lens used in these tests. The fiducials were imaged at distances ranging from 600mm to 900mm in 100mm increments. The diameter of the printed fiducials 7.1mm, to match the scale of the physical dolos models.

Figure 7 shows an image captured under the conditions used to evaluate the accuracy of distance measurements between fiducials. The combination of sensor noise, paper grain, and toner unevenness results in an estimated additive noise component of between 0.5% and 1% of the dynamic range. The same configuration was also modeled and rendered using POVRay. Table 3 lists mean distances measured between the fiducials, computed from a sample of 10 images at each camera-to-target distance. From the table one can see that the captured

[1] `http://www.povray.org`

[2] Our fiducial central dot is smaller, relative to the outer track, than the one used in TRIP. This accounts for the fact that the same size image, 35×35 pixels, results in different distances from the camera.

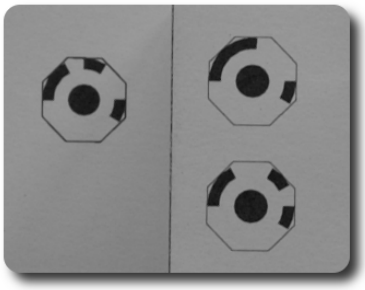


Figure 7: Fiducial test configuration image: the fiducial on the left ($\vec{v}_3$) is 25mm further from the camera than the two coplanar fiducials ($\vec{v}_1$, $\vec{v}_2$) on the right.

Table 3: Mean relative distance measures (and standard deviation) obtained from real and synthetic images. The matrix $\mathbf{P}$ denotes a projection onto the z-axis, and $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$ denote the 3D centre coordinates of three different fiducials. The expected values for the measures are 25mm and 10mm, respectively.

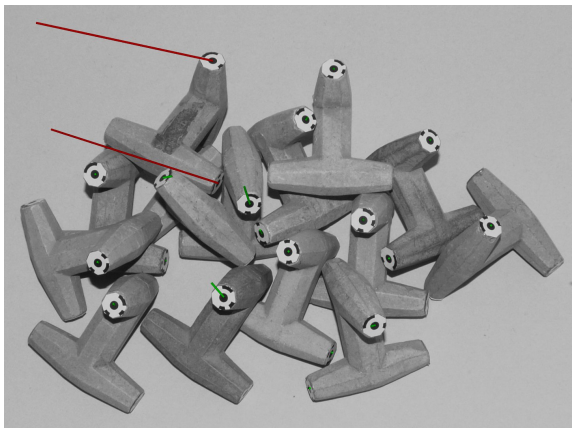
Camera distance (mm)	$\|\mathbf{P}\vec{v}_1 - \mathbf{P}\vec{v}_3\|$ (mm)		$\|\vec{v}_1 - \vec{v}_2\|$ (mm)	
	Real	Synth.	Real	Synth.
600	27.700 (0.219)	24.956 (0.309)	9.964 (0.0187)	10.090 (0.0483)
700	24.272 (0.441)	23.633 (0.328)	10.043 (0.0676)	10.043 (0.0057)
800	22.067 (0.470)	23.241 (0.387)	10.146 (0.0790)	10.200 (0.0427)
900	21.474 (0.965)	23.992 (0.549)	10.464 (0.1632)	10.107 (0.0301)

images ("Real") exhibit a slight trend, so that the z-distance between the fiducials appears to decrease as the camera moves further away from the fiducials. This effect can be partly attributed to the difficulty of obtaining the exact same focus quality at multiple camera-to-target distances — Table 2 clearly shows that increased blur, corresponding to poorer focus, leads to larger z-coordinate errors. The degraded synthetic images (produced with a blur σ of 0.5, and a noise σ of 1%) did not appear to suffer from this effect, as could be expected.

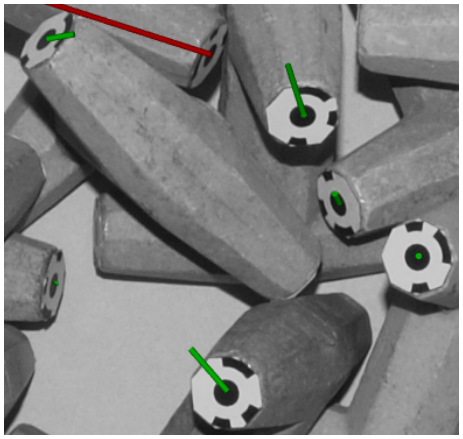
It is encouraging, to see that the standard deviation of z-distances captured at a distance of 800mm is less than 0.5mm. Distances measured between coplanar fiducials at the same distance from the camera appear to be much more robust, yielding an error of less than 0.2mm at a distance of 800mm, with a standard deviation of less than 0.08mm. From Table 3, it appears that a distance of 700mm offers sufficient accuracy to measure displacements on the order of 0.5mm with the camera specified above. Since these measurements were performed at a slant angle of 0°, it will still be necessary to filter fiducials with large slant angles, or to apply a weighted Euclidean metric to compensate for the large z-coordinate measurement errors that occur at large slant angles.

5. Application

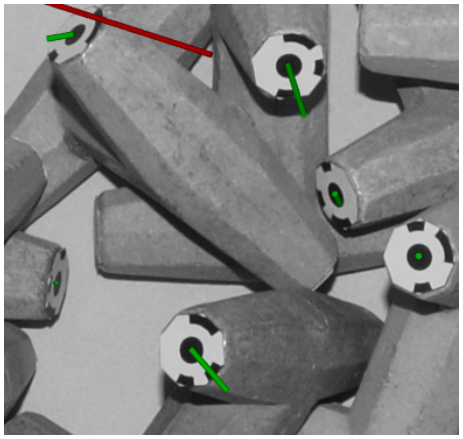
The system described in Section 3 was used to track the movement of fiducials attached to physical breakwater armour unit models. For this experiment, the "dolos" type armour unit was selected, and four different fiducial patterns were attached to the four end-points of the dolos models. Note that the same four



(a) Before movement, 50% cropped image



(b) Before movement, 5% cropped image



(c) After movement, 5% cropped image

Figure 8: Displacement vectors, calculated automatically by computing the movement of the fiducials between the two frames.

fiducial patterns were attached to all the dolosse, hence is it is not possible to uniquely identify a given dolos only by the code associated with its fiducial patterns. This method is at one extreme, where there are many objects with identical fiducials — it is possible to use more fiducial codes, but it may not be possible to assign unique fiducials to all the models in large simulations involving hundreds of dolosse. This experiment thus re-

lies heavily on correspondence matching to correctly track the movement of fiducials, and is therefore considered to be a more stringent test of the system.

An array of dolosse were arranged as shown in Figure 8(a). A number of dolosse were manipulated by hand to approximate the (hypothetical) movement induced by a wave-tank simulation. A second image was captured after the induced movement — a close-up of a region containing significant movement is shown in Figures 8(b) and (c). The green and red cylinders represent 3-dimensional displacement vectors. They were rendered using POVRay, and superimposed on top of the original images. Green cylinders represent a displacement of a matched pair of fiducials, *i.e.*, the exact same fiducial pattern occurred in the *before* and *after* images. Red cylinders indicate that the fiducial pattern types did not match, but that these are still likely candidates for a match, based on their physical proximity. For example, the two red cylinders visible in the upper left corner of Figure 8(a) are the result of the upper left-most dolos being displaced and overturned. This implied that the fiducials visible in the *before* image were facing away from the camera in the *after* image, but the correspondence algorithm still matched them with the fiducials on the reverse side of the model since they were still considered to be the most likely candidates.

Allowing matches between fiducials with different patterns can help to identify large displacements, but these matches are inherently less reliable than matches with identical patterns, and are only allowed here to illustrate the advantage of using a global correspondence matching algorithm.

If a large number of fiducial codes is used, then each individual dolos may receive its own code, unique within a certain radius in the original packing. This will reduce the possibility of incorrect matches to zero for most simulations.

6. Conclusions

This paper demonstrated that fiducials can used to track the movement of physical breakwater armour unit models to a submillimetre scale. The sensitivity and robustness of the system was investigated using both synthetic and captured images. Estimating the z-coordinate of a circular target using a monocular 3D system is feasible, but the accuracy and robustness of this estimate is heavily influenced by the size of the target, and the slant angle. On captured images, the absolute error in extracted x- and y-coordinates can be kept below 0.2mm; the absolute z-coordinate errors are on the order of 2–3mm, but with a standard deviation of less than 0.5mm.

The fiducial pattern used in our experiments depends on the central dot for the position calculations. In retrospect, this seems to have been a poor choice, since a different design, like that of Naimark and Foxlin [4], allows one to use the outer perimeter of the fiducial as circular reference. This would imply that the effective diameter of the circle would increase by a factor three, without increasing the physical size of the pattern. Even a more modest increase by a factor of two could reduce the position errors by a factor of three, as shown in Section 4. Future work will focus on repeating the experiments with an alternative fiducial design that maximises the size of the circle used to perform pose estimation.

7. Acknowledgements

The authors would like to thank the Strategic Research Panel (SRP) for providing support for this research through a CSIR project entitled "Advanced Digital Image Technology for Port Engineering".

8. References

[1] P. Bakker, A. van den Berge, R. Hakenberg, M. Klabbers, M. Muttray, B. Reedijk, and I. Rovers, "Development of Concrete Breakwater Armour Units," in *1st Coastal, Estuary and Offshore Engineering Specialty Conference of the Canadian Society for Civil Engineering*, New Brunswick, Canada, 2003.

[2] A. K. Cooper, J. M. Greben, F. van den Bergh, I. M. A. Gledhill, B. R. Cannoo, W. J. V. D. M. Steyn, and R. de Villiers, "A preliminary physics-engine model of dolosse interacting with one another," in *Proceedings of the Sixth South African Conference on Computational and Applied Mechanics (SACAM08)*, Cape Town, South Africa, March 2008.

[3] J. W. Berger, T. R. Patel, D. S. Shin, J. R. Piltz, and R. A. Stone, "Computerized stereochronoscopy and alternation flicker to detect optic nerve head contour change," *Ophthalmology*, vol. 107, no. 7, pp. 1316–1320, 2000.

[4] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, 2002, pp. 27–36.

[5] C. B. Bose and J. Amir, "Design of fiducials for accurate registration using machine vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 12, pp. 1196–1200, Dec 1990.

[6] C. B. Owen, F. Xiao, and P. Middlin, "What is the best fiducial," in *The First IEEE International Augmented Reality Toolkit Workshop*, Sept. 2002, pp. 98–105.

[7] J. Rekimoto and Y. Ayatsuka, "CyberCode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*, Elsinore, Denmark, Apr. 2000, pp. 1–10.

[8] D. López de Ipiña, P. R. S. Mendonça, and A. Hopper, "TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing," *Personal and Ubiquitous Computing*, vol. 6, no. 3, pp. 206–219, 2002.

[9] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[10] J. Maciel and J. P. Costeira, "A Global Solution to Sparse Correspondence Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 187–199, 2003.

[11] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13–21, 2007.

[12] F. Chang, C. J. Chen, and C. J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, vol. 93, no. 2, pp. 206–220, 2004.

[13] J. N. Ouellet and P. Hebert, "A Simple Operator for Very Precise Estimation of Ellipses," in *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, 2007, pp. 21–28.

PORTING A SPOKEN LANGUAGE IDENTIFICATION SYSTEM TO A NEW ENVIRONMENT

Marius Peché[1], Marelie Davel[2] and Etienne Barnard[2,]
mpeche@csir.co.za, mdavel@csir.co.za, ebarnard@csir.co.za

[1]Department of Electrical, Electronic and Computer Engineering, University of Pretoria.
[2]HLT Research Group, Meraka Institute, CSIR.

ABSTRACT

A speech processing system is often required to perform in a different environment than the one for which it was initially developed. In such a case, data from the new environment may be more limited in quantity and of poorer quality than the carefully selected training data used to construct the system initially. We investigate the process of porting a Spoken Language Identification (S-LID) system to a new environment and describe methods to prepare it for more effective use. Specifically we demonstrate that retraining only the classifier component of the system provides a significant improvement over an initial system developed using acoustic models channel-normalized to the new environment. We also find that the most accurate system requires retraining of both the acoustic models and the final classifier.

Index Terms — Spoken Language Identification, S-LID.

1. INTRODUCTION

Spoken Language Identification (S-LID) is the process whereby a sample of audio speech from an unknown source is classified as one of several possible languages [1]. This can be done in a number of ways, including sampling the prosodic information or processing information extracted from specified tokens, where such tokens may be phonological or syntax related [2]. In the latter case, spoken LID differs significantly from textual LID because text already consists of properly defined and accurate tokens (such as alphabetical letters) while these tokens (such as phonemes) must first be extracted from audio speech, and may not be accurate.

In addition, more accurate S-LID systems usually are more complex and require a larger amount of data to create systems with sufficient performance [3]. The popular Parallel Phone Recognition and Language Model (PPR-LM) approach [1] provides reasonably high system accuracy with acceptable data requirements, and is the approach experimented with in this paper.

In a PPR-LM system, separate phone recognizers are used to tokenize an incoming audio signal individually, and a classifier trained to identify the language spoken based on the token strings received in parallel from the various phone recognizers. Initially based on language modeling scores, various classifiers have since been used in literature, with Support Vector Machines (SVMs) achieving high accuracy [4].

Once developed for a specific environment, it is often required that a S-LID system be ported to a new environment. Data from such a new environment may be more limited in quantity and of poorer quality than the carefully selected training data used to construct the system initially.

We investigate the process of porting a PPR-LM based S-LID system to a new environment and describe methods to prepare it for more effective use. Specifically we compare the effect of re-training the classifier component with that of re-training both the acoustic modeling and classifier component and report on results.

The paper is structured as follows: In section 2 we describe the design of our baseline system. In section 3 we describe the porting process step by step, specifically focusing on data preparation, initial system adaptation, classifier adaptation, acoustic model adaptation and final analysis. Section 4 contains some concluding remarks.

2. BASELINE SYSTEM DESIGN

We develop an initial S-LID system able to identify three languages: English, French and Portuguese. English data is obtained from the Wall Street Journal corpus, and French and Portuguese data from the GlobalPhone corpus [6]. (These two corpora have similar acoustic characteristics.)

Using a PPR-LM approach, we develop three Automatic Speech Recognition (ASR) systems capable of performing phone recognition, each in one of the languages English, French or Portugues. These ASR systems utilize Hidden Markov Models (HMM) which have been trained to recognize bi-phones from Mel Frequencies Cepstral Coefficients (MFCC). The training of the HMMs as well as the extraction of the MFCCs from the audio signal are performed using the HMM Tool Kit (HTK) [7]. These phone recognition systems run in parallel with one other, each yielding a phoneme string for a given speech sample.

We use the 'Bag-of-Sounds' principle to model the frequencies of phonemes as a vector, with the frequency of each phoneme within the sample of speech representing an element of this vector. These vectors are then used to train a Support Vector Machine (SVM) using the LIBSVM [5] toolkit. In all experiments, a radial basis function kernel is used and the kernel width and misclassification cost are optimized using a grid search. Multiple classes are handled using a 1 against n-1 scheme.

Using a flat phone grammar with approximately 40 phones per language, we achieve phone recognition accuracies of 48% to 66% for the three ASR systems on an independent test set. While these accuracies seem fairly low, they are sufficient to obtain highly accurate S-LID results, as displayed in Table 1. S-LID results are obtained using the same test set as used to report on ASR accuracies, and durations of the speech samples range from 10 to 60 seconds each.

Language	Word recognition accuracy	S-LID accuracy
English	52.8%	98.9%
French	66.2%	94.9%
Portuguese	48.1%	97.7%

Table 1: Accuracies achieved by baseline system

3. PORTING THE S-LID SYSTEM

In this section we first discuss the new environment investigated and the data available from this environment, before providing detail with regard to the different aspects of our approach to porting the S-LID system, specifically consisting of (1) data preparation, (2) initial system adaptation, (3) classifier adaptation, (4) acoustic model adaptation and (5) final analysis.

3.1 Data description

In order to investigate porting of the S-LID system to a new environment, we utilize a telephone corpus of African variants of the three languages of interest (referred to from here onwards as the African corpus). The African corpus consists of approximately 45 hours of speech separated into English, French and Portuguese variants spoken on the African continent. The speech is untranscribed and no additional speaker information is available. Single calls are assumed to be from a single speaker and most calls are assumed to be from different speakers. The amount of data, in hours, is displayed in Table 2.

The initial S-LID system is therefore required to perform in a new environment with significantly different channel conditions and speech dialects. In addition, the new data contain non-speech signals as well as competing background noises.

Data from the three different languages in the new corpus are identified according to language. The new corpus is separated into a training and test corpus as indicated below, with the same test set used to report on results. Care is taken to ensure that the same speaker is not included in both the training and test set.

Language		**GlobalPhone**		**African Corpus**	
		train	test	train	test
English	Hours	20.2	4.85	16.77	4.3
	Speakers	83	19	250	25
French	Hours	21.6	5.3	8.25	2.07
	Speakers	80	21	109	26
Portuguese	Hours	14.4	3.6	11.18	2.84
	Speakers	77	25	108	28

Table 2: Training and testing data statistics.

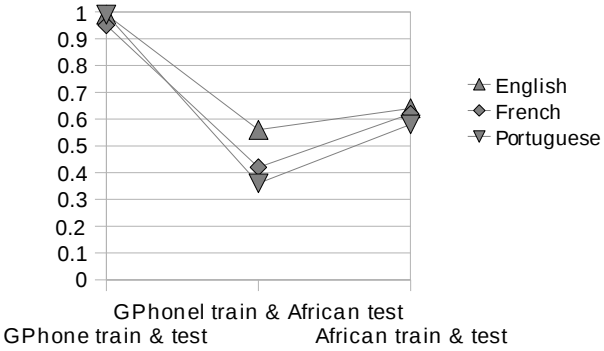


Figure 1: S-LID Accuracy when the GlobalPhone ASR system is used, but the SVM is trained on different corpora.

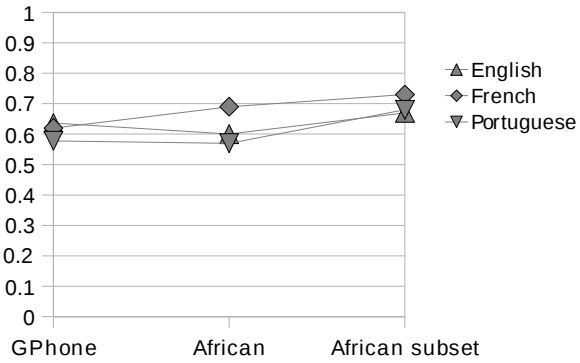


Figure 2: S-LID Accuracy with the African corpus when different ASR systems are used

3.2 Data Preparation

Our first task is to pre-process the new data. We use diarization techniques to separate the different speakers and to remove any non-speech signals. We also remove long sections of silence from the audio, perform amplitude normalization and segment the new audio files into sections of no larger than one minute each.

3.3 Initial System Adaptation

Once the new data has been preprocessed (as above) it can be used to estimate the channel conditions of the new environment. The GlobalPhone corpus can now be downsampled (to 8KHz, the sampling frequency of the African corpus), amplitude normalized and channel normalized in order to better match the new environment [8].

In order to verify the new system, ASR and S-LID accuracies are calculated using the same test corpus as before. While ASR accuracies decrease with between 7% and 14% absolute, overall S-LID accuracy increases from 97.18% to 98.26%.

It should be noted that these results still refer to data from the previous (GlobalPhone) environment. Once this system is tested using the new data, an S-LID accuracy of only 47.02% is obtained.

3.4 Classifier Adaptation

We now adapt the classifier to the new environment: we tokenize the new training data using the normalised GlobalPhone recognizers and use these phone stings to re-train the SVM. S-LID accuracy improves dramatically, from the previous 47.02% to 62.57%. The differences in performance between an optimal system (GPhone train&test), an unported system (GPhone train & African test) and the ported system with only the SVM adapted (African train & test) are depicted in Figure 1.

3.5 Acoustic Model Adaptation

In order to further improve the performance of the system with the African corpus, we train new acoustic models for the tokenizers. We use the normalised GlobalPhone recognisers to bootstrap transcriptions for the new data (since the African corpus is not transcribed), and use these transcriptions to train new acoustic models. [9] Once new acoustic models are trained, these are used to re-tokenize the new audio data and re-train the classifier.

Initially results are disappointing as S-LID accuracy falls to 60.58%. However, when transcriptions are filtered to exclude the training and test utterances that were clearly hard to recognize (transcriptions that contain fewer than one and a half phones per second) S-LID accuracy increases to 68.88%. This is the highest accuracy obtained using the full test set. The effect of using different ASR systems on S-LID accuracy is depicted in Figure 2.

3.6 Final Analysis

While an improvement from 47.02% to 68.88% is significant, these results are still lower than anticipated and further analysis of the data set is required. When subsets of the new data set are systematically listened to by human verifiers it is noted that the new corpus contains data of highly variable quality.

While initial random testing of the corpus provided some indication of the quality of the data, a systematic analysis by human verifiers indicates that significant portions of the corpus contain the following problematic subsets:

- Data incorrectly labeled or unusable, meaning that the language spoken is neither English, French or Portuguese ('Unusable').
- Data correctly labeled, but spoken with a strong accent ('Accented').
- Data correctly labeled but consisting mostly of noise with similar spectral characteristics as speech, which the diarization system did remove (also included as 'Unusable').

Data correctly labeled and identifiable as either English, French or Portuguese are indicated as 'Correct' by the human verifiers. The number of samples evaluated that falls within each category for each of the languages is listed in Table 3. (Note that only a subset of the full corpus was evaluated.)

Language	Unusable	Accented	Correct
English	109	72	118
French	110	4	159
Portuguese	35	1	109

Table 3: Number of samples per category as verified by human verifiers

This table provides a new perspective on the results obtained (in Section 3.5). As a large percentage of the samples are in fact unusable, an S-LID accuracy of 68.88% is indeed highly encouraging. Further analysis and optimisation can now be done using the smaller "correct" subsets in order to obtain a better indication of system performance.

4. CONCLUSION

In this paper we describe the process of adapting an existing S-LID system to a new environment. We describe the different stages in such a process and provided results for each stage. We highlight the importance of verifying the quality of the data from the new environment systematically as an important step during system porting. We show that bootstrapping transcriptions from existing ASR systems, and re-training the classifier using the bootstrapped transcriptions provide a significant improvement in performance and that the most accurate system requires retraining of both the acoustic models and the final classifier, this is with a small margin only

In further work we are currently repeating some of the above experiments using only the small portions of data identified as "Correct" during human verification. We are also investigating automated mechanisms to identify problematic audio samples during system development.

5. REFERENCES

[1] Y.K. Muthusamy, E. Barnard and R.A. Cole, "Reviewing Automatic Language Recognition", IEEE Signal Processing Magazine, Oct 1994.

[2] Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li and Eng Siong Chng, "Integrating Acoustic, Prosodic and Phonotactic Features for Spoken Language Identification", In ICASSP-2006, Toulouse, France. pp 205-208.

[3] Marc A. Zissman, Kay M. Berkling, "Automatic language identification", Speech Communication, 35 (2001) pp 115-124.

[4] Haizhou Li, Bin Ma, "A phonotactic language model for spoken language identification," In Proc. ACL-2005, pp. 515–522.

[5] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[6] Tanja Schultz, "GlobalPhone: a multilingual speech and text database developed at Karlsruhe University", In ICSLP-2002, Denver, Colorado, USA. pp 345-348.

[7] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland, “The HTK book. revised for HTK version 3.3,” September 2005. Software available at http://htk.eng.cam.ac.uk/

[8] Neil Kleinhans, “Channel normalization for speech recognition in mismatched conditions”, *accepted for publication*, In PRASA-2008, Cape Town, South Africa.

[9] Marius Peché, Marelie Davel, Etienne Barnard, “Phonotactic spoken language identification with limited training data” In Interspeech-2007, Antwerp, Belguim. pp 1537-1540.

Relationship between Structural Diversity and Performance of Multiple Classifiers for Decision Support

R. Musehane[1], F. A Netshiongolwe[1], L. Masisi[1], F. V. Nelwamondo[2], T. Marwala[1]

[1]School of Electrical & Information Engineering,
University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa
[2] Modelling and Digital Science, CSIR, South Africa

rofhiwa.musehane@students.wits.ac.za

Abstract

The paper presents the investigation and implementation of the relationship between diversity and the performance of multiple classifiers on classification accuracy. The study is critical as to build classifiers that are strong and can generalize better. The parameters of the neural network within the committee were varied to induce diversity; hence structural diversity is the focus for this study. The number of hidden nodes and the output activation function are the parameters that were varied. The diversity measures that were adopted from ecology such as Shannon and Simpson were used to quantify diversity. Genetic algorithm is used to find the optimal ensemble by using the accuracy as the cost function. The results observed shows that there is a relationship between structural diversity and accuracy. It is observed that the classification accuracy of an ensemble increases as the diversity increases. However, there is a point where as diversity increases, the accuracy does not increase. Furthermore, the paper also presents the effect of ensemble size on the prediction accuracy. This investigation is necessary in order to know and ensure the optimal size of classifiers that can be used in an ensemble. It has been observed that as the size of the ensemble increases, the accuracy increases.

Key words: Classification, Diversity Measures, Ensemble Size, Genetic Algorithm, Structural Diversity

1. Introduction

Computational intelligence techniques have been used in many classification problems. The literature emphasises that a group of classifiers is better than one classifier [1-5]. This is because the decision that is made by a committee of classifiers is better than the decision made by one classifier. In this paper the committee of classifiers will be referred as an ensemble. The most popular way to gain confidence on the generalisation ability of an ensemble is by introducing diversity within the ensemble [1, 2, 5]. This has led to the development of measures of diversity and various aggregation schemes for combining classifiers. However, diversity is still not clearly defined [6, 7]. Thus, a proper measure of diversity that will relate diversity to accuracy is to be adopted. Current methods commonly use the outcome or generalization performance of the individual classifiers of an ensemble to measure diversity. Hence an ensemble is considered diverse if classifiers within the ensemble produce different outcomes as opposed to having the same outcomes [1, 6, 7].

In this paper, as opposed to looking at the outcomes of the individual classifiers, ensemble diversity is viewed as the structural variation within classifiers that form an ensemble [1, 5]. Thus, diversity will be induced by changing structural parameters of a neural network [5]. The paper investigates the relationship between structural diversity within an ensemble and the prediction accuracy of the ensemble. It has been intuitively accepted that the classifiers to be combined should be diverse [8]. This is because it has been found meaningless to combine identical classifiers because no improvement can be achieved when combining them [8, 9]. Hence, measuring structural diversity and relating it to accuracy is crucial in order to build better learning machines. However, it is necessary to find the optimal size of an ensemble that gives better generalization. Therefore, a study on the size of the ensemble was done as to find the optimal size that can be used for the investigation. The methods for measuring structural diversity are to be devised and implemented. Moreover, the outcome diversity of structurally different classifiers is critical to be measured. This is because it is essential to show how correlated the outcomes of the structurally different classifiers is.

Different methods for creating diversity such as bagging and boosting have been explored [1, 3]. However, the aggregation methods are to be used to combine the ensemble predictions. Methods of voting and averaging have been found to be popular [9, 10] and hence are used in this study. The paper first discusses the background in section 2. Analysis of the data used for this study is presented in section 3. The accuracy measure and structural measures of diversity used are discussed in section 4 and section 5. The methodologies used in investigating the effect of diversity on generalization are presented in section 6. The results and future work are then discussed in section 7.

2. Background

2.1. Neural Networks

Neural Networks (NN) are computational models that have the ability to learn and model linear and non-linear systems [11]. There are many types of neural networks but the most common neural network architecture is the multilayer perceptron (MLP) [11]. The neural network architecture that is used in this paper is a MLP network as

shown in Figure 1. The MLP network has the input layer, the hidden layer and the output layer. An MLP network has parameters such as learning rate, number of hidden nodes and the activation function. These parameters can be varied to induce structural diversity [5]. The general equation of the output function of MLP neural network is shown below (1).

$$y_k = f_{outer}(\sum_{j=1}^{M} w_{kj}^{(2)} f_{inner}(\sum_{i=1}^{N} w_{ij}^{(1)} x_j + w_{j0}^{(1)}) + w_{k0}^{(2)}) \quad (1)$$

where: y_k is the output from the neural network, f_{outer} is the output activation function that can be linear, softmax or logistics, f_{inner} is the hidden layer tangential activation function. M is the number of the hidden units, N is the number of input units, $w_{kj}^{(2)}$ and $w_{ij}^{(1)}$ are the weights in the first and second layer moving from input i to hidden unit j, $w_{0j}^{(1)}$ and $w_{0k}^{(2)}$ is the biases for the unit j.

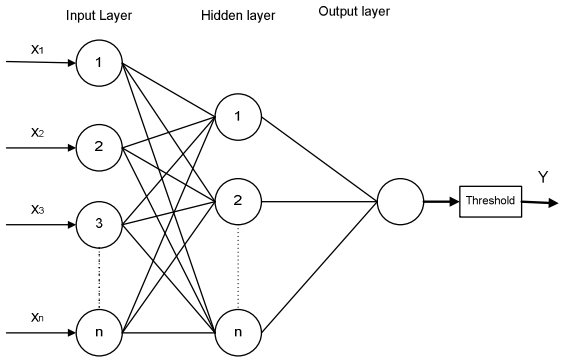


Figure 1: *The MLP neural network architecture.*

The input into the neural network is the demographic data from the antenatal survey and the output is the HIV status which is zero to indicate negative and one for positive. The weights of the NN are updated using a back propagation algorithm during the training stage and the focus is on minimizing the error in predicting the output [11]. The structural variation that was used to achieve structural diversity in this study is the number of hidden nodes and the output activation function. That means that the number of hidden nodes and the output activation were varied to achieve structural diversity. This implies that every neural network that is different from the other in terms of the structure is considered diverse.

2.2. Genetic Algorithm

The genetic algorithms (GA) are computational models that are based on the evolution of biological population [2]. Potential solutions are encoded as the chromosomes of some individual. These individuals are initially generated randomly. The individuals are evaluated through the defined fitness function. Each preceding generation is populated by the fitness solution (members) of the previous generation and their offspring. The offsprings are created through crossover and mutation. The crossover process combines genetic information of two previous fittest solutions to create new offsprings. Mutation alters the genes of the individual to introduce more diversity into the population. In this way, the initial generated solution can be improved over time [2, 12].

In applying the GA to the study of structural diversity, the evaluation function can be the structural diversity or the generalization performance. The main idea is to relate the structural diversity to the generalisation performance. If the evaluation function is the generalization performance, the GA will then look for a defined number of classifiers that give that performance and then the diversity of those classifiers can be measured. This method then helps to relate generalization performance and the structural diversity.

3. Data Analysis

3.1. Data Collection

The dataset used for the study is from the antenatal clinics in South Africa. The dataset was that collected by the department of health in 2001 [13]. The features in the data include the age, gravidity, parity, education, etc. The data was collected from the pregnant woman only. The demographic data used in the studies is shown in table 1 below. The province was provided as a string so it was converted to integer form 1 to 9.

Table 1: *The features from the survey*

	Variable	Type	Range
1	Age	integer	13-50
2	Education	integer	0-13
3	Parity	integer	0-9
4	Gravidity	integer	1-12
5	Province	integer	1-9
6	Age of father	integer	14-90
7	HIV status	binary	0-1

The age is that of the pregnant mother visiting the clinic. Education represents the level of education the mother has and ranges from 1-13, where 1-12 corresponds to grade 1 to 12 and 13 represents tertiary education. Parity is the number of times the mother has given birth whilst gravidity is the number of times the mother has been pregnant. Both these quantities are important, as they show the reproductive activity as well as the reproductive health state of the women. The age of the father responsible for the current pregnancy is also given and the province entry corresponds to the geographic area where the mother comes from. The last feature is the HIV status of the mother where 0 represents a negative status whilst 1 represents a positive status.

3.2. Data Pre-Processing

The data preprocessing includes elimination of the impossible situations like when parity is greater than gravidity. It is not possible for the mother to give birth without falling pregnant; therefore it is not possible to find a case where parity is greater than gravidity. The pre-processing of the data resulted in a reduction of the dataset. To use the dataset for training, it needs to be normalized. This ensures that all variables can contribute to the final network weights of prediction model [14]. If the data are not normalized, some of the data variables with larger variances will influence the result more than others. Therefore, all the data is to be normalized between 0 and 1 using (2).

$$x_{norm} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \qquad (2)$$

where: $x_{\min}$ and $x_{\max}$ is the minimum and maximum value of the features of the data samples respectively.

The data were divided into three sets, the training, validation and testing data. This was done as to avoid over-fitting of the network. The neural networks are trained by 60% of the data, validated with 20% and tested with 20%.

4. Accuracy Measure

Regression problems mostly focus on using the mean square error between the actual outcome and the predicted outcome as a measure of how well neural networks are performing. In classification problems, the accuracy can be measured using the confusion matrix [15]. Analysis of the dataset that is being used showed that the data is biased towards the negative HIV status outcomes. Hence, the data was divided such that there is equal number of HIV positive and negative cases. This was an advantage in order to allow the use of confusion matrix to measure accuracy. The accuracy measure that is used in this study is given by (3).

$$Accuracy\,\% = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \qquad (3)$$

where:
TP is the true positive -1 classified as a 1,
TN is the true negative - 0 classified as a 0,
FN is the false negative -1 classified as a 0,
FP is the false positive - 0 classified as a 1.

5. Measurements of Structural Diversity

5.1. Shannon-Wiener Diversity Measure

Shannon entropy is a diversity measure that was adopted from ecology and information theory to understand ensemble diversity [16]. This measure is implemented to measure structural diversity. The Shannon-Wiener index is commonly used in information theory to quantify the uncertainty of the state [16, 17]. If the states are diverse one becomes uncertain of the outcome. It is also used in ecology to measure diversity of biological species. Instead of biological species, the species are considered as the individual base classifiers. The Shannon diversity measure is given by (4).

$$D = -\frac{\sum_{i=1}^{M} \left(\frac{n_i}{N}\right) \ln\left(\frac{n_i}{N}\right)}{\log(N)} \qquad (4)$$

Where:

n_i = number of neural networks that have the same structure

N = total number of neural networks in an ensemble

M = total number of different neural networks/species

D = the diversity index

The diversity ranges from 0 to 1, where 0 indicates low diversity and 1 indicates highest diversity.

5.2. Simpson Diversity Measure

The other measure that was implemented is the Simpson diversity measure. This measure is also adopted from ecology to quantify diversity. It is quantified by (5).

$$D = \sum_{i=1}^{M} \frac{n_i(n_i - 1)}{N(N-1)} \qquad (5)$$

n_i = number of neural networks that have the same structure

N = total number of neural networks in an ensemble

M = total number of different neural networks/species

The diversity index is given by $1 - D$. The diversity increases as the index increases. It ranges from 0 to 1 where 0 means there is no diversity and 1 indicate the highest diversity.

6. Implementation

6.1. Creation of Diverse Classifiers

Since the focus of the study is the structural diversity, the output activation function, learning rate and the number

of hidden nodes were varied as to induce diversity. However, varying all the parameters was found to be ineffective because the classifiers tend to generalize the same way. Therefore, only number of hidden nodes and activation function were varied for this investigation.

The classifiers are trained individually using the back propagation method; where the error is propagated back so as to adjust the weights accordingly. The data used for training, validation and testing are the HIV data. All the features of the input are fed to all the networks. The classifiers which have the training accuracy of 60% were accepted. The training accuracy between 60% and 63% was achieved. The classifiers were trained using quasi-Newton algorithm for 100 cycles at the same learning rate of 0.01. However, the number of hidden nodes is increased from 1 to 55 and the activation function is randomly changed from linear and logistics to induce structural diversity.

6.2. Committee of Classifiers

The committee of classifiers improves efficiency and classification accuracy [18]. This ensures that the results are based on the consensus decision of the base classifiers. The base classifiers operate concurrently during the classification and their outputs are integrated to obtain the final output [18]. The model for the committee of classifiers is shown in figure 2.

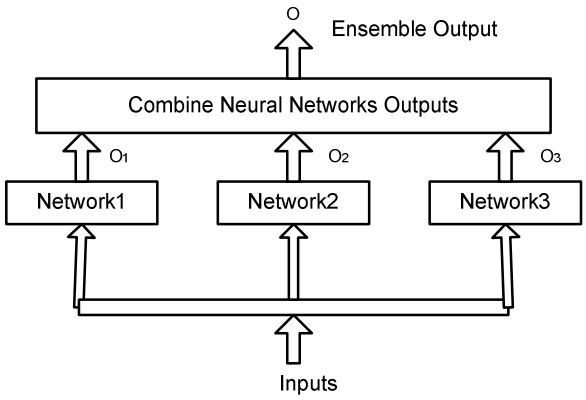


Figure 2: *The classifier ensemble of neural networks*

There are many aggregation methods that can be used to combine the outcomes of classifiers. The ensemble outcomes were all aggregated using simple majority voting. This was chosen because it is popular and easy to implement [9]. The outcomes of each individual from an ensemble are first converted to 0 or 1 using 0.5 as a threshold. The majority voting method chooses the prediction that is mostly predicted by different classifiers [19]. The other method that was implemented was averaging. All the outcomes from all the classifiers are taken and averaged.

6.3. Evaluation of Optimal Ensemble Size

It is important to use the optimal size of an ensemble that results in better generalisation of the data [20]. The ensemble size is determined by the number of classifiers that belong to an ensemble. The created classifiers were used to carry out this experiment. The ensemble size was incremented by one from 1 to 50. However, the structure of the networks was made to be different by varying the number of hidden nodes as the ensemble size increases. Hence, the size of the network itself is increased as the number of classifiers in the ensemble increases [4]. Figure 3 below shows the results obtained.

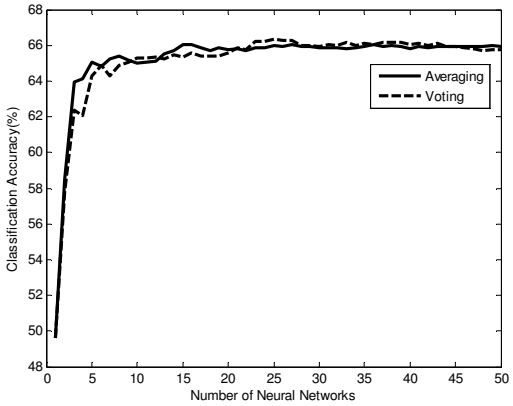


Figure 3: *The ensemble size and classification accuracy*

It was however observed that the relationship between the size and accuracy of the ensemble depends on the accuracy of the individual classifiers that belong to the ensemble. Increasing the size of the neural network by increasing the hidden nodes tends to improve the classification accuracy as the number of the classifiers in an ensemble increases. However, an increase in size results in an increase in the prediction accuracy. Consequently, after the optimal size of 19 classifiers is reached, the accuracy tends to remain constant. Nevertheless, the size of 19 was found to be optimal since it produced the best accuracy. The results obtained are found to be concurrent with literature. Currently the optimal size of an ensemble is 25 [19, 20]. Therefore, an ensemble size of 19 is used for evaluating the relationship between diversity and performance of classifiers on HIV classification.

6.4. Evaluation of Diversity and Accuracy

The created classifiers were used to investigate the relationship between the diversity and accuracy. There were ten base classifiers or species that were selected from the created classifiers which are all structurally different based only on the number of hidden nodes and the output activation function. The networks with the number of hidden nodes from 10 to 55 in steps 5 were chosen from the created classifiers. The GA has the capabilities to search large spaces for a global optimal

solution [5]. GA was therefore used to search for 19 classifiers from the 10 base classifiers using the generalization performance as the fitness function. The fittest function is given by:

$$Fittest\ Function = -(T_{Acc} - Acc)^2 \quad (6)$$

Where: T_{Acc} is the targeted accuracy and Acc is the obtained accuracy.

The GA continues to search until the error between the targeted accuracy and the obtained accuracy is minimal. Firstly, it was necessary to optimize the accuracies that could be attained in order to minimize the computational cost. Thereafter, the attained accuracies were used in the second run as the target accuracy. The size of the neural network committee used is 19 classifiers which are formed from a combination of 10 unique base classifiers. Hence, each ensemble will have a repetition of certain classifiers. Once the ensemble of 19 classifiers produces the targeted diversity, the corresponding structural diversity is obtained using both Simpson and Shannon diversity measures given in (4) and (5). The algorithm implemented is shown in figure 4.

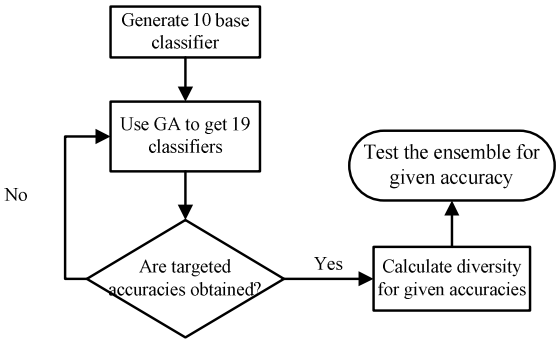


Figure 4: *Algorithm for implementation of the relationship between diversity and accuracy*

7. Results Analysis

7.1. Impact of Diversity on Prediction Accuracy

In this study, diversity was induced by varying the parameters of the classifiers that form an ensemble [5, 17]. The investigation was done on an ensemble of 19 classifiers. Figure 5 shows the obtained results using the Shannon diversity measure. Figure 6 shows the results obtained using the Simpson diversity measure. The figures indicate that an increase in structural diversity results in an increase in accuracy which is in agreement with [17]. The experiment was done several times observing the relationship between diversity and accuracy using both Simpson and Shannon diversity measure. Therefore the results shown above are the average of ten different experiments that were performed. The results show that the two measures are concurrent. In the Shannon diversity measure, the GA was able to attain wide range of diversity whereas in the Simpson measure, the range is limited from 0.8 to 0.9. This was because the Shannon diversity index depends on the number of base classifiers whereas the Simpson's index depends on how evenly distributed the base classifiers are [16]. Shannon has shown that the more uncertain one is of the outcome, the more diverse an ensemble is.The results clearly show that structural variation of the parameters of the neural network (classifier) does have a relationship with prediction accuracy. As the structural diversity increased so did the accuracy.

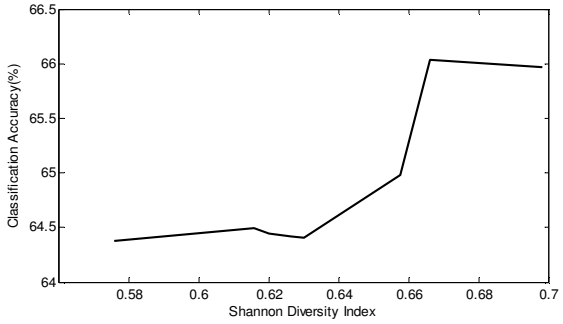


Figure 5: *The evaluation of Shannon index with accuracy*

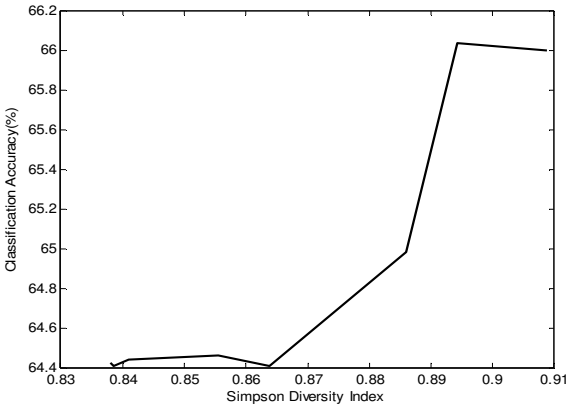


Figure 6: *Evaluation of Simpson index with accuracy*

7.2. Discussion and Recommendations

It was however observed that the individual classifiers within the ensemble were highly correlated in the outcomes. This had affected the results because very low and high accuracies could not be attained. It is however recommended that a strategy of adding classifiers in an ensemble such that only classifiers that are uncorrelated are accepted in an ensemble can be adopted. The experiment focuses on training the classifiers using all the features of the data. It is however recommended that different networks can be fed different features of the

data. This might ensure that the outcomes of classifiers are not highly correlated. Hence, a higher range of accuracy and diversity index can be attained.

During the training stage of the machine, the weights are normally randomly initialised. However, it has been found that different initial weights induce diversity within the ensemble [1]. The Shannon and Simpson diversity measures focuses on how structurally different the classifiers in an ensemble are. These measures do not consider diversity induced during initialisation of weights. Therefore, it is recommended that for future work, a better measure of structural diversity that incorporates the effect of weight initialisation should be developed.

8. Conclusion

The paper presented the relationship between structural diversity and generalization accuracy using Shannon and Simpson diversity measures to quantify diversity. The investigation is necessary as to build learning machines or committee of networks that can generalize better. The results have clearly shown that as the structural diversity index based on the measures used increases, the ensemble accuracy increases. Hence, the classifiers can be made structurally different in order to gain good classification accuracy. This has brought an increase of 3% to 6% in the classification accuracy. The method used to compute the results was found to be computationally expensive due to the use of GA. There is however limitations brought about by the individual classifiers producing similar outcomes even though they are structurally different. However, the use of measuring structural diversity in building good ensembles of classifiers is still to be explored.

9. References

[1] G. Brown, J. Wyatt, R. Harris and X .Yao. "Diversity Creation Methods: A Survey and Categorization," *Journal of information Fusion*, pp 5-20, Vol. 6, No. 1, 2005.

[2] J. Sylvester, N.V. Chawla, "Evolutionary Ensemble Creation and Thinning", Proc. Of International Joint Conference on Neural Networks, pp 5148-5155, 2006.

[3] N.V. Chawla, J. Sylvester, "Exploiting Diversity in Ensembles: Improving the Performance on Unbalanced Datasets", *Multiple Classifier Systems*, Lecture Notes in Computer Science, Springer, Vol. 4472, pp 397-406, 2007.

[4] Y. Kima, W.N. Street, Filippo Menczer, "Optimal ensemble construction via meta-evolutionary ensembles", *Expert Systems with Applications,* Vol. 30, No. 4, pp 705-714, 2006.

[5] L. Masisi, F.V. Nelwamondo, T. Marwala,"The effect of structural diversity of an ensemble of classifiers on classification accuracy", *IASTED International Conference on Modelling and Simulation (Africa-MS)*, pp 1-6, 2008.

[6] L.I Kuncheva, C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy", Machine Learning, Vol. 51, No. 2, pp 181–207, 2003.

[7] L. I. Kuncheva and C. J. Whitaker, "Ten measures of diversity in classier ensembles: limits for two classiers", *In Proc. of IEE Workshop on Intelligent Sensor Processing,* pp 1-10, 2001.

[8] R. Polikar, "Ensemble based system on decision making", *IEEE Circuit and System Magazine*, pp 21-45,

[9] C.A Shipp, L.I Kuncheva, "Relationship between combination methods and measures of diversity in combining classifiers", *Information Fusion*, Vol. 3, No.2, pp 135-148, 2002

[10] A. Lipnickas, "Classifiers fusion with data dependent fusion with data dependent aggregation schemes", *International Conference on Information Networks, Systems and Technologies*, pp147-153, 2001

[11] M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science and Business Media, 2006

[12] T Marwala. *Bayesian Training of Neural Networks Using Genetic Programming*. Pattern Recognition Letters, Vol. 28, pp. 1452-1458, 2007.

[13] South African Department of Health: HIV and Syphilis Sero-Prevalence Survey of Women Attending Public Antenatal Clinics in South Africa, http://www.info.gov.za/view/DownloadFileAction?id=70247, last accessed 10 November 2008.

[14] I.T Nabney. Netlab: Algorithms for Partten Recognition. Springer, 2001.

[15] B.B Leke, T. Marwala, T. Tettey, "Autoencoder networks for HIV classification", *Current Science*, Vol. 91, No. 11, 2006.

[16] D.G. Mcdonald, J. Dimmick, "The conceptualization and Measurement of Diversity", *Communication Research,* SAGE publications, Vol. 30, No. 1, pp 60-79, 2003

[17] L. Masisi, F.V. Nelwamondo, T. Marwala, "The use entropy measures to measure the structural diversity of an ensemble of classifiers via the use of Genetic Algorithm", *School of Electrical and information Engineering Witwatersrand University, ICCC,* 2008,accepted

[18] D. Opitz, R. Maclin, "Popular Ensemble Methods", *Journal of Artificial Intelligence Research,* Vol. 11, No. 8, pp 169-198, 1999.

[19] A. Lipnickas, "Classifiers fusion with data dependent fusion with data dependent aggregation schemes", *International Conference on Information Networks, Systems and Technologies,* ICINASTe, page 147-page 153, 2001

[20] W.D. Penny, S.J. Roberts."Bayesian Neural networks for Classification: How useful is the Evidence Framework," *Neural Networks,* Vol. 12, No 1, pp.877-892, 1999

A channel normalization technique for speech recognition in mismatched conditions

Neil Kleynhans and Etienne Barnard

Department of Electrical, Electronic and Computer Engineering
University of Pretoria, South Africa
ntkleynhans@csir.co.za and
Human Language Technologies (HLT) Group
Meraka Institute
ebarnard@csir.co.za

Abstract

The performance of trainable speech-processing systems deteriorates significantly when there is a mismatch between the training and testing data. The data mismatch becomes a dominant factor when collecting speech data for resource scarce languages, where one wishes to use any available training data for a variety of purposes. Research into a new channel normalization (CN) technique for channel mismatched speech recognition is presented. A process of inverse linear filtering is used in order to match training and testing short-term spectra as closely as possible. Our technique is able to reduce the phoneme recognition error rate between the baseline and mismatched systems, to an extent comparable to the results obtained by the widely-used cepstral mean subtraction. Combining these techniques gives some additional improvement.

1. Introduction

In this paper, we investigate a channel normalization technique that reduces the speech data channel mismatch between varied sources by estimating the average short-term spectral energy and then filtering the speech data with an appropriate mapping filter.

Any mismatch between training and testing speech data significantly degrades the performance of trainable speech-processing systems. The mismatch is introduced by physical processes such as background noise, non-stationary noise, recording transducers and transmission channels, as well as population differences such as speaker dialects, age and gender distributions, etc. Only the combined effect of these varying processes are generally observable in the data; therefore all these effects are treated as one "channel" mismatch process. Once a mismatch has been identified, channel normalization techniques are employed to reduce the effect it has on the speech system. Such issues are often dealt with by recording sufficiently variable training data, but the penalty introduced by the channel mismatch becomes critical when a resource scarce language is used. One of the major problems in dealing with resource scarce languages is that collecting speech data is expensive and the amount of data is not comparable to that traditionally used for global languages. One method to reduce the impact of data scarcity is to use different recording devices such as cellular phones, land-line phones and computer microphones. However, this method would inevitably introduce a channel mismatch. Thus, an effective channel normalization technique is needed to satisfactorily reduce the channel mismatch. Ideally, one would want the speech system to behave as if the speech data originated from one source.

There are many strategies that are used to minimize the effect of channel mismatch. In the fortunate case that speech data is available from all the channels, channel-dependent acoustic models can be trained or existing acoustic models could be adapted to better handle incoming speech data. Even though this strategy works the best, it is rare that enough speech data is available to develop robust acoustic models for each channel. In the speech signal domain, blind channel estimation and inverse filtering have been used to reduce the channel influence on the speech data [1]. However, it is difficult to make assumptions about the channel response and spectral nature of speech data. Experiments have shown that if a non-linear channel response is encountered, the blind channel estimation technique did not provide an increase in recognition accuracy [1].

Feature vector mapping tries to overcome the channel mismatch by treating the channel effect as feature transformation in the model domain [2, 3, 4]. More traditional techniques are Cepstral mean subtraction (CMS) and Relative spectra (RASTA) filtering [5, 6, 1]. CMS subtracts a long-term average cepstral component from each extracted cepstral component. This method has gained significant popularity in speech and speaker recognition systems for removing slow-varying channel changes [7], but a small amount of speech information is also removed [1]. The CMS method can only be used in speech-based systems that use cepstral feature vectors to represent the speech data. The RASTA filtering method applies a filter that rejects spectral components that move too slowly or quickly compared to the normal rate of change of speech spectral components [7]. However, RASTA filtering violates the standard hidden Markov model (HMM) assumption of piecewise stationary [6] and introduces phase distortion [5], which negatively impacts on recognition accuracies. The simple CMS technique has been proved equally as good as phase corrected RASTA for telephony experiments [5].

Based on the previous work done, the three main criteria that were used to develop a new channel normalization technique, were:

- a resource scarce language environment is assumed, therefore generating channel-dependent acoustic models becomes impractical,
- more complex channel normalization techniques afford little benefit over simpler methods, and

- feature vector independence is required in order to benefit a variety of systems.

The CMS technique meets two of the three criteria; therefore it was used as a baseline channel normalization method. The new channel normalization technique should provide a performance gain over no normalization and the resulting error rate of the mismatch data system should be similar to the error rate given by a CMS implementation.

2. Method

As in speech parametrization techniques, which encode short-term speech information, an initial step was to calculate the average short-term spectral energy over the frames of speech. The frame length was chosen to roughly ensure stationarity of the signal, shifted to create overlap between adjacent frames and each frame windowed. Given frames of speech, $X_i^N = \{X_1, X_2, ..., X_N\}$, the average short-term spectral energy is calculated as

$$Y_c(f) = \frac{1}{N} \sum_{i=1}^{N} \mid H_c(f) X_i(f) W_{HAM}(f) \mid^2 \qquad (1)$$

where $H_c(f)$ is the channel frequency response, $X_i(f)$ represents the frame level spectrum and $W_{HAM}(f)$ is the Hamming window frequency response.

It is assumed that the filter response is linear and time-invariant, therefore remaining constant across the frames of speech and speakers in the database. It would be a difficult task to calculate the channel response using just this information, but the goal here is not to determine the most probable frequency response. The desire is to transform the data from a channel, to better match a channel with a different response, through the use of inverse filtering. An approximation of the mapping filter can be found, if the ratio between two average short-term spectral energies is calculated:

$$\widetilde{H}_{Inv}(f) = \frac{\mid H_{C1}(f) \mid^2 \sum_{i=1}^{N} \mid X_i(f) \mid^2}{\mid H_{C2}(f) \mid^2 \sum_{j=1}^{N} \mid X_j(f) \mid^2} \qquad (2)$$

If the assumption is made that the speech characteristics are similar across the data collected from varying channels, the difference that is present in the energy distribution is directly as a result of the channel responses. Figure 1, shows the average short-term spectral energy for a subset of data collected from TIMIT and Wall Street journal corpora, which demonstrates a clear difference in the spectral energy distributions.

The assumption that the speech characteristics are similar across corpora could easily be in error. For instance, the phonetic distribution could be skewed, which would result in more energy being present in certain frequency bands. Therefore, as an average short-term spectral energy estimation improvement, confining the estimator and inverse filter calculation to broad phonetic classes should improve the assumption that the speech characteristics of the two sources are similar; we report on experiments involving both the basic idea and the refined approach below.

3. Experiments

A triphone-based HMM phoneme recognizer, developed using the Cambridge University HMM Toolkit (HTK) [10], was used to perform a variety of channel normalization experiments. The task we used for our benchmarking experiments was phone recognition; this allows us to focus on acoustic modelling exclusively. The two corpora chosen for experimentation were TIMIT and Wall Street Journal (WSJ). A difference in channel characteristics can be expected due to the varied recording environments (room acoustics) and setup (type of microphone used to record the utterances). The sample average short-term spectral energy distribution for the two corpora are shown in Figure 1.

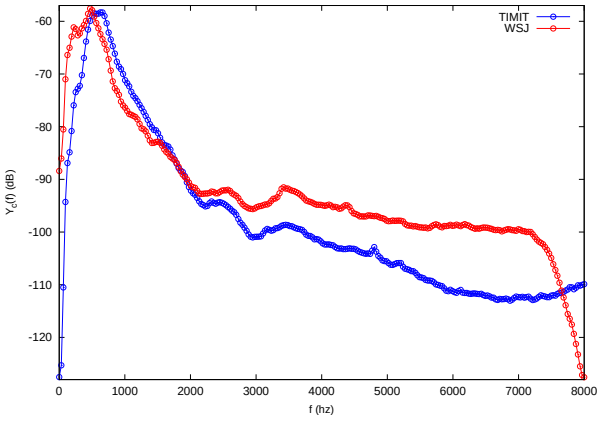


Figure 1: Average short-term spectral energy calculated from a subset of data using the TIMIT[8] and Wall Street Journal[9] corpora.

Task	# Speaker	# File	# Minutes
Recognizer Training	462	2772	143
Recognizer Testing	168	1344	69
Channel Estimator	462	924	46
Broad Classifier Training	462	924	46

Table 1: TIMIT corpus statistics.

The data from both corpora was partitioned into separate sets for phoneme recognizer training and testing, where no speakers were in common between the sets (though for TIMIT, certain sentences did occur in both). The same data was used for channel estimation and training the broad phonetic class classifier; this data was obtained and removed from the phone recognizer training data. The phone recognizer testing data was used to verify the accuracy of the broad phonetic class classifier. The broad phonetic class classifier used six classes: consonants, fricatives, glides, nasals, stops and vowels. Silence was an additional class, but was ignored in the mapping filter calculations.

The TIMIT corpus partitioning statistics are shown in Table 1, while those for the WSJ corpus are given in Table 2.

A number of experiments were run using different channel normalization techniques. The accuracy results are shown in Table 3, which used TIMIT trained acoustic models, and in

Task	# Speaker	# File	# Minutes
Recognizer Training	77	2404	275
Recognizer Testing	24	914	103
Channel Estimator	77	707	88
Broad Classifier Training	77	707	88

Table 2: WSJ corpus breakup statistics.

System Type	Testing Data	
	TIMIT	WSJ
PR	56.80	
BPCC	74.92	
NO NORM		45.30
CMS		51.42
AVG		49.89
CMS + AVG		52.90
SEGRAT		50.51
SEGLS		49.37
COMB		60.48

Table 3: Recognition accuracy results using models trained with TIMIT data.

System Type	Testing Data	
	TIMIT	WSJ
PR		62.03
BPCC		71.69
NO NORM	52.29	
CMS	55.23	
AVG	56.65	
CMS + AVG	56.50	
SEGRAT	56.88	
SEGLS	51.71	
COMB	61.92	

Table 4: Recognition accuracy results using models trained with WSJ data.

Table 4, which contains results for WSJ acoustic models. The system type codes given in Tables 3 and 4 are as follows;

- **PR** - Phoneme Recognizer acoustic models trained and tested using channel-specific data - i.e TIMIT only or WSJ data only.
- **BPCC** - Broad Phonetic Class Classifier, which was trained using unique channel-specific data. PR testing data was used to obtain the accuracy results.
- **NO NORM** - The channel-specific phoneme recognizer was used to decode the unseen channel testing data, e.g. TIMIT trained models decoding WSJ testing data.
- **CMS** - Cepstral Mean Subtraction used by HTK to remove a mean cepstral vector from a set of cepstral vectors extracted from one speech file. The process was applied to both the training and testing data.
- **AVG** - The average short-term spectral energy from each channel was used to derive the mapping filter. The estimation was calculated using unique channel estimation data.
- **SEGRAT** - The channel estimation data was segmented using the BPCC system, which was then used to generate six class-specific average short-term spectral energy estimates. The mapping filter was derived from the average estimates.
- **SEGLS** - Same as SEGRAT, except that the mapping filter was derived using a least squares fit between the six average estimates.
- **COMB** - PR acoustic models were trained using data from both channels. No channel normalization methods were used.

4. Discussion

A 5% difference in the corpus-specific phoneme recognizer (PR) results can be explained by the greater number of speakers found in the TIMIT corpus and a larger amount of speech data per speaker in the WSJ corpus. However, the TIMIT phoneme recognition accuracy did improve when the WSJ training data was added to the acoustic model training phase. This improvement was not observed when these acoustic models were tested with the WSJ data. This may indicate that the PR TIMIT acoustic models require much more data to approach the stability of the PR WSJ acoustic models. Considering the channel normalization experiments, the channel-specific acoustic model (PR) results gave an upper bound with which to compare the results obtained from the varying channel normalization tests. The COMB experiment accuracies gave an upper bound for the complete system, and could be considered as an upper bound that can be achieved when both channel normalization and normalization for other factors discussed in Section 1 are employed.

When no channel normalization techniques were used, the phoneme recognizers drop in performance by 10%, which was to be expected. With TIMIT training, the HTK CMS method reduced the drop in accuracies by 5%; that is, about half of the loss is recovered. For WSJ training, only about 30% of the cross-channel loss is recovered with CMS. The average short-term spectral energy filtering method (AVG) gave similar improvements to CMS, being somewhat better for WSJ and somewhat worse for TIMIT. When the CMS and AVG methods were combined (CMS+AVG) and applied to the testing dataset, an improvement in performance was observed compared to the CMS results; now, about 60% of the cross-channel loss is recovered for TIMIT training, and 45% for WSJ training. The AVG and CMS methods can be seen to perform approximately the same task, where AVG modifies the speech waveform and CMS transforms the cepstral coefficients.

The more elaborate BPCC segmentation system gave only small improvements compared to the basic AVG method. Our least-squares approach was clearly not successful, but the SEGRAT was slightly better on both corpora. The statistical significance of the SEGRAT results, compared to the AVG results, were measured using McNemar's test with a chi-squared statistic and the McNemar table of values setup found in Gillick and Cox [11]. A large statistical significance ($P < 0.000001$) was found for the TIMIT trained acoustic models, while the gain obtained for the WSJ trained acoustic models was insignificant ($P < 0.61$). However, many other sensible ways to combine the filters obtained for the different broad phonetic classes remain to be explored. We are therefore confident that the small observed improvement points the way towards even more successful methods.

5. Conclusion

The adverse effect of recording speech data on different channels was demonstrated using the TIMIT and WSJ corpora. A channel normalization technique, which derives a mapping filter from the average short-term spectral energy estimates was shown to give results comparable to the cepstral mean subtraction method. The benefit provided by the new technique is that it is applied to the speech waveform and is therefore indepen-

dent of the chosen speech parametrization calculations. The broad phonetic class classifier approach provided a small boost to the performance, but the additional work required to implement this method is not justified. However, the marginal improvement was surprising, which indicates that further experimentation must be done to determine how the channel estimation and filtering should be combined to increase the system's performance. The best experimental results that were obtained, came from the case where the acoustic models were trained with speech data from both channel datasets. During the training process, the means and variances of phonetic models are updated to better represent the observed data, therefore a channel normalization process that can translate a model space transform to the speech signal domain should theoretically provide performance enhancements comparable to updated phonetic models. This approach will be further investigated.

6. References

[1] S.J. Wenndt and A.J. Noga, "Blind channel estimation for audio signals", in Proceedings of the Aerospace Conference, March 2004, vol. 5, pp. 3144-3150.

[2] D. Kim and D. Yook, "Feature transform in linear spectral domain for fast channel adaptation", IEE Electronics Letters, vol. 40, no. 20, pp. 1313-1314, September 2004.

[3] Y-F Liao, J-S Lin and S-H Chen, "A Mismatch-Aware Stochastic Matching Algorithm For Robust Speech Recognition", in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2003, vol. 2, pp. 101-104.

[4] B. Theobald, S. Cox, G. Cawley and B. Milner, "A Fast Method of Channel Equalisation for Speech Signals and its Implementation on a DSP", IEE Electronics Letters, vol. 35, no. 16, pp. 1309-1311, August 1999.

[5] J. de Veth and L. Boves, "Channel normalization techniques for automatic speech recognition over the telephone", Speech Communication, vol. 25, no. 1-3, pp. 149-164, August 1998.

[6] H. Bourlard, H. Hermansky and H. Morgan,"Towards increasing speech recognition error rates", Speech Communication, vol. 18 , no. 3, pp. 234-235, May 1996.

[7] H. Hermansky and N. Morgan, "RASTA processing of speech", IEEE Trans. Speech Audio Process., vol. 2, no. 4, pp. 578-589, October 1994.

[8] "The DARPA TIMIT Acoustic-Phonetic Continuous speaker space Speech Corpus" (CD-ROM), Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1 (Last Accessed: 3 October 2008)

[9] "The DARPA Continuous Speech Recognition Corpus II: Wall Street Journal Sentences" (CD-ROM), Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC94S13A (Last Accessed: 3 October 2008)

[10] S. Young, "Large Vocabulary Continuous Speech Recognition.", IEEE Signal Process. Mag., vol. 13, no. 5, pp. 45-57, April 1996.

[11] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms", in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 1989, vol. 1, pp. 532-535.

3D Phase Unwrapping of DENSE MRI Images Using Region Merging

Joash N. Ongori, Ernesta M.Meintjes, Bruce S. Spottiswoode

MRC/UCT Medical Imaging Research Unit, Department of Human Biology, Faculty of Health Sciences
University of Cape Town, Observatory, 7925, South Africa
ongjoa001@uct.ac.za

Abstract

Cine displacement encoding with stimulated echoes (cine DENSE) is an MRI technique that encodes displacement over a time series into the phase of complex MRI images. Phase aliasing is unavoidable and this phase needs to be unwrapped to determine the displacement fields. Phase unwrapping is complicated by image noise, phase shear and the fact that only a few pixels span the myocardial walls. This work investigates the effectiveness of a 3D cost function based region merging method for unwrapping cine DENSE images. The new technique is shown to provide comparable accuracy to an existing technique.

1 Introduction

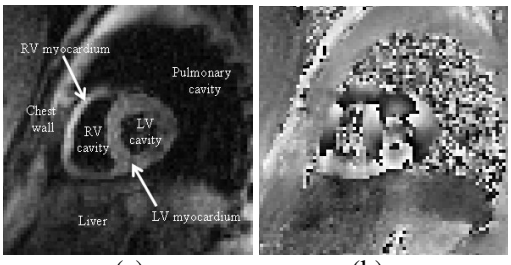
Cardiovascular disease is the leading cause of death in many developed countries. The ability to quantify the motion of the heart muscle, or myocardium, is valuable for understanding both normal and diseased cardiac kinematics [1].

Magnetic resonance imaging (MRI) is a powerful tool for imaging the heart. Cardiac MRI is superior to other modalities in imaging myocardial mechanics. Computed tomography (CT) methods rely on movement of the heart boundaries and they give limited insight into intramyocardial motion. Radionuclide single-photon emission CT (SPECT) and positron emission tomography (PET) provide valuable information about myocardial metabolism but do not reliably measure myocardial motion. Doppler ultrasound is capable of measuring myocardial velocities and strain rates, but the technique has a low spatial resolution and there are limited anatomical viewing windows.

MRI is capable of monitoring intramyocardial motion in any imaging plane using a variety of techniques including myocardial tagging [2, 3] phase contrast velocity encoding [4], harmonic phase (HARP) [5], and most recently displacement-encoded imaging using stimulated echoes (DENSE) [6].

DENSE measures the motion of myocardial tissue by encoding displacement in a particular direction into the phase of the complex MRI image. Cine DENSE [7] measures myocardial displacement throughout the cardiac cycle.

DENSE provides a better spatial resolution than myocardial tagging, and a greater tissue tracking accuracy than velocity encoding. Example magnitude and phase cine DENSE images are shown for end-systole in Figure 1a and 1b, respectively. The phase images are confined to the range $[-\pi,\pi]$ and unavoidable phase aliasing occurs in the walls of the heart as it contracts.

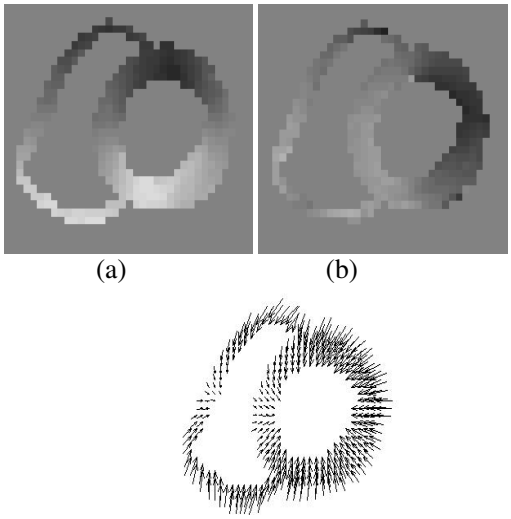


(a) (b)

Figure 1: (a) DENSE magnitude short axis view of the heart depicting the anatomical structures, and (b) the corresponding phase image where the displacement is encoded and constrained between $-\pi$ (black) and π (white). LV – left ventricle; RV – right ventricle.

Phase unwrapping is required to determine displacement fields. Figure 2a and 2b show unwrapped phase images for respective motion encoded vertically and horizontally. In Figure 2a white represents upward motion and black represents downward motion, and in Figure 2b

white represents motion to the right and black represents motion to the left. The corresponding displacement field is shown in Figure 2c. These can be used to derive strain by applying finite element methods. These strains can then be used to discern between healthy and diseased myocardium, and to assess mechanical dyssynchrony [8].



(a) (b)

(c)

Figure 2: (a) Unwrapped phase image for vertical motion, (b) unwrapped phase image for horizontal motion, and (c) the corresponding displacement field.

All phase unwrapping algorithms require a known phase reference point. In cine DENSE, phase aliasing is not present in early systole, at the beginning of the cine series. Referencing a true phase value can thus be achieved by unwrapping in three dimensions, i.e. two spatial and one temporal dimension [10]. This paper discusses an alternative 3D phase unwrapping technique for 2D cine DENSE data sets.

2 Phase unwrapping

Phase unwrapping is the process of determining the absolute phase given its principal value. The relationship between the measured, or wrapped, phase ϕ_j and the actual phase θ_j is

$$\theta_j = \phi_j + 2\pi m_j \qquad (1)$$

where j is an N-dimensional index specifying the spatial location and m_j at each voxel specifies the corrective offset required. The phase unwrapping problem reduces to determining m_j for each voxel in an image. Phase unwrapping in cardiac MRI images is complicated by image noise, phase shear and the fact that only a few pixels span the myocardial walls.

Existing techniques for phase unwrapping can be grouped according to their

(i) Dimensionality (1D, 2D, 3D etc);
(ii) Application (Synthetic Aperture Radar, general optical interferometry, MR angiography, MR chemical shift mapping or MR field mapping); or
(iii) Approach (fitting functions, cost function optimisation, filtering, region growing / merging).

Phase unwrapping algorithms typically fall into two classes:

Path–following algorithms. These use localized operations by following paths through the wrapped phase. Variations include Goldstein's algorithm, Quality-guided algorithms, Mask Map algorithm, and Flynn's Minimum Discontinuity algorithm [9].

Minimum– norm algorithms. These adopt a more global minimisation approach and include unweighted least-squares algorithm, pre-conditioned conjugate gradient (PCG) algorithm, weighted multigrid algorithm, and Minimum L^p - norm algorithm [9].

Previously, 2D cine DENSE images have been analysed using a quality-guided (QG) path following algorithm that unwraps the phase through both space and time by using a measure of phase quality to guide the path of unwrapping [10].

This method can be summarised as follows.

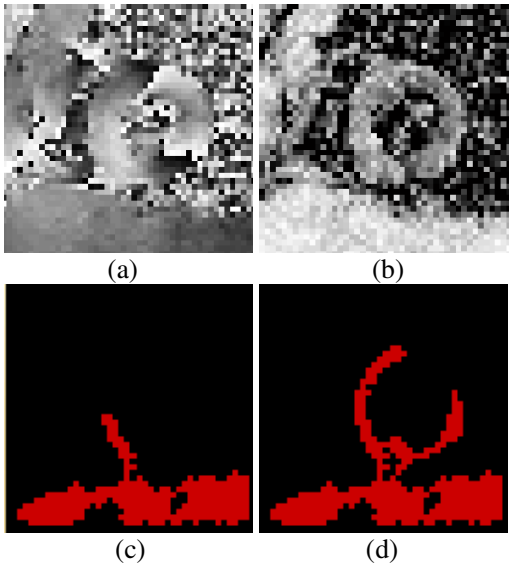
1. A measure of phase quality for each pixel is calculated by

$$Z_{pqr} = \frac{\sqrt{\Sigma(\Delta^{x_1}_{i,j,k} - \overline{\Delta}^{x_1}_{p,q,r})^2} + \sqrt{\Sigma(\Delta^{x_2}_{i,j,k} - \overline{\Delta}^{x_2}_{p,q,r})^2} + \sqrt{\Sigma(\Delta^{x_3}_{i,j,k} - \overline{\Delta}^{x_3}_{p,q,r})^2}}{n^3} \qquad (2)$$

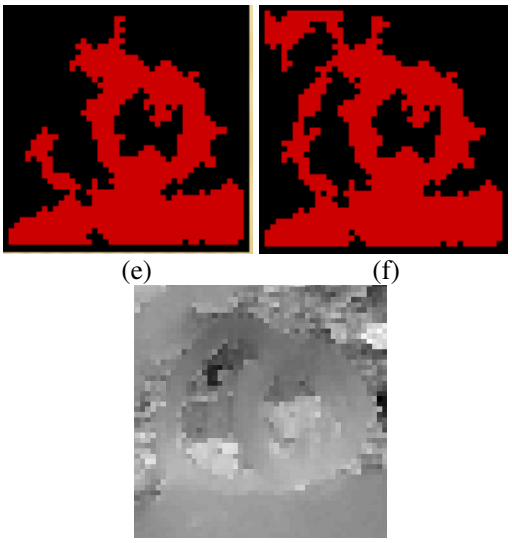
where for each sum the indexes *(i, j, k)* range over the $n \times n \times n$ window centered at the pixel *(p,q,r)*. The terms $\Delta^{x_1}_{i,j,k}$, $\Delta^{x_2}_{i,j,k}$, and $\Delta^{x_3}_{i,j,k}$ are the partial derivatives of the locally unwrapped phase, and the terms $\overline{\Delta}^{x_1}_{p,q,r}$, $\overline{\Delta}^{x_2}_{p,q,r}$, and $\overline{\Delta}^{x_3}_{p,q,r}$ are the averages of these partial derivatives in the $n \times n \times n$ windows.

2. A starting point with known phase is selected and stored in a solution matrix.
3. The four pixels adjacent to the starting point are placed in an 'adjoin' matrix, which keeps track of wrapped pixels with adjacent unwrapped pixels.
4. The pixel in the adjoin matrix with the highest phase quality is selected and unwrapped using its adjacent unwrapped pixel. This pixel is removed from the adjoin matrix and added to the solution matrix.
5. The new wrapped nearest neighbours are included in the adjoin matrix.
6. Steps 4 and 5 are repeated until the adjoin matrix is empty.

Figure 3a and 3b shows the wrapped phase image and corresponding phase quality map, respectively. Figure 3c to 3f show the phase unwrapping path flooding from regions of high to low phase quality, and Figure 3g shows the resulting unwrapped image. The unwrapped image is smooth with no 2π phase transitions in the myocardium.



(a) (b) (c) (d)



(e) (f) (g)

Figure 3: (a) Phase image, (b) corresponding phase quality map, (c-f) the progression of unwrapping in the quality guided (QG) method, and (g) unwrapped phase image

3 Methods

Phase Region Expanding Labeller for Unwrapping Discrete Estimates (PRELUDE) is an *N*-dimensional region merging phase unwrapping algorithm developed for MR images [11]. Here an image is divided into a number of regions corresponding to specific phase brackets. Neighbouring regions are interrogated and merged based on a cost function.

To penalise the phase differences along the interfaces the sum of the square of the phase difference along the interface is used, that is

$$C_{AB} = \sum_{j,k \in N(j)} \left(\phi_{Aj} - \phi_{Bk} + 2\pi M_{AB} \right)^2 \qquad (3)$$

where $M_{AB} = M_A - M_B$ with M_A and M_B being integer offsets for adjacent regions A and B. The summation is taken over two indices, *j* and *k*, where *j* is the index of a voxel in region *A*, while *k* is the index of a voxel in region *B*. The total cost over the whole (*N*-dimensional) volume is the sum over all the interfaces: $C = \sum_{A,B} C_{AB}$

Differentiating by the parameters, M_{AB} gives the equation for the minimum cost solution

$$M_{AB} = M_A - M_B = \left(\frac{-P_{AB}}{2\pi N_{AB}} \right) \quad (4)$$

where N_{AB} is the number of interfacing voxel pairs and $P_{AB} = \sum_{j,k \in N(j)} (\phi_{Aj} - \phi_{Bk})$

In order to solve the integer programming problem generated by Equation 4 we can treat it as follows

$$M_{AB} = round \left(\frac{-P_{AB}}{2\pi N_{AB}} \right) \quad (5)$$

so that we can get a low (ideally minimum) cost.

Let, $K_{AB} = -P_{AB} / (2\pi N_{AB})$ and

$$L_{AB} = round(K_{AB}) \quad (6)$$

The difference in cost between $M_{AB} = L_{AB}$ and $M_{AB} = L_{AB} \pm 1$ is

$$\Delta C_{AB} = 8\pi^2 N_{AB} \left(\frac{1}{2} \pm (K_{AB} - L_{AB}) \right) \quad (7)$$

Since $| K_{AB} - L_{AB} | \leq \frac{1}{2}$, then $\Delta C_{AB} \geq 0$ for both cases, confirming that $M_{AB} = L_{AB}$ is the local minimum, with the smallest cost difference being

$$\Delta C_{AB} = 8\pi^2 N_{AB} \left(\frac{1}{2} - | K_{AB} - L_{AB} | \right) \quad (8)$$

This implementation involves the following steps:

1. Determine masks to dichotomise the myocardium from the background noise in the blood pools and lung cavity. This was done using both image magnitude and phase discontinuities [12]. Phase inconsistencies, or residues, were identified by integrating the phase in small 4-pixel loops, and removing the pixel of lowest magnitude that lies adjacent to each residue. This provided the basis for identifying a threshold level for the magnitude image.
2. Create initial connected regions which will be merged during the unwrapping process. The myocardium was phase partitioned into regions according to the intervals {[-π, -2π/3], [- 2π/3, - π/3], [-π/3, 0], [0, π/3], [π/3, 2π/3] and [2π/3, π]}.
3. Identify the pair of regions that has the largest border weight (ΔC). The pair is selected according to Equations 6 and 8, and $AB = \arg\max_{AB} \Delta C_{AB}$, where A and B are adjacent regions with the highest border weights. Merge the two regions by adding $2\pi L_{AB}$ to region B.
4. Update the statistics for all interfaces (with other regions) to this new region. This involves updating the matrices P_{AB}, N_{AB} and ΔC.
5. Select a new pair of regions to merge. Region merging continues until there are no more interfaces.

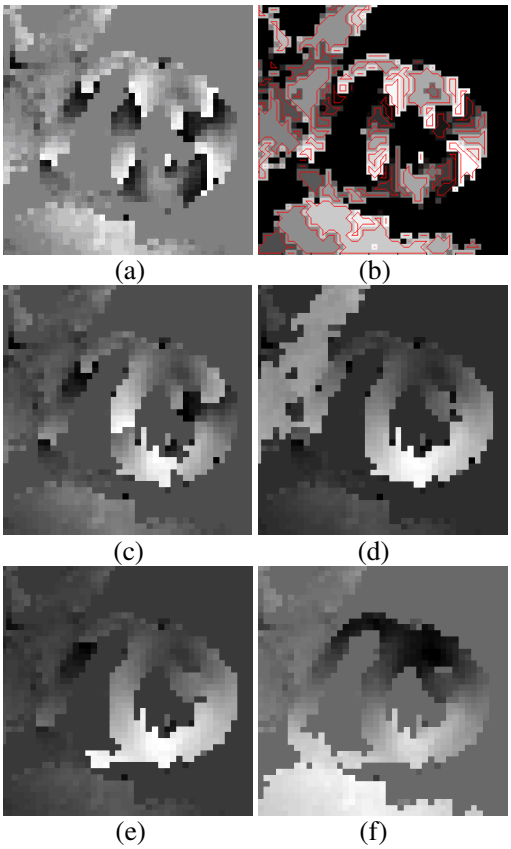


Figure 4: The PRELUDE unwrapping process. (a) Masked phase image, (b) partitioned phase image, (c-f) progression of unwrapping phase image after 15, 55, 90, and 115 iterations, respectively, and (f) unwrapped phase image after all regions have been merged.

Figure 4 illustrates the PRELUDE unwrapping process. The regions corresponding to the masked phase image in Figure 4a are shown in Figure 4b. Figure 4c to 4f shows the progression of the

unwrapping process, and Figure 4f shows the final unwrapped image.

Figure 5 depicts the maximum border weight for pairs of adjacent regions during the unwrapping process for Figure 4. The sharp increases occur when two merged regions yield a new united border with a higher border weight.

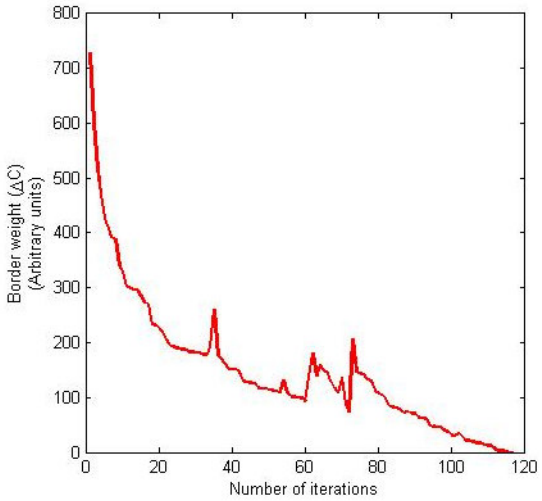
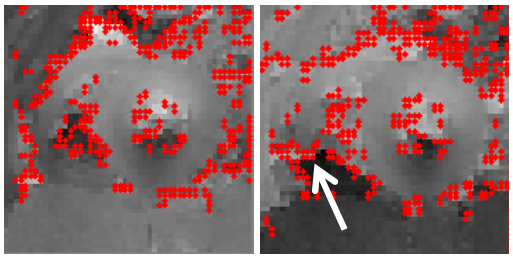


Figure 5: Maximum border weight ΔC as a function of the number of iterations.

The PRELUDE algorithm was implemented in MATLAB (The Mathworks, Natick, MA) and run on an Intel® Core™ 2 Duo processor with 2GB of RAM. The reliability of the PRELUDE algorithm was compared to the quality guided (QG) algorithm on 300 cine DENSE images. Phase unwrapping errors were identified visually with the assistance of discontinuity maps [9]. A discontinuity map highlights pixels where an adjacent pixel contains a phase offset greater than $\pi/2$ or less than $-\pi/2$. If a line of discontinuities is seen to span the walls of the left or right ventricles, then the image is deemed incorrectly unwrapped. An example is shown in Figure 6.



(a) (b)

Figure 6 (a) Unwrapped phase image with no discontinuities spanning the myocardial walls, (b) Unwrapped phase image with discontinuities across the right ventricle (white arrow).

4 Results

The results are summarised in Table 1 below. The PRELUDE technique provides comparable results to the quality-guided algorithm but the processing time is slower.

Table 1: Comparison of 3D quality-guided and PRELUDE phase unwrapping algorithms. LV-left ventricle; RV – right ventricle.

	Quality-guided	PRELUDE
LV correctly unwrapped	99.3 %	99.0 %
RV correctly unwrapped	73.3 %	74.6 %
Processing time per frame	< 0.6 s	≤ 90 s

5 Discussion and Conclusions

There are several limitations to the PRELUDE technique. Partitioning the image into phase bracket should be done judiciously. If a single region includes two areas where the original phase differs by more than 2π, the algorithm can never successfully recover this original phase difference. Secondly, to obtain reasonable processing speeds the method is reliant on a mask to remove background noise.

The PRELUDE algorithm was considerably faster than the quality-guided algorithm for 2D phase unwrapping, but the 3D extension of PRELUDE results in prohibitively long processing times. Alternative phase

unwrapping methods will be required for volumetric cine DENSE studies [13] where 4D phase unwrapping is required.

Phase unwrapping is particularly challenging for the right ventricle, where the myocardial wall thickness is similar in width as the cine DENSE pixel size.

Spatio-temporal phase unwrapping remains a challenge for cine DENSE, but it is an unavoidable step in computing displacement fields which can then be used to determine strain patterns within the wall of the heart. This will help to detect abnormal wall motion which could be useful in the diagnosis of heart disease.

6 References

[1] Montillo A., Metaxas D. and Axel L. Automated segmentation of the left and right ventricles in 4D cardiac SPAMM images. In: International Society and Conference Series on Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS 2488: 620–633, 2003.

[2] Zerhouni E.A., Parish D.M., Rogers W.J.,Yang A. and Shapiro E.P. Human Heart: tagging with MR imaging - a method for non-invasive assessment of myocardial motion. Radiology 169(1): 59-63, 1988.

[3] Axel L. and Dougherty L. MR imaging of motion with spatial modulation of magnetization. Radiology 171: 841-845, 1989.

[4] Pelc N.J., Drangova M., Pelc L.R., Zhu Y., Noll D.C., Bowman B.S. and Herfkens R.J. Tracking of cyclic motion with phase – contrast cine MR velocity data. Journal of Magnetic Resonance 5:339-45, 1995.

[5] Osman N.F. and Prince J.L. Imaging heart motion using harmonic phase MRI. IEEE Transactions on Medical Imaging 19(3):186-202, 2000.

[6] Aletras A.H., Ding S., Balaban R., and Wen H. DENSE: Displacement encoding with stimulated echoes in cardiac functional MRI. Journal of Magnetic Resonance 137:247-252, 1999.

[7] Kim D., Gilson W.D., Kramer C.M. and Epstein F.H. Myocardial tissue tracking with two-dimensional cine displacement-encoded MR imaging: Development and initial evaluation. Radiology 230:862-871, 2004.

[8] Patel R.A., Zhong X., Spottiswoode B.S. *et al*. Cine DENSE MRI of left ventricular dyssnchrony: Development and initial clinical experience. In: Proceedings of the 14th International Society for Magnetic Resonance in Medicine (ISMRM), #3600, 2006.

[9] Ghiglia D.C. and Pritt M.D. Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software. New York: Wiley-Interscience, 1998.

[10] Spottiswoode B.S., Zhong X., Hess A.T., Meintjes E.M., Mayosi B.M. and Epstein F.H. Tracking myocardial motion from cine DENSE images using spatiotemporal phase unwrapping and temporal fitting. IEEE Transactions on Medical Imaging, 26(1):15-30, 2007.

[11] Jenkinson M. Fast, automated N-dimensional phase-unwrapping algorithm. Magnetic Resonance in Medicine 48:193-197, 2003.

[12] Sinele K., Bennett E. and Wen H. Automatic masking and phase unwrapping of DENSE myocardial tissue tracking images in human. Journal of Cardiovascular Magnetic Resonance 8(1): 352, 2006.

[13] Zhong X., Spottiswoode B.S. and Epstein F. H. Myocardial tissue tracking using volumetric cine DENSE with 3D displacement encoding – development and preliminary results. In: Proceedings of the 15th International Society for Magnetic Resonance in Medicine (ISMRM), #3599, 2007.

Fast Calculation of Digitally Reconstructed Radiographs using Light Fields

Cobus Carstens

Department of Medical Radiation
iThemba LABS
Faure
cobus.carstens@gmail.com

Neil Muller

Division of Applied Mathematics
Department of Mathematical Sciences
University of Stellenbosch
neil@dip.sun.ac.za

Abstract

This study aims to improve the calculation time of DRRs for use in intensity based 2D-3D Image Registration. Carstens and Muller [1] showed that image order algorithms, which are trivially parallelisable, can easily be adapted to take advantage of hardware systems with more than one CPU. The ray casting algorithm and light field rendering were found to be suitable for this purpose. A discussion of the ray cast algorithm and light field rendering is presented and followed by performance measurements. A significant performance increase is achieved when using the parallelised light field algorithm over the serial ray casting algorithm. The DRRs calculated using the light field algorithm are also shown to be feasible for use in an image registration algorithm.

1. Introduction

The current patient positioning system used at iThemba LABS uses a close-fitting, patient specific mask with markers located on it. A CT scan of the patient is taken with the mask fitted to the patient in order to establish a relationship between the marker positions and the anatomy of the patient. A real-time stereophotogrammetry (SPG) system is used to detect the markers in the treatment room and compute their respective positions in a 3D coordinate system. A motorised chair and an immobilisation device is used to fix the mask and, theoretically, the patient to the chair. The motorised chair is then instructed by the SPG system to move the markers and, by implication, the patient to the position required for treatment.

Fitting the mask to the patient on different occasions opens up the possibility of small differences being introduced in the relative positions of the markers and the patient anatomy. When treating a patient, it is necessary to verify that the patient's anatomy is correctly positioned according to the treatment plan before the patient can be treated. This verification is currently accomplished by visual comparison of a film X-ray image, called a portal radiograph (PR), taken when the patient is positioned and a digitally reconstructed radiograph (DRR), generated by the treatment planning system, in which the patient has the correct treatment position. A DRR is a synthetic image that approximates the physics involved when an X-ray image is generated. The verification procedure described is manual, time-consuming and needs to be repeated for each of the treatment fields [2]. Furthermore, visual inspection is prone to errors. Because proton therapy is used for treatment of lesions close to sensitive organs, these errors will be detrimental to the patient's health.

Image registration is a process whereby the spatial correspondence between two coordinate spaces are established. The result is a transformation linking the two spaces. In this problem we want to establish G_E, the error between the treatment position and the observed position.

Van der Bijl [3] proposed a 2D-3D image registration system that is accurate, robust and automatic. 2D-3D image registration is a process where 3D CT data acquired pre-operatively are registered to a 2D PR image obtained intra-operatively [4]. The PR is compared to various DRRs calculated from the CT data. The comparison is done using a similarity measure and an optimiser searches for the transformation that produces a DRR most similar to the PR. Van der Bijl [3] used Powell's minimiser and the Correlation Coefficient or Mutual Information similarity measures to perform registration using DRRs generated with the ray cast algorithm. A schematic representation of the process is shown in figure 1.

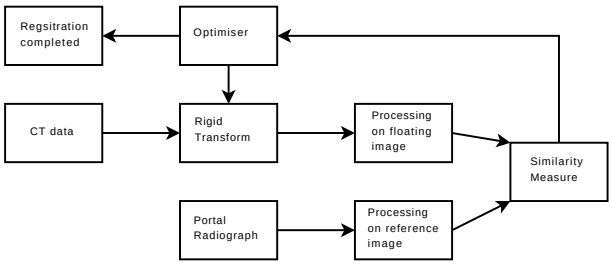


Figure 1: An overview of the 2D-3D registration process. A PR (reference image) is compared to various DRRs (floating images) using a similarity measure. The optimiser searches for the rigid transformation that produces a DRR most similar to the PR.

Van der Bijl [3] reported that his image registration implementation took about 7.5 minutes to verify the patient position. This is too slow for use in practice. The longer a patient needs to wait for the registration process to complete, the higher the probability that the patient will move out of his initial position. The generation of DRRs is computationally expensive and since hundreds of DRRs may be required for the registration process, it is very important to speed up DRR generation.

The aim of this study is to find a fast DRR generation algorithm. In particular we only have to cater for the creation of DRR images that are contained in a known limited vicinity. The image registration process should complete in less than three minutes, which is the minimum time the current manual verification process takes. Furthermore, it must be shown that the new DRR generation algorithm does not destroy the accuracy or robustness of the registration process.

2. Discussion

2.1. DRR generation methods

Carstens and Muller [1] discussed various DRR generation algorithms and concluded that the ray casting and light field algorithms can be trivially parallelised. The ray casting algorithm is used as the gold standard for DRR generation and the light field algorithm can be used to improve the DRR generation time by pre-operative computation of scene dependent data and the intra-operative computation of images using interpolation. Both methods will be discussed briefly.

2.1.1. Ray casting

Let $\rho(i,j,k)$ denote the voxel density or attenuation coefficient in a 3-dimensional CT volume and $l(i,j,k)$ the length of the intersection of an X-ray with that voxel, then the radiological path length is defined as

$$d = \sum_i \sum_j \sum_k l(i,j,k)\rho(i,j,k) \tag{1}$$

The radiological path length is an approximation of the physics involved when an X-ray image is generated. Computing DRRs using the radiological path definition is $O(n^3)$ and very inefficient. Only a few voxels actually contribute to a path, since most $l(i,j,k)$ values will be zero. Siddon [5] proposed viewing CT voxels as the intersection of equally spaced, parallel planes. The intersection of the ray with the planes is then calculated, rather than the intersection of the ray with the different voxels. Determining the intersection of a ray with equally spaced, parallel planes is a simple problem. One needs to calculate the intersection with the first plane and the rest follows at fixed intervals because the planes are equally spaced.

An optimised version of Siddon's algorithm was proposed by Jacobs [6]. The new algorithm reduces computation by eliminating the need to explicitly compute the voxel indices for every interval. Also, it removes the need to allocate memory for the different arrays containing the intersection points. Figure 2 shows how a DRR is constructed.

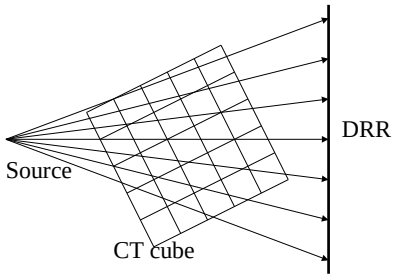


Figure 2: *A 2D view of DRR generation. The DRR is the set of values for the radiological paths from the source to the pixels on the image plane.*

The ray casting method is used as a benchmark of DRR quality, but it is too slow for real-time computations [7] and in its standard form slower than most other algorithms [7][4][8].

2.1.2. Light fields

Light fields is a method that was originally proposed by Levoy and Hanrahan [9]. It can be described as a way of parameterising the set of all rays that emanate from a static scene. Each ray is identified by its intersection with two arbitrary planes in space. It is convention that the coordinate system on the first plane is (u,v) and that this plane is called the *focal plane*. The second plane has a coordinate system (s,t) and is called the *image* plane. It follows that every ray in this space can be represented as a point or pixel value $\mathbf{p}_i = (u_i, v_i, s_i, t_i)$ in 4-dimensional space.

A *light slab* is the shape that is created when the focal plane and the image plane are connected. This represents all the light that enters the restricted focal plane and exits the restricted image plane.

If one can generate infinitely many rays inside a light slab, one can recreate almost any image with a focal point inside the light slab. This is done by finding the associated rays and their corresponding pixel values (figure 3). In practice one cannot generate infinitely many rays and are thus constrained to generate a large number and compute the missing rays using some form of interpolation.

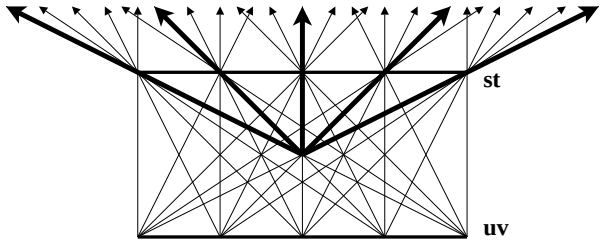


Figure 3: *A 2D view of a light slab, illustrating the view (in bold) generated for an arbitrary focal point.*

Light fields is a simple method to construct novel views from arbitrary camera positions. This is achieved by resampling a set of existing images and is therefore called *image-based rendering* [9]. Image-based rendering algorithms have the advantages that they are suitable for real-time implementations since they are not computationally expensive, the cost of generating a scene is not dependent on the complexity of the scene, and the set of base images can be real images, artificially created ones or both.

The amount of light travelling along any arbitrary ray in space is called its *radiance*. For any arbitrary scene with static illumination the radiance of all rays is called the *plenoptic function*. In the plenoptic function rays are represented by the coordinates x, y, z and the angles θ and ϕ. Each ray has an associated radiance value. When the radiance along a ray does not change the 5D plenoptic function can be reduced to a 4D function. This 4D function is the formal definition of a light field.

The light slab is also called a *plane-plane* representation. This type of parameterisation does not include, for instance, rays that are parallel to the two planes. However, multiple light slabs can be used to represent these. In 3D six light slabs would be required to recreate any arbitrary view of an object.

It is important to note that since the projection space for our DRRs is constrained, only a single light slab is necessary to represent the sampling space. Figure 4 shows how a light slab can be viewed as a 2D array of 2D images where the (u,v) coordinates identify a sub-image in the light slab and the (s,t) coordinates identify a pixel in the sub-image.

What makes light fields attractive for the DRR generation problem is the fact that most computation can be done pre-operatively. During patient treatment, when computation time

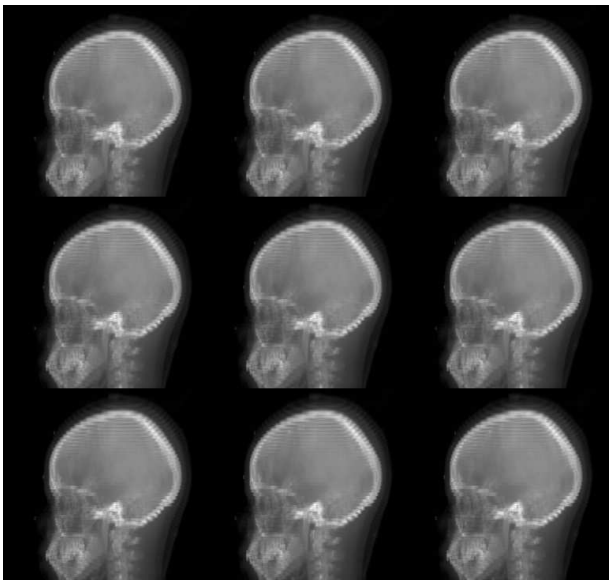


Figure 4: A light slab can be interpreted as a 2D collection of 2D images from different observation points. The (u, v) coordinates identify a sub-image and the (s, t) coordinates a pixel in the sub-image.

must be minimised, it is then possible to quickly generate images from the pre-computed data. The generation of images is achieved by interpolation of the pre-computed data. This can be done in constant time, since the computation time is not dependent on the complexity of the image.

A pixel value in a general light field is an indication of the amount of light reflected off the first surface a ray intersects with. When evaluating DRRs, however, the pixel values are the radiological path lengths (equation 1) the rays encounter from the projection point to the image plane.

To accommodate the generation of DRRs, we can associate each point $\boldsymbol{p}_i = (u_i, v_i, s_i, t_i)$ with a scalar function $\boldsymbol{p}_i \mapsto q(\boldsymbol{p}_i)$ which maps a point to the radiological path length of the ray $R_{\boldsymbol{p}_i}$.

In order to trace a ray through the CT data and maintain the same parameterisation of rays in space as traditional light fields one must cast the rays beyond a *virtual image plane* onto an *effective image plane*. The values on the effective image plane is used for the light field generation.

In traditional light field rendering as well as light field DRR generation, the generated image is a skewed perspective image. However, where in traditional light field rendering the image plane remains fixed and between the scene and the focal plane, in DRR generation the virtual image plane remains fixed while the effective image plane can move and the effective image plane lies on the other side of the scene from the focal plane. Figure 5 illustrates this. Figure 6 shows how an arbitrary DRR can be created from a light slab. For each ray from an arbitrary projection point to an arbitrary projection plane, the intersections with the focal and virtual image planes are computed. These intersections are then used to calculate the indices into the light slab as well as the weights used to perform the interpolation.

The implementation of the light field DRR generation algorithm is parallelised using OpenMP [10] as follows:

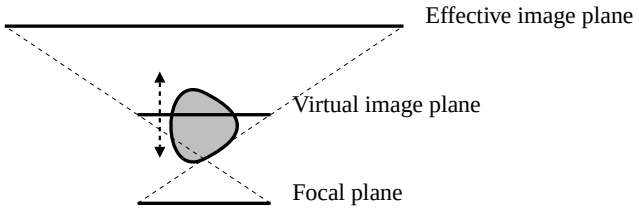


Figure 5: The positions of the focal-, effective image- and virtual image planes used when constructing DRRs using light fields. The grey object is the CT data positioned relative to the planes. The virtual image plane can be positioned anywhere between the focal- and effective image planes.

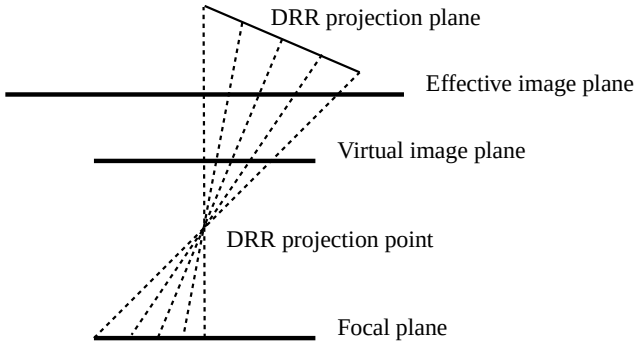


Figure 6: An example of constructing a DRR from a light slab. The dashed lines show the intersections of rays with the focal and virtual image planes. The effective image plane is shown for illustrative purposes only.

```
#ifdef _OPENMP
#pragma omp parallel for
#endif
for (int j=0; j<yResolution; j++)
#ifdef _OPENMP
#pragma omp parallel for
#endif
  for (int i=0; i<xResolution; i++)
    image(i,j)=getValue(focalPoint,
                        targetPoint(i,j),
                        lightSlabData);
```

The algorithm iterates over the pixel indices of the DRR image and for each pixel computes the value using a specified focal point and the target point associated with the pixel index as input to the `getValue` function. The `getValue` function computes the intersections of the ray passing through the focal and target points with the focal and virtual image planes. These intersections are then used to compute the interpolated value using quadrilinear interpolation. Because of the parallelisation multiple pixel values are computed simultaneously.

3. Evaluation

3.1. Quantifying the error

The error transformation is defined as

$$G_E = (R_z(\theta_z)R_y(\theta_y)R_x(\theta_x), (\delta_x, \delta_y, \delta_z)^T) \quad . \qquad (2)$$

The components of equation 2 can also be represented in vector form:

$$\boldsymbol{v} = (\delta_x, \delta_y, \delta_z, \theta_x, \theta_y, \theta_z) \quad , \tag{3}$$

where $\boldsymbol{v}$ is referred to as a position vector.

Let the *exact* (correct) solution to an optimisation process be called $\boldsymbol{s}_e$ and the *calculated* solution $\boldsymbol{s}_c$, where both vectors are position vectors. The difference between the exact solution and the calculated solution is

$$\boldsymbol{s}_d = \boldsymbol{s}_e - \boldsymbol{s}_c \quad .$$

The first three elements of $\boldsymbol{s}_d$ have *millimetre* units and the last three elements have *degree* units. Converting the differences of the individual elements to percentages removes this difference. Since the size of the allowed error range for a dimension is $2\epsilon_\theta$ degrees or $2\epsilon_\delta$mm, we use these ranges when calculating the percentage error.

The Euclidean distance of the individual errors are calculated and scaled by a factor $\frac{1}{\sqrt{6}}$ to produce a dimensionless value quantifying the total error. In practice we also evaluate the errors on the individual elements when looking at the accuracy of the registration process, but for brevity the individual results are not included in this article.

3.2. Evaluation of DRR generation methods

The similarity of DRRs can be evaluated qualitatively or quantitatively, although only the latter is feasible for use in algorithms. *Difference images* are a means of qualitative comparison. The absolute difference image d of two $M \times N$ images f and g is given by

$$d(m,n) = \mid f(m,n) - g(m,n) \mid \quad ,$$

where m and n are indices in the images. The quantitative measures used in this study was the Correlation Coefficient and the Mutual Information similarity measures, as suggested by Van der Bijl [3].

The correlation coefficient of two images f and g is defined as

$$CC(f,g) = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} ab}{\sqrt{\sum_{n=1}^{N}\sum_{m=1}^{M} a^2 \sum_{n=1}^{N}\sum_{m=1}^{M} b^2}} \quad ,$$

where $a = f(m,n) - \bar{f}$, $b = g(m,n) - \bar{g}$, $f(m,n)$ and $g(m,n)$ denote the pixel values at position (m,n) in image f and g, respectively, and $\bar{f}$ and $\bar{g}$ denote the mean pixel value in image f and g, respectively.

The Mutual Information shared between two images f and g is defined as

$$MI(f,g) = H(f) + H(g) - H(f,g) \quad ,$$

where $H(f)$ and $H(g)$ are Shannon's entropies of images f and g, respectively, and $H(f,g)$ is the *joint* entropy of the two images.

3.3. The optimiser

Powell's method is an unbounded minimisation algorithm to determine local minima for multidimensional functions and forms part of a class of methods called *direction set methods*. It accomplishes the minimisation by repeatedly performing line minimisations. What makes Powell's method very attractive for the purposes of this study is the fact that it does not involve computation of the cost function's gradient [11]. Firstly, no analytic gradient exists for our cost function and secondly, approximating the gradient numerically using finite difference or forward difference methods would be computationally expensive as it requires the generation of more DRRs.

3.4. The cost function

The cost function is defined as a minimising function. This function needs to be defined for all $\boldsymbol{x} \in \Re^6$, since Powell's minimiser is an unconstrained optimiser. Using ray casting we can easily generate arbitrary DRRs, but using the light field algorithm we are constrained to the sampled space contained in the light slab. To overcome this limitation we define an arbitrary function with the criteria that it provides values for points outside the sampled space and guides the optimiser to the minimum.

Let $g(\boldsymbol{x})$ be a similarity measure defined in the interval $-\epsilon \leq x_i \leq \epsilon$ for all six dimensions. It compares two DRRs and returns lower values for higher similarity and higher values for lower similarities.

Let $c(\boldsymbol{x})$ be a cost function, where

$$c(\boldsymbol{x}) = \begin{cases} -x_i & \text{if } x_i < -\epsilon \\ g(\boldsymbol{x}) & \text{if } -\epsilon \leq x_i \leq \epsilon \\ x_i & \text{if } \epsilon < x_i \end{cases}$$

The two functions investigated in this study as possibilities for $g(\boldsymbol{x})$ are the Mutual Information and Correlation Coefficient similarity measures. Since the Mutual Information and Cross Correlation similarity measures both always return positive results, it must be negated when used in the cost function. So, for Mutual Information,

$$g(\boldsymbol{x}) = -MI\left(DRR(\boldsymbol{x}), DRR_{ref}\right) \quad ,$$

where $DRR(\boldsymbol{x})$ is a function returning the DRR when a transformation $\boldsymbol{x}$ is used and DRR_{ref} is a reference image. The same applies when using the Correlation Coefficient similarity measure.

4. Results

4.1. Similarity performance

A similarity experiment was used to evaluate the effect of using DRRs from various light slab configurations on the cost function of the optimiser. The effect was compared to the similarity performance of ray casted DRRs.

The similarity measures peaked where the DRRs were most similar and gradually worsened in a decreasing fashion when DRRs were positioned further away from the reference DRR.

For the light field DRRs to be useful, we expect that the similarity curves do not contain local minima or maxima (extrema) and the location of the extremum must coincide with the extremum of the ray casted DRRs.

The similarity measurements were taken for movements along the six error dimensions as well as four other arbitrary complex movements. The different error dimensions are all limited to $\pm 5mm$ or $\pm 5^\circ$. On the graphs all movements are parameterised to the interval $[0 \ldots 1]$ and are expected to peak at $\frac{1}{2}$. Figure 7 shows an example of one such test. In most of the tests both the Correlation Coefficient as well as the Mutual Information similarity measures performed quite well in terms of the criteria set out initially. The curves does not contain local extrema and the locations of the extrema coincide with the extremum of the ray casted DRRs.

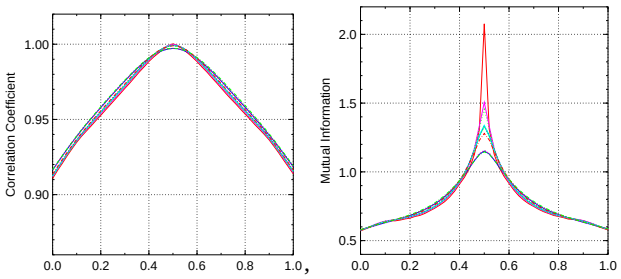


Figure 7: Similarity measures for translations along the y and z axes using DRRs generated from various light slab resolutions.

It is worth noting that the Correlation Coefficient similarity measure were sometimes close to a straight line, having a range of $[0.995 \ldots 1.00025]$. Figure 8 shows an example. This makes it susceptible to numerical errors which produces local extrema. This behaviour was most prevalent for translations on the x axis, which has a zoom or shrink affect on the resulting DRR.

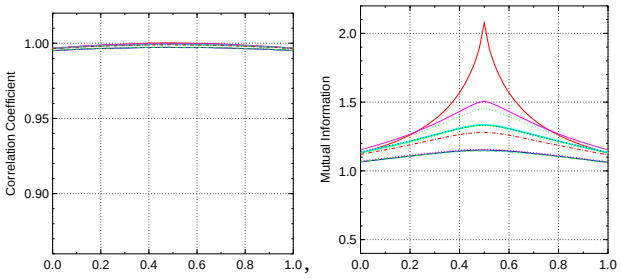


Figure 8: Similarity measures for translations along the x axis using DRRs generated from various light slab resolutions.

4.2. DRR computation time performance

The tests were performed on a machine with the following specifications:

CPUs Two quad-core Intel Xeon CPUs running at 2GHz. This effectively translates to *eight* processors.

RAM 2GB

Operating System Scientific Linux SL release 5.1 (Boron) (based on Red Hat)

Compiler gcc version 4.1.2 20070626 (Red Hat 4.1.2-14)

Table 1 shows the average ray casting performance times when using sequential compilation and when using parallelisation on the ray cast and light field algorithms.

	RC	LF	PRC	PLF
avg	2.781	0.439	0.450	0.056
σ	0.023	0.001	0.004	0.001

Table 1: Generation time (s) of a 512×512 DRR using the Ray Cast, Light Field, Parallelised Ray Cast and Parallelised Light Field algorithms.

4.3. Light slab computation time

Although the light slab computation time is performed pre-operatively and therefore does not add to the patient treatment time, it is informative from a practical point of view to show that these computation times are not excessive. A parallelised light slab generation algorithm was used. The average computation time of the light slabs are given in table 2.

Focal Res	Image Res	Min (s)	Max (s)	Avg (s)
32×32	128×128	10.352	38.106	21.494
32×32	256×256	41.063	108.216	72.302
32×32	512×512	160.915	338.579	260.800
64×64	128×128	43.112	156.291	87.766
64×64	256×256	163.989	418.143	285.329

Table 2: The computation times of light slabs with varying focal and image plane resolutions.

The differences between the minimum and maximum light slab computation times can be attributed to the fact that the ray cast algorithm, which is used to generate the light slab, has a computation time which is dependent on the complexity of the scene. For certain configurations, most notably a DRR with a view diagonally through the CT cube, the number of voxel traversals is significantly higher than other configurations.

The maximum time measured in this experiment is approximately 7 minutes for the light slab with a virtual image plane resolution of 256×256 and a focal plane resolution of 64×64. Being a pre-operative computation, it is completely acceptable.

4.4. Image registration performance

In this experiment we evaluated the effect of using light field DRRs in an image registration algorithm compared to ray casted DRRs. Each algorithm was tested in conjunction with the two similarity measures and as in the similarity tests, a comparison was made between solutions found using ray casted DRRs and those found using the light field algorithm.

The tolerance of the line search method used by Powell's algorithm was 0.1. No tolerance was set on the value of the cost function. The algorithm terminates when the absolute difference in all the individual dimensions of the solution are less than 0.1mm or 0.1° from the previous solution.

The results are presented using the following definitions:

S The similarity measure used. The two options are:

- **M** Mutual Information
- **C** Correlation Coefficient

A The DRR generation algorithm used. The two options are:

- **R** Ray Cast
- **L** Light Field

ξ The total error.

Time The time (in seconds) required by the image registration process.

It is important to note that, since the sizes of the search spaces are 10mm (-5mm $\ldots$ 5mm) and 10° ($-5^\circ \ldots 5^\circ$), a one percent error in one of the dimensions translates to a real error of 0.1mm and 0.1°.

The SPG system aims to achieve sub-millimetre accuracy. This is the aim of the image registration system as well, which

means that errors larger than 10% are considered unacceptably high.

The tests were performed using various light slab dimensions. Also, solutions were varied to lie within the sampled space and at the discontinuities in the cost function, which are boundary cases. The results corresponded and an arbitrary sample is provided in table 3.

S	A	ξ	Time (s)
M	L	0.35	29.19
M	R	0.36	86.63
C	L	1.83	14.51
C	R	1.17	63.43

Table 3: Image registration performance for perturbations in all dimensions.

4.4.1. Performance in terms of accuracy

The performance of the Mutual Information similarity measure in terms of accuracy was very good. Most of the errors were less than 1% and all errors were less than 2%.

The Correlation Coefficient similarity measure performed badly when determining the x translation parameter. This is not unexpected. It was seen in the similarity tests that the Correlation Coefficient similarity measure performed poorly and this reflects directly in the optimiser accuracy performance. All errors were less than 6% and most were less than 2%. Disregarding the x translation parameter, all the errors were smaller than 2% and in most cases the error was less than 1%, similar to the Mutual Information similarity measure.

All tests using the light field algorithm performed well compared to the tests using the ray cast algorithm, with accuracy that was mostly in the same order of magnitude and sometimes even better.

4.4.2. Performance in terms of computation time

In all the experiments the optimiser using the Correlation Coefficient similarity measure combined with DRRs generated from light slab performed the fastest, with registration times faster than 21 seconds. The second fastest setup in all experiments are the Mutual Information similarity measure combined with DRRs generated from light slabs, with registration completing in less than 40 seconds. This is almost double the worst case time when using the Correlation Coefficient similarity measure, but still a big improvement over the time taken for manual verification. It is important to note that in all experiments the optimisation processes using the light field algorithm always outperformed those using ray casting.

5. Conclusions and future work

This study set out to find a DRR generation algorithm that is fast and that, when used in conjunction with an image registration algorithm, produces a fast, accurate and robust method for verifying the patient position. The goal was to be able to accurately determine the error in the patient position in under three minutes. A parallelised implementation of the light field algorithm was shown to satisfy all the requirements.

Accurate registration is performed in under a minute and the algorithm will automatically perform even better if more CPUs are added to the machine on which it is executed. This greatly improves the 7.5 minutes reported by Van der Bijl [3]. The parallelised light field algorithm performs roughly 50 times faster than than the serial ray cast algorithm as proposed by Jacobs [6]. This improvement did not directly translate to the optimiser time, as the implementation in this study required substantially more cost function evaluations than was the case in [3].

An arbitrary definition of an unconstrained cost function was used in the image registration implementation. This cost function was shown to work with both the Mutual Information and the Correlation Coefficient similarity measures. Although the Mutual Information similarity measure took longer to complete than the Correlation Coefficient similarity measure in some cases, it produced lower errors overall. The definition of the unconstrained cost function and possibly the parameters used by the optimisation algorithm would be worthwhile areas for further investigations.

6. References

[1] Cobus Carstens and Neil Muller, "Fast calculation of digitally reconstructed radiographs using parallelism," in *Proceedings of the Eighteenth Annual Symposium of the Pattern Recognition Association of South Africa*, J.R. Tapamo and Fred Nichols, Eds., Durban, South Africa, 2007, pp. 57–62, Pattern Recognition Association of South Africa.

[2] Evan A. de Kock, "Concepts and definitions required for the development of a portal radiographic verification system at iThemba LABS," Tech. Rep., June 2004.

[3] Leendert van der Bijl, "Verification of patient position for proton therapy using portal X-rays and digitally reconstructed radiographs," M.S. thesis, University of Stellenbosch, 2006.

[4] D. B. Russakoff, T. Rohlfing, D. Rueckert, R. Shahidi, D. Kim, and C. R. Maurer, Jr., "Fast calculation of digitally reconstructed radiographs using light fields," in *Medical Imaging 2003: Image Processing. Edited by Sonka, Milan; Fitzpatrick, J. Michael. Proceedings of the SPIE.*, May 2003, vol. 5032, pp. 684–695.

[5] Robert L. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Medical Physics*, vol. 12, no. 2, pp. 252–255, March 1985.

[6] F. Jacobs, E. Sundermann, B. De Sutter, M. Christiaens, and I. Lemahieu, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *Journal of computing and information technology*, vol. 6, no. 1, pp. 89–94, 3 1998.

[7] C.T. Metz, "Digitally reconstructed radiographs," M.S. thesis, Utrecht University, 2005.

[8] Philippe Lacroute and Marc Levoy, "Fast volume rendering using a shear-warp factorization of the viewing transformation," *Computer Graphics*, vol. 28, no. Annual Conference Series, pp. 451–458, 1994.

[9] Marc Levoy and Pat Hanrahan, "Light field rendering," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1996, pp. 31–42, ACM Press.

[10] OpenMP Architecture Review Board, "OpenMP Application Program Interface 2.5," 2005.

[11] D. M. Greig, *Optimisation*, Longman Inc., New York, NY, USA, 1980.

Traffic Sign Detection and Classification using Colour and Shape Cues

F.P. Senekal

Council for Scientific and Industrial Research, South Africa

fsenekal@csir.co.za

Abstract

This paper presents a new technique for the recognition of road traffic signs. The technique is based on colour and shape analysis of a single image. It is aimed at the detection and classification of triangular traffic signs, such as warning and yield signs. The technique is applied to a set of images obtained from a camera mounted on a moving vehicle. Good detection and classification performance is achieved.

1. Introduction

The ability to recognise road and traffic signs is becoming an important research area in *Intelligent Transport Systems* (ITS) and has a number of applications. In *driver support systems*, such a system could focus a driver's attention to road conditions ahead, such as pedestrians that may be crossing the road or a change in the allowed speed limit, allowing the driver to take appropriate action on time. In *intelligent autonomous vehicles*, the ability to recognise and interpret such signs could contribute greatly to their control and safe navigation. For example, a sign indicating that there is a stop ahead may lead the control system to reduce the speed of the vehicle. In *highway maintenance* and *sign inventory* applications, the ability to recognise and possibly to evaluate the condition of the signs, can greatly reduce the effort in maintaining current road infrastructure.

Traffic signs are designed to have specific saturated colours that are easily distinguishable from their environment. In South Africa and many other countries, typical control, prohibition and warning signs contain red, black and/or white; typical command and reservation signs contain blue and/or white; and typical route markers and tourism signs contain green, blue or brown with white and/or yellow lettering. They also have specific shapes; command and prohibition signs are circular, warning signs and yield signs are triangular, reservation, route markers and tourism signs are rectangular and stop signs are octagonal. They are placed near the road surface in a clearly visible position, usually free from any occlusions. Figure 1 shows examples of commonly occuring traffic signs.

Figure 1: *Typical traffic signs, showing their unique colour and shape (from left to right: stop, yield, pedestrians only, 100km/h speed limit, no u-turn, pedestrian crossing ahead) (note that images are available in colour).*

The fact that traffic signs have unique colours and shapes are often exploited in algorithms designed to recognise them. These algorithms typically follow a two step process. In the *detection* phase, the position and shape of the signs (if any) in the image are determined. In the *classification* phase, the aim is to assign class labels to the signs that were detected. Detection and classification usually constitute recognition in the scientific literature. A third *applicability* phase may be required to determine whether a given sign in the visual field is applicable in the current situation or, put differently, to recognise whether a particular sign is relevant in the current context of the application. This is particularly important in applications such as driver support systems and intelligent autonomous vehicles. Although robust detection and classification algorithms have been developed, determining the applicability of a sign is a difficult task that has not been adequately addressed in the literature and presents an opportunity for future research.

Detection is usually performed on colour images, although some studies have also been executed on grayscale images. When colour images are used, segmentation through colour thresholding, region detection and shape analysis are usually performed. The choice of colour space is important during the detection phase. When the RGB colour space is used [1, 2, 3], thresholding is usually based on relations between the colour components. Others work in the HSI or HSV colour space [4, 5, 6], where the relations between the components is somewhat simplified. Other colour spaces, such as LUV [8] and CIECAM [9] have also been used. Due to the varying colour conditions that may occur, more extensive approaches have also been developed. Databases for colour pixel classification are used in [10] and [11]. Fuzzy classification [12] and neural networks [13] have also been tried. Border detection on grayscale images [14] is another approach that have been taken.

Classification can be accomplished by a number of approaches. Template matching is used in [15] and [16]. Multilayer perceptrons [1, 17], radial basis function networks [18], Laplace kernel classifiers [19] and genetic algorithms [4] have also been studied.

The recognition of traffic signs presents a number of difficulties, both in terms of the *image formation process* and in terms of the *environment* in which the sign is found. In the image formation process, the size of the sign in the image depends on its physical size and its distance from the camera and in general could be arbitrarily rotated. There will be an aspect modification in the projection of the sign in the image if the optical axis of the camera is not perpendicular to the sign (i.e. perspective distortion). There is also no standard colour associated with the signs, as the colour will depend on various photometric effects. In addition, effects such as sensor noise and motion blur may be present in the image. Difficulties in the environment in which the sign is found can be divided into four groups, illustrated in Figure 2. The *physical condition* of the sign may make recognition difficult, such as the effect of deteriorating paint quality over time (Fig. 2a), signs that are damaged (Fig. 2b), signs that are incorrectly placed, the presence of graffiti

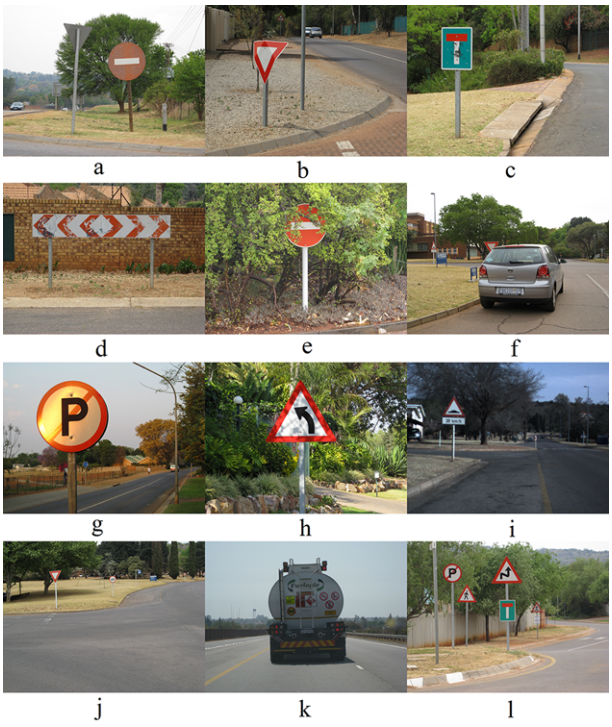


Figure 2: *Difficulties in the recognition of traffic signs: (a) deteriorating paint quality, (b) damaged sign, (c) graffiti, (d) general deterioration, (e) partial occlusion by a static object, (g) partial occlusion by a dynamic object, (g) reflections, (h) shadows, (i) low-light conditions, (j) sign not applicable, (k) sign not applicable, (l) too many signs.*

(Fig. 2c) or just general deterioration of the sign (Fig. 2d). *Partial occlusions* of the sign, both of a static (Fig. 2e) and dynamic (Fig. 2f) nature, and *lighting conditions* such as reflections (Fig. 2g), shadows (Fig. 2h) and low-light conditions (Fig. 2i) may also have a severe influence. Finally, there may be difficulty in determining the *applicability* of a traffic sign. In Fig. 2j the yield sign is only applicable to drivers using the side road, in Fig. 2k the speed sign is applicable only to the vehicle with which it is associated and in Fig. 2l there may be general confusion due to the many signs present.

2. Method

In the work presented here, the interest is in recognising *triangular signs* such as warning and yield signs. These signs have a red triangular frame that usually surrounds a black iconic representation of an object on a white background. The algorithm discussed here can be applied to a single image, i.e. it is not dependent on temporal consistencies between successive frames in a video sequence. It is assumed that the traffic sign is not occluded by objects in the environment in such a way as to segment its projection onto the image plane into different regions or in such a way that the visible portion of the interior of the sign is fundamentally altered. A further assumption is that the sign is fully contained in the interior of the image, i.e. it does not protrude beyond the boundaries of the image.

An overview of the steps in the algorithm is shown in Figure 3. An example of the output of some of these steps is shown in Figure 4, using the source image shown in Fig. 4a. The steps

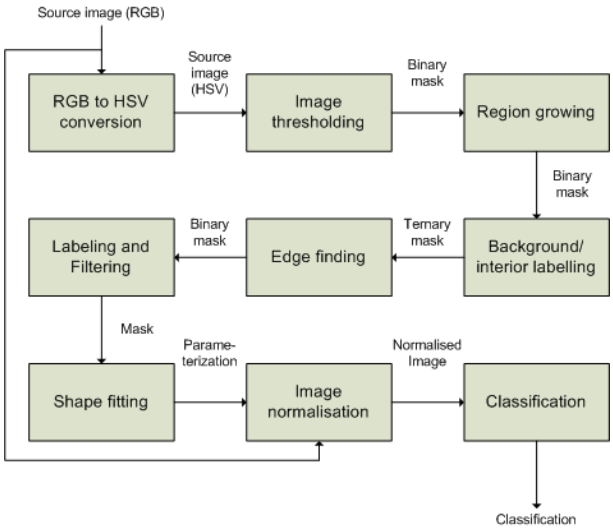


Figure 3: *Steps in the recognition of traffic signs.*

are discussed in detail in the subsections that follow; here a brief overview is given.

The image is converted from the RGB to the HSV colour space, after which an initial threshold is applied to determine red regions (possible traffic signs) (Fig. 4b - background shown as white). These red regions are "grown" to include other possible regions that may qualify but which may not have qualified during the initial thresholding step. Next, the interiors and exteriors (background) of possible signs are marked (Fig. 4c - background in white, interiors in yellow). This process is likely to fail in the presence of occlusions, where the interior and exterior regions are connected and thus not be easily separable. Edges are then extracted from the image where the interiors touch the possible signs (Fig. 4d). A component labelling algorithm is then applied to determine different edge segments that are 8-connected. Small edge segments that are likely to be noise is discarded (Fig. 4e). Separate edge segments are tested to determine whether they provide a good fit for a triangle (Fig. 4f). If such a fit is established, the three vertices of the triangle are noted. Using these vertices, interpolation based on barycentric coordinates is applied to map the triangle in the original image onto a new normalised triangle with fixed scale and rotation (Fig. 4g). This normalisation also aims to reduce the effect of perspective distortion. Classification is achieved by matching this normalised triangle to a set of reference templates.

2.1. Colour Conversion

Most digital image formats store a digital image as a series of two-dimensional arrays, specifying the red, green and blue (RGB) channels. The first step is to convert each pixel of the source image to its equivalent in the hue, saturation and value (HSV) colour space. The HSV colour space provides a convenient interpretation of the meaning of colour. The reader is referred to [22, p.623] for a description of the conversion process.

2.2. Image Thresholding

The fact that triangular signs have a characteristic red frame can be exploited to identify regions in the image that could possibly contain such signs. The image is thresholded to identify regions

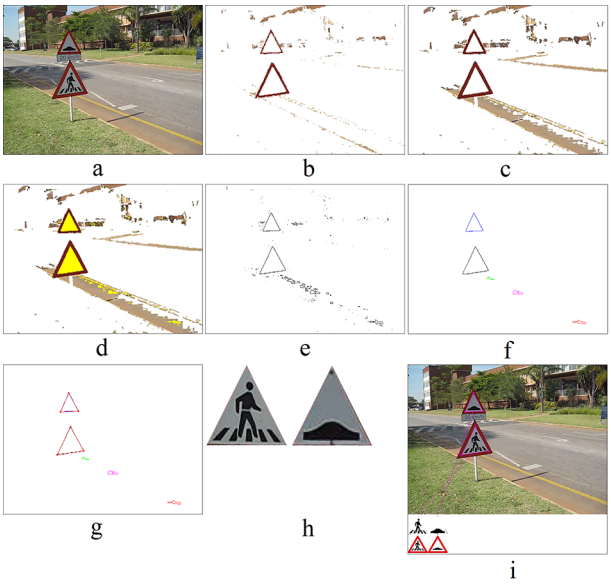


Figure 4: *Output of the various steps in the recognition process: (a) source image, (b) after thresholding, (c) after background/interior labelling, (d) after edge finding, (e) after labelling and filtering, (f) after shape fitting, (g) after normalised images have been extracted, (h) after classification.*

with red pixels. The output of this process is a mask that specifies for every pixel whether it is adequately red or not.

The hue and saturation components are sufficient in identifying red regions in warning signs. The mask is defined as

$$\mu_1(x,y) = \begin{cases} 1, & \text{if } S(x,y) \geq T_S \text{ and} \\ & (H(x,y) \leq T_H \text{ or } H(x,y) \geq 1 - T_H), \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where T_S is a threshold related to the saturation of the pixel, and T_H is a threshold related to the hue of the pixel and S and H are the saturation and hue respectively at coordinate (x, y).

Since hue values close to 0 and 1 are indicative or red, there are two conditions related to the hue value. The output of this step is shown in Figure 4b (object pixels are shown in their original colour, background pixels are white).

2.3. Region Growing

For a variety of reasons, such as a sign's paint that fade over time or the presence of reflections and shadows, the frame of the sign may contain regions that are not a highly saturated red colour. Such regions may not be detected under the mask defined by (1).

Under the assumption that such regions will be close to the regions identified by (1), and that they will be "somewhat" red, the mask can be grown to include such regions. The new red mask μ_2 is expressed by

$$\mu_2(x,y) = \begin{cases} 1, & \text{if } \mu_1(x,y) = 1 \text{ or} \\ & (S(x,y) \geq t_S \text{ and} \\ & (H(x,y) \leq t_H \text{ or } H(x,y) \geq 1 - t_H) \text{ and} \\ & \exists (x_0,y_0) \in N(x,y) \text{ s.t. } \mu_1(x_0,y_0) = 1), \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where t_S and t_H are new threshold values and $N(x, y)$ is a neighbourhood of (x, y). Note that proximity to a pixel that is already classified as red is required, but the thresholding conditions are relaxed such that $t_S \leq T_S$ and $t_H \geq T_H$. The procedure can be applied iteratively, replacing the previous mask by the new mask, and can be stopped after convergence.

2.4. Background/Interior Labelling

At this stage, it is undecided whether a given non-object pixel (corresponding to a 0 in the mask) is internal or external (background) to the object.

As was previously mentioned, the assumption is made that the sign is fully contained in the interior of the image (we are not making predictions about signs that protrude across the boundaries of the image). Under this assumption, all internal pixels are completely surrounded by at least a single line of object pixels. No internal pixels are thus found in the boundary (the first and last rows and columns) of the image.

We can exploit this assumption by noting where mask boundary that have a value of 0 and correspondingly marking them as background. Using these coordinates as seed values, we recursively find 4-connected pixel neighbours, and each such neighbour which also has a mask value of 0 is then also marked as background. After the recursion process, all coordinates with a mask that has a value of 0 and which has not been marked as background are interior pixels. It is possible for remaining values not to be "true" interiors but rather to exist due to the boundaries of the objects surrounding them touching each other. However, such regions will be completely surrounded by the object boundaries and thus cannot be distinguished from true interiors.

Mathematically, this can be expressed as

$$\mu_3(x,y) = \begin{cases} 1, & \text{if } \mu_2(x,y) = 1, \\ -1, & \text{if } \exists (x_0,y_0) \in B \text{ s.t. there exists} \\ & \text{a 4-connected path } P \text{ between } (x,y) \text{ and} \\ & (x_0,y_0) \text{ s.t. } \mu_2(x_i,y_i) = 0 \forall (x_i,y_i) \in P, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where B is the set of pixel coordinates defining the boundary of the image, P is a set of pixel coordinates defining a 4-connected path between $(x0, y0)$ and (x, y)..

The output of this step is shown in Figure 4c, where yellow is used to indicates an internal pixel and white a background pixel. Object pixels are shown in their original colours.

2.5. Edge Finding

The next step in the algorithm is to find the pixels corresponding to the object-interior edges. An edge in this context is defined as any interior pixel that is 4-connected to an object pixel, and is given by

$$\mu_4(x,y) = \begin{cases} 1, & \text{if } \mu_3(x,y) = 0 \text{ and} \\ & (\mu_3(x+1,y) = 1 \text{ or } \mu_3(x-1,y) = 1 \text{ or} \\ & \mu_3(x,y+1) = 1 \text{ or } \mu_3(x,y-1) = 1), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The output of this step is shown in Figure 4d, where a black indicates an edge and a white a non-edge.

2.6. Labelling and Filtering

A connected component labelling algorithm is applied to the mask to determine which edges are 8-connected. A two-pass algorithm is applied. In the first pass, an initial labelling of

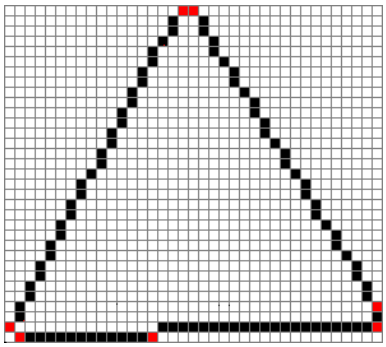


Figure 5: *Connection points (indicated in red) used for line fitting.*

the edges in single scan lines is performed. Labelling conflicts between successive scan lines are noted. On completion of the first pass, the union find algorithm [20, pp. 441-440] is applied to resolve labelling conflicts. A second pass is performed to re-label each of the original edge labels.

The labelling algorithm produces a new mask μ_5. A positive value of $i = \mu_5(x, y)$ indicates that the coordinate (x, y) is associated with the *i*'th edge object.

The area of each edge object is calculated as

$$A_i = \sum_{\mu_5} a_i, \text{ where } a_i = \begin{cases} 1, & \text{if } \mu_5(x,y) = i, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where the sum is taken over all image coordinates (x, y). Edge objects with a large enough areas are retained, that is edge objects for which $A_i \geq T_A$, where T_A is a threshold specifying the minimum area. The step is likely to filter out edge objects that are present due to noise. In addition, edge objects that correspond to small areas for which a classification would in any case not be possible are filtered out.

The output of this step is shown in Figure 4e, where different colours are used to represent the different edge objects that are retained.

2.7. Shape Fitting

The previous step will retain all edge objects that have a large enough area to merit further consideration. In this step, the objective is to determine whether these edge objects provide a good fit to a triangle. The approach taken is to fit various lines through edge pixels. An algorithm such as RANSAC could be applied for this purpose, but a deterministic approach is sought for robust detection.

This is achieved by means of "connection points" (illustrated in Figure 5). Given the bounding box of the edge object, the connection points are defined as the most top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom and left-top pixel coordinates in the bounding box that form part of the edge object. Connection points with the same coordinates are noted as a single point. There are thus a maximum of 8 unique connection points. Let N be the number of such unique points.

A line segment is fitted through successive pairs of successive connection points (modulo N), using their coordinates as beginning and end points for the line segment. For each such line segment, the closest distance d from each edge object coordinate to the line segment is calculated. All edge object coordinates within a distance $d \leq D$ are noted. Let the number of such points be S_i (i varies from 1 to N). This represents a score associated with the line segment i. For each line segment, a linear least squares approximation is performed to determine the equation of a line that fits through the S_i points.

The N lines are now sorted according to the score S_i associated with each line, in descending order. Some of these lines may be associated with the same side of a triangle and thus need to be filtered out. To achieve this, a new list of lines is created. Working in descending order of score, a line is added to the new list if it has a non-overlapping angle with any of the lines already in the list. Two lines are overlapping if their angular difference is less than a threshold T_θ. The three top scoring, non-overlapping lines are used for triangle estimation. If there are less than three such lines, the detection process is stopped.

The sum of the scores associated with these three lines is noted (filtering out coordinates that contribute more than once in each of the individual scores). Let the sum of these scores be S. For a good fit, it is required that $S \geq k_S A_i$, for some $0 \leq k_S \leq 1$. If such a good fit exists, the intersection points of the three lines are determined. Let these points be $P_i = (x_i, y_i), i = 1, 2, 3$. A final test is performed to determine whether these intersection points are within a certain distance from the bounding box of the edge segment and within the bounds of the image. If this is the case, it is assumed that a triangle is successfully detected.

A distinction is made between the "yield" (pointing to the bottom) and "warning" (pointing to the top) configuration of the triangle. Let y_{min} represent the minimum of the three triangle y-coordinates and y_{max} the maximum. The yield configuration is assumed if two of the y-coordinates of the triangle are less than $\frac{ymin+ymax}{2}$ and the warning configuration is assumed otherwise. The coordinates P_i defining the triangle are reordered. In the case of the yield configuration the order is top-left, top-right, bottom-centre and in the case of the warning configuration the order is top-centre, bottom-left, bottom-right.

The output of this step is shown in Figure 4f, where the detected triangle sides are indicated in red.

2.8. Image Normalisation

A normalised image with dimensions $L \times L$ pixels is now created. A useful choice, if multiresolution techniques is to be applied, is to let L be of the form 2^n. In the case of the yield configuration, the coordinates defining the normalised triangle is given by p_1 = (0, 0), p_2 = (0, L-1) and p_3 = ($\frac{L-1}{2}$, L-1) and in the case of the warning configuration, these coordinates are p_1 = (0, $\frac{L-1}{2}$), p_2 = (L-1, 0) and p_3 = (L-1, L-1). A mapping is required that will map the triangle defined by the coordinates P_i in the original image to a triangle defined by the coordinates p_i in the normalised image.

To achieve this, barycentric coordinates are used. A point $p = (x, y)$ within the bounds of the triangle defined by the p_i coordinates is expressed as $p = w_1 p_1 + w_2 p_2 + w_3 p_3$, where w_i are weights such that $w_1 + w_2 + w_3 = 1$. $[w_1, w_2, w_3]$ are the barycentric coordinates. For a warning configuration, the coordinates are given by

$$w_1 = \frac{-1}{L-1} y + 1 \quad (6)$$

$$w_2 = \frac{-1}{L-1} x + \frac{1}{2(L-1)} y + \frac{1}{2} \quad (7)$$

$$w_3 = 1 - w_1 - w_2, \quad (8)$$

and for the yield configuration, the coordinates are given by

$$w_3 = \frac{1}{L-1}y \tag{9}$$

$$w_2 = \frac{1}{L-1}x - \frac{1}{2(L-1)}y \tag{10}$$

$$w_1 = 1 - w_2 - w_3. \tag{11}$$

The barycentric coordinates are calculated for each pixel in the normalised image that lies within the triangle. A corresponding point P in the original image is then calculated as $P = w_1P_1 + w_2P_2 + w_3P_3$. Using this coordinate, bilinear interpolation is applied to determine a red, green and blue value for the pixel in the normalised image.

2.9. Classification

A grayscale version of the normalised image is calculated. The classification approaches taken for warning and yield signs are slightly different. For warning signs a binary image is created from the grayscale image through thresholding. Due to the possible variety of lighting conditions, a single threshold value will not be sufficient in all cases. To address this, a dynamic thresholding algorithm as described in [22, pp. 599-600] is implemented. The histogram of the grayscale values is calculated. The objective is to find a threshold value that will clearly distinguish between dark and light regions in the image, which is akin to finding a "good" separation between the two peaks in the histogram. The median grayscale value is chosen as the initial threshold. Two means are calculated: the mean of pixels darker than the threshold and the mean of pixels lighter than the threshold. The average of the two means is taken as the next threshold value. Threshold values are iteratively calculated until convergence is achieved.

Let $b(x, y)$ represent the resulting binary image with dimensions $L \times L$. The geometric mean of the binary image is calculated as

$$(m_x, m_y) = \frac{1}{N_b}(\sum_{x=0}^{L-1}\sum_{y=0}^{L-1}(1-b(x,y))x, \sum_{x=0}^{L-1}\sum_{y=0}^{L-1}(1-b(x,y))y), \tag{12}$$

where the values 0 and 1 in the binary image represent black and white respectively and N_b is the number of black pixels.

The binary image is compared to a set of reference templates. This is achieved by aligning the binary image with each reference template by their mean coordinates and calculating the number of pixel differences δ_i in the intersection of the binary image with the i^{th} reference image. Let δ_{min} be the minimum over all δ_i and I the index associated with the minimum. The sign is classified as belonging to class I if $\delta_{min} \leq T_\delta$, where T_δ is a threshold specifying the maximum allowed difference between the image and the template. If the minimum distance is larger than the threshold, no classification is made.

For yield signs the approach taken is different. Since the proper yield sign consists only of light pixels, a dynamic thresholding technique would fail, thus necessitating a different technique. The Euclidean distance between the grayscale image and the template image is calculated and the class associated with the minimum distance is assigned. Since there are only two types of yield signs, this approach works well.

3. Results and Discussion

To create the template images, reference sheets of the official traffic sign designs were obtained from the Department of Transport in South Africa [21]. From these sheets, the templates for 87 warning signs (which is further subdivided into road layout signs, direction of movement signs and symbolic signs) and two yield signs were created.

The colour threshold parameters used were T_S = 0.75, T_H = 0.05, t_S = 0.5 and t_H = 0.1. The area threshold was set at T_A = 50 pixels. The neighbourhood operation in Equation 2 was taken to mean 8-connected pixels. For line fitting, D = 2 pixels, $T_\theta = 5$ degrees and k_S = 0.9 was used. Images were normalised to L = 256 pixels in the vertical and horizontal dimensions. No threshold was applied during classification, that is $T_\delta = \infty$.

The algorithm was tested on images extracted from a number of video sequences. The images were captured at a resolution of 640 x 480 pixels in RGB format and with 8 bits per channel. Video sequences 1 to 5 were captured under good daylight conditions, with the focus on a specific traffic sign(s) and with the sign occupying a relatively large area of the image (from 26 to 235 pixels in the horizontal dimension). Video sequences 6 to 8 were captured from a moving vehicle, with the camera pointed forward in the direction of the vehicle movement, so that different signs are present in the video. These videos present a greater challenge, since the signs are relatively small (from 20 to 60 pixels in the horizontal dimension) and the camera is not always focussed on them.

The results obtained by applying the algorithm are shown in Table 1. Classification was attempted only on signs where a true positive detection was made. To describe the results of the detection and classification processes in a meaningful way, the positive predictive value (PPV) was defined as $PPV = \frac{C}{TPD+FPD}$ and the sensitivity as $SN = \frac{C}{TPD+FND}$. Note that a classification is attempted for each detection ($T_\delta = \infty$). the PPV and sensitivity values may be improved by rejecting detections for which there is a low confidence in correct classification.

As may be expected, the PPV and sensitivity are significantly better for video sequences 1 to 5 than for sequences 6 to 8. An analysis of the images for which errors occurs reveals that false negatives are mainly the result of the signs having a darkish red colour that is not detected through the thresholding process. Noise on the object-interior boundary also result in detection failures. False positives are mainly the result of areas in the background (such as ground or buildings) that masquerade as reddish areas that surround a triangular interior. Classification errors are typically the result of a weak triangular fit that rotates the normalised image.

Table 1: *Summary of the results obtained using the algorithm described in this paper (legend: PR - (horizontal) pixel range, #S - number of signs, TPD - true positive detections, FPD - false positive detections, FND - false negative detections, C - correct classifications, PPV - positive predictive value, SN - sensitivity).*

No	PR	#S	TPD	FPD	FND	C	PPV	SN
1	53-137	640	635	4	5	635	99.4	99.2
2	26-124	484	484	2	0	483	99.4	99.8
3	192-235	244	244	4	0	244	98.4	100
4	61-107	354	354	13	0	354	96.5	100
5	107-204	198	198	9	0	198	95.7	100
6	20-60	95	68	1	27	59	85.5	62.1
7	20-55	115	104	2	11	94	88.7	81.7
8	21-57	102	93	0	9	84	90.3	82.4

4. Conclusions and Future Work

The paper presents a new algorithm for the detection and classification of triangular traffic signs such as warning and yield signs, using colour and shape cues. The algorithm offers robust recognition capabilities under normal daylight conditions in the absence of occlusions.

The algorithm can be extended to other classes of traffic signs, such as control, command and prohibition signs. The approach for these signs could be similar to the approach presented in this algorithm, except that additional colours are used in the threshold process and that other types of shapes (ellipses, octagons, etc.) need to be fitted. In the case of command and prohibition signs, an additional difficulty that needs to be addressed during the normalisation step is to produce a normalised image that is rotation invariant. A more difficult challenge is the recognition and interpretation of sign boards, where there is no standard template and each such sign needs to be interpreted individually for its content.

An important problem to address is the presence of occlusions. The approach presented here is applicable only in the case where occlusions do not intersect the object such that its interior and exterior are connected. One way to solve this problem is to search directly for object-interior boundaries. This could be accomplished by a metric that specifies the extent to which two adjacent pixels are red and white respectively (or other colours for the other classes of signs). These "fuzzy" edges could be thresholded and the algorithm could proceed with labelling and filtering, shape fitting, etc.

The work also needs to be extended to track a traffic sign across multiple frames in a video sequence.

5. Acknowledgments

The research conducted and reported on in this paper was funded by the Council for Scientific and Industrial Research (CSIR), South Africa, under the CSIR Autonomous Rover (CAR) project. The author would also like to thank Willie Brink for his suggestions during the development of the algorithm.

6. References

[1] A. de la Escalera, L. Moreno, M.A. Salichs and J.M. Armingol, "Road traffic sign detection and classification", *IEEE Transactions on Industrial Electronics*, Vol. 44, No. 6, pp. 848-859, 1997.

[2] S.K. Kim and D.A. Forsyth, "A new approach for road sign detection and recognition algorithm", 30th International Symposium on Automotive Technology and Automation, Robotics, Motion and Machine Vision in Automotive Industries, ISATA, 1997.

[3] M.M. Zadeh, T. Kasvand and C.Y. Suen, "Localization and recognition of traffic signs for automated vehicle control systems", *Intelligent Transportation Systems*, SPIE, 1998.

[4] A. de la Escalera, J.M. Armingol and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles", *Image and Vision Computing*, Vol. 11, No. 3, pp. 247-258, 2003.

[5] P. Arnoul, M. Viala, J.P. Guerin and M. Mergy, "Traffic signs localisation for highways inventory from a video camera on board a moving collection van", Intelligent Vehicles Symposium, IEEE, 1996.

[6] T. Hibi, "Vision based extraction and recognition of road sign region from natural colour image, by using HSL and coordinates transformation", 29th International Symposium on Automotive Technology and Automation, Robotics, Motion and Machine Vision in the Automotive Industries, ISATA, 1996.

[7] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition", 10th EWGT Meeting and 16th Mini-EURO Conference, 2005.

[8] D.S. Kang, N.C. Grisworld and N. Kehtarnavaz, "An invariant traffic sign recognition system based on sequential color processing and geometrical transformation", Southwest Symposium on Image Analysis and Interpretation, IEEE, 2004.

[9] X. Gao, K. Hong, P. Passmore, L. Podladchikova and D. Shaposhnikov, "Colour vision model-based approach for segmentation of traffic signs", *EURASIP Journal on Image and Video Processing*, Vol. 2008.

[10] L. Priese, J. Klieber, R. Lakmann, V. Rehrmann and R. Schian, "New results on traffic sign recognition", Intelligent Vehicles Symposium, IEEE, 1994.

[11] L. Priese, R. Lakmann and V. Rehrmann, "Ideogram identification in a real-time traffic sign recognition system", Intelligent Vehicles Symposium, IEEE, 1995.

[12] G.Y. Jiang and T.Y. Choi, "Robust detection of landmarks in color image based on fuzzy set theory", Fourth International Conference on Signal Processing, IEEE, 1998.

[13] N. Bartneck and W. Ritter, "Colour segmentation with polynomial classification", 11th International Conference on Pattern Recognition, IAPR, 1992.

[14] H. Austerirmeier, U. Büker, B. Merstching and S. Zimmermann, "Analysis of traffic scenes using the hierarchical structure code", International Workshop on Structural and Syntactic Pattern Recognition, 1992.

[15] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection", In IEEE Intelligent Vehicles Syposium (IV), pp. 566-571, Parma, Italy, 2004.

[16] J. Miura, T. Kanda and Y. Shirai, "An active vision system for real-time traffic sign recognition", IN Proc. IEEE Conf. on Intelligent Transportation Systems, pp. 52-57, Dearborn, MI, 2000.

[17] J. Torresen, J.W. Bakke and L. Sekania, "Efficient recognition of speed limit signs", In Proc. IEEE Conf. on Intelligent Transportation Systems, Washington, DC, 2004.

[18] D.M. Gavrila, "Traffic sign recognition revisited", In Mustererkennung (DAGM), Bonn, Germany, 1999, Springer Verlag.

[19] P. Paclik, J. Novovicova, P. Somol and P. Pudil, "Road sign classification using Laplace kernel classifier", *Pattern Recognition Letters*, Vol. 21, pp. 1165-1173, 2000.

[20] R. Sedgewick, *Algorithms in C*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[21] "Road and traffic information", $http://www.kzntransport.gov.za/rd\_traffic/enforcement/traffic\_signs/$, Last accessed 17 September 2008.

[22] R.C. Gonzalez and R.E. Woods, "Digital Image Processing", 2nd ed., Prentice Hall, New Jersey, 2002.

Hough Transform Tuned Bayesian Classifier for Overhead Power Line Inspection

Z.R.S. Gaspar, Shengzhi Du, B.J. van Wyk
French South African Technical Institute in Electronics
Tshwane University of Technology South Africa
zulfikhargaspar@univtech.ac.za, dus@tut.ac.za, vanwykb@tut.ac.za

Abstract

Algorithms for automatic electrical power line detection are investigated. In particular, this paper presents a novel tuning technique using the Hough-Transform to improve Bayesian pixel classification accuracy. Connected component analysis is used to remove small components (misclassified pixels) from the classified images. Experimental results presented show that the proposed algorithm outperforms traditional Bayesian classification for power line detection.

1. Introduction

Research on the use of robots to perform inspection of electrical power lines started in the mid-eighties [4]. Ma and Chen [3] proposed an Unmanned Aerial Vehicle (UAV) for overhead power line inspection. In their work automation technology for acquiring videos is described and problems that could be encountered when performing inspections outlined. These problems related to pattern recognition, camera stabilization, acquiring and maintaining the target in the camera's Field Of View (FOV), image degradation, as well as data analysis.

Jones and Earp [8] indicated that the allowable blur caused by camera motion for a static image should be 1-2% in order to have a good image quality for video inspection of power lines. In [9] Jones et al. introduced the "detect and avoid obstacles" principle to be used in the airspace of a small helicopter, thus proposing machine vision and automated path planning as a potential solution for overhead power lines inspection. Jones and Golightly [7] later proposed corner detection and matching methods to keep the intersection of the pole and its cross-arm in the image.

In this study, a novel scheme, combining the Hough Transform (HT) and a Bayesian Classifier (BC), is proposed to detect power lines in the images obtained by a camera mounted underneath a helicopter. The experimental results demonstrate that the proposed algorithm outperforms the BC and HT when used in isolation.

This paper is structured as follows. Bayesian classification and connected component analysis are reviewed in section 2. A description of Hough-Transform tuning technique for the Bayesian classifier is discussed in section 3. Simulation results are presented in section 4, and concluding remarks in section 5.

2. Target detection

2.1. Bayesian pixel classification

Bayesian pixel classification is popular for classifying pixels belonging to a target or the background. The main idea is to determine the posterior probability $P(\text{target}\,|\,(R,G,B))$ that a pixel belongs to objects given its Red, Green, and Blue (RGB) pixel values. Bayes' rule [2] can be used to find the posterior probability:

$$P\left(\text{powerline}|\text{R,G,B}\right)=\frac{p(\text{R,G,B}|\text{powerline})P(\text{powerline})}{p(\text{R,G,B})} \quad (1)$$

RGB values of power line pixels and non-power line pixels were collected to generate a three-dimensional training dataset. Samples of each class (power line and non-power line) were then used to calculate the prior probabilities $P(\omega_i)$. The general multivariate normal class-conditional probabilities were then calculated. The discriminant functions for each class are given by

$$g_i(x) = X^{\mathrm{t}} W_i X + w_i^t X + \omega_{i0},\ i = 1, 2 \quad (2)$$

$$\text{where } W_i = -\frac{1}{2}\sum_i^{-1}, \quad (3)$$

$$w_i = \sum_i^{-1} \mu_{i,} \quad (4)$$

$$\omega_{i0} = -\frac{1}{2}\mu_i^t \sum_i^{-1} \mu - \frac{1}{2}\ln|\sum_i| + \ln P(\omega_i) \quad (5)$$

μ_i is the mean vector, and $\sum_i$ is the covariance matrix of the i th class (Class I is the power lines, Class II is the background).

Images in the database were classified using the decision rule:

ω_1 (powerline class) if $g(x) > 0$; otherwise

ω_2 (non-powerline class) where

$g(x) = g_1(x) - g_2(x)$.

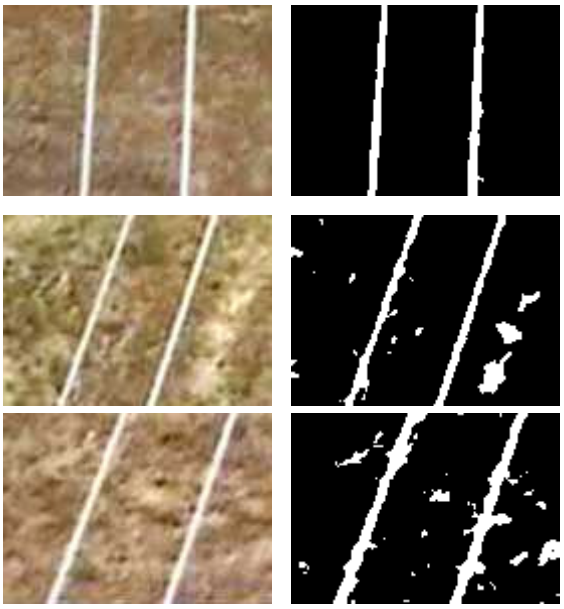


Fig. 1: Bayesian classification results. (On the left: input images from the database. On the right: classification result where the pixels belonging to class I (power lines) are set to be white the pixels belonging to the background are set to be black.)

Fig. 1 shows the classification results. Clearly, there are various small image regions not belonging to power lines that have been misclassified.

2.2. Connected component analysis

Due to uncertainties and noise, some pixels in the images were not classified correctly. After applying the Bayesian pixel classification algorithm (see Figure 1) connected component analysis was used to remove unconnected small regions from classified images.

In this step, the 4 connectivity labeling of regions was used to detect connected components. Assuming that there are at least two power lines, the two largest connected regions were considered as power lines and the remaining smaller regions were removed. Figure 2 shows the detected power lines after the connected component analysis.

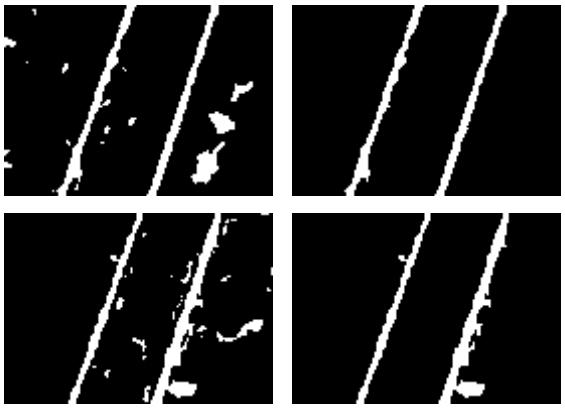


Figure 2: The results of connected component analysis. (On the left: output images from the Bayesian pixel classification. On the right: resulting image after connected component analysis.)

Although unconnected small components due to misclassified are removed (as shown in Fig. 2), this method can only remove the unconnected misclassified regions but not the regions connected to the power lines.

3. Hough transform for tuning Bayesian pixel classification

After the connected component analysis it is clear that there are still some misclassified regions left. In this section a Hough transform tuning scheme is used to improve the performance of Bayesian classifier.

The Hough transform is used to detect lines and curves in images [6]. In this section, we discuss how the power line information obtained by Hough transform can be used to improve the classifier. Our goal in this technique is to:

- Recalculate the prior probabilities $P(\omega_i)$ taking into consideration the power line obtained by Hough transform.
- Recalculate the discriminant functions for power lines and background (see eq. (2))
- Redefine a single discriminant function (see eq. (6)). Notice that the prior probability $P(\omega_i)$ appears in eq. (5)
- Reapply the Bayesian pixel classification algorithm in the original image for a better classification result

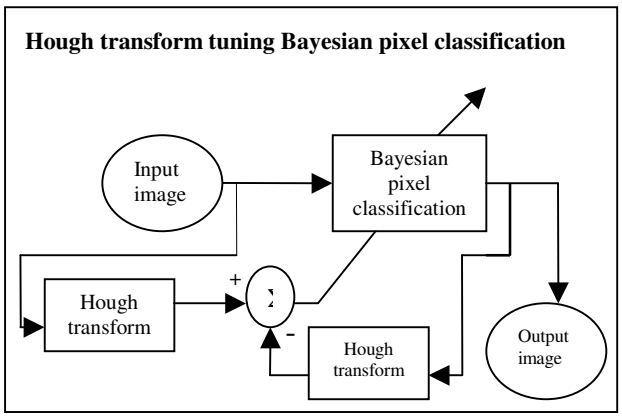


Figure 4: A block diagram representation of Hough transform tuning Bayesian pixel classification

Fig. 4 shows the block diagram for the proposed technique. The Hough transform algorithm is used to find the angle (θ) and position (x) of the power lines in both the binary image obtained as the output of the Bayesian classifier and the original image. The angles and positions obtained are compared and if the angles and positions of the detected objects (candidate power lines) in the two images are not the same, the following steps are executed:

- Use the positions and angles of the lines found in the original image to calculate the lengths of the lines using Eq. (7) and Eq. (8);
- Re-estimate the prior probabilities $P(\omega_i)$ using Eqs. (9) and (10);
- Redefine a single discriminant function using the new prior probabilities;
- Reclassify the image to improve accuracy.

The lengths of the candidate power lines are used to re-estimate the prior probabilities as follows:

$$l_1 = \sqrt{(x_{11} - x_{21})^2 + (y_{11} - y_{21})^2} \qquad (7)$$

$$l_2 = \sqrt{(x_{12} - x_{22})^2 + (y_{12} - y_{22})^2}\,, \qquad (8)$$

where x_{11}, x_{21}, y_{11}, y_{21} are the x and y of line 1 and x_{12}, x_{22}, y_{12}, y_{22} the x and y of line 2 as shown in Fig. 5.

The prior probabilities $P(\omega_i)$ are estimated as

$$P(\omega_1) = \frac{(l_1 + l_2) w_l}{WH} \qquad (9)$$

$$P(\omega_2) = 1 - P(\omega_1)\,, \qquad (10)$$

where w_l (tuning parameter) is an estimate of the width of the candidate power lines, W and H the width and height of the image, respectively.

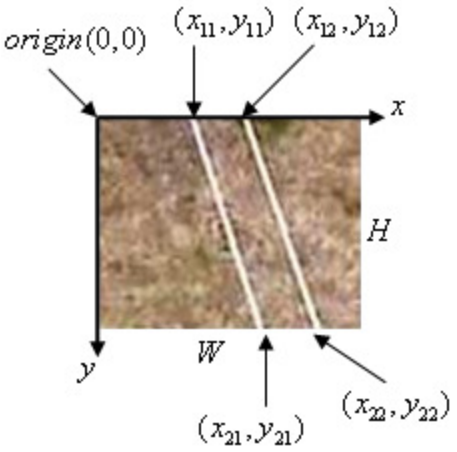


Figure 5: the parameters used in Eq. (7), (8), (9) and (10).

Figure 5 depicts the idea behind the technique. Note that the lines are similar therefore we assume that the width w_l is the same for both lines. The optimal width w_l was determined by trial and error.

4. Simulation results

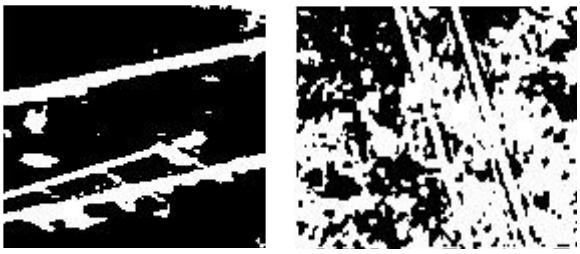
In this section, two images where the Bayesian pixel classifier failed to detect power lines, are used to demonstrate the performance of the proposed scheme. Refer to Fig. 8.

Figs. 8(a) are the original images with two power lines in the view. Figs. 8(b) are the results of Bayesian pixel classification. The classification performance is so poor that the power lines are almost totally obscured by misclassified regions (indicating that the original Bayesian method is severely affected by noise and uncertainties).

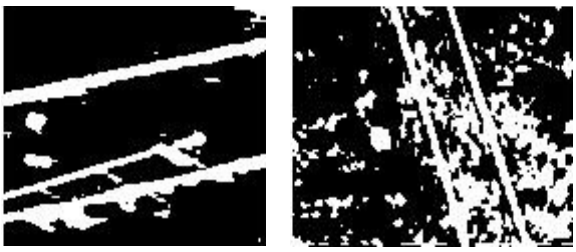
Figs. 8(c)-(e) are the results obtained using the Hough transform to tune Bayesian pixel classification. It is clear that when the Hough transform is used for tuning that classification accuracy is significantly improved. The classification performance is remarkably better when the prior probabilities are more accurately estimated.

(a) original images

(b) poor classified images



(c) $w_l = 1$



(d) $w_l = 0.5$

(e) $w_l = 0.0005$

Figure 8: Hough transform tuning Bayesian pixel classification

5. Conclusion and future work

A technique to improve Bayesian pixel classification using the Hough transform was proposed in this paper. The Hough transform tuned Bayesian pixel classifier outperformed the classical Bayesian approach. Future work aims at generalizing the work to a wider class of problems.

6. Acknowledgements

The authors would like to thank the support of Tshwane University of Technology together with French South African Institute in Electronics.
Special thanks also go to CSIR and Aerial Concepts for providing the real-time video recordings.

7. References

[1] R.O. Duda, P.E. Hart and D.G. Stork, "Pattern classification", 2nd Edition, ISBN 0-471-05669-3, Wiley-Interscience, 2001.

[2] S. Theodoridis, K. Koutroumbas, "Pattern recognition", ISBN 0-12-686140-4, Academic Press, 1999.

[3] L. Ma, Y. Chen, "Aerial Surveillance System for Overhead Power Line Inspection", December 2004.

[4] J.M. Giraldo, "Robotics Applications in Power Line Maintenance", 2007. [Online] Available: [http://powerencounter.blogspot.com/2007/02/robotics-applications-in-power-line.html

[5] D. Jones, M. Williams, G. Earp, "Robotic Inspection of Power Lines". [Online] Available: http://www.informatics.bangor.ac.uk/~dewi/ci_grp/ripl_web.htm

[6] R.O. Duda, P.E. Hart, "Use of the Hough Transform to detect lines and curves in pictures", in Communications of the ACM, volume15, pp. 11-15.

[7] Jones, D.I., Golightly, I., Earp, G.K., *Corner detection and matching for visual tracking during power line inspection,* EA Technology Report No. 5537, 2002.

[8] Jones, DI, Earp, GK, *Camera sightline pointing requirements for aerial inspection of overhead power lines*, Electric Power Systems Research 57(2), p. 73-82, 2001.

[9] Williams, M, Jones, DI, Earp, GK, *Obstacle avoidance during aerial inspection of power lines*, Aircraft Engineering & Aerospace Technology, 73(5), p. 472-479, 2001.

Alignment invariant image comparison implemented on the GPU

Hans Roos
Highquest, Johannesburg
hans.jmroos@gmail.com

Yuko Roodt
Highquest, Johannesburg
yuko@highquest.co.za

Willem A. Clarke, MIEEE, SAIEE
University of Johannesburg
willemc@uj.ac.za

Abstract

This paper proposes a GPU implemented algorithm to determine the differences between two binary images using Distance Transformations. These differences are invariant to slight rotation and offsets, making the technique ideal for comparisons between images that are not perfectly aligned. The parallel processing capabilities of the GPU allows for faster implementation than on traditional desktop processors. In order to take full advantage of this all aspects of the algorithm was implemented on the GPU.

Key words: Distance transform, binary image, GPU, parallel processing.

1. Introduction

In the field of image processing, image comparison has a wide variety of applications. These applications range from image retrieval to image registration [1]. In this paper we are proposing to make use of graphics processing units (GPU), parallel processing techniques and distance transformations to compare images invariant to slight rotation or offsets.

The GPU was selected for this purpose due to its computational power. Recent advances in graphics architecture have ensured that GPUs have extensive memory bandwidth along with tremendous increases in its computational horsepower. These increases are clearly advantageous. Other advantages of GPU algorithm implementations include the fact that GPUs can perform these operations faster and their cost versus computational power is much lower than that of central processing units (CPU) [7, 8]. GPUs also provide better performance per thread than CPUs can provide [7]. The mentioned advantages have given GPUs a popular position amongst researchers to use them for general purpose computations [8, 9]. GPUs do however have their own set of disadvantages: "they lack some fundamental computing constructs" [8]. The absence of these constructs make GPUs ill suited for tasks such as cryptography.

The Distance Transformation (DT) is an operation performed on binary images (images containing black and white pixels; or feature and non-feature pixels) which returns a greyscale representation where each pixel value represents *that* co-ordinate's distance from its nearest feature pixel in the binary image [3, 9]. The Distance Transform is an important tool in image processing; however its uses have extended into other fields including that of pattern recognition computer vision, computer graphics to name a few [4, 9].

Various methods of determining Distance Transformation exist. In this paper we utilize the 4-connected distance (otherwise known as the city block distance map) [6]. Other distance maps such as the Euclidean distance map may also be used. The Euclidean map is described as a map which corresponds to how real world objects are measured, which makes it easily interpreted. That said, the brute force approach to calculating the Euclidian distance is not feasible as it involves measuring the distance between every feature pixel and every non-feature pixel yielding a computational complexity of $O(n^2)$ for every pixel [11]. However the 4-connected approach is the least complex and provides a good enough approximation of the distance for the purpose of this application.

2. Definitions

In this section we will more clearly define the concepts of binary images and distance transformations. These definitions are to be used at a later stage.

A point on an image can be defined in terms of x and y such that $x \in \{1,...,width\}$ and $y \in \{1,...,height\}$, where width and height are the dimensions of the image. Hence (x, y) is an arbitrary point on the image.

Adding to the earlier definition of a binary image it can be stated that binary images contain foreground pixels and background pixels. The foreground pixels represent the objects in the image. Thus it can be written as follows:

A binary image can be represented as a function, $I(x, y)$ where $I(x, y) \in \{O, B\}$. O and B represents object and background pixels respectively; in terms of implementation $I(x, y) \in \{1,0\}$. In other words the notation states that the texture value at the point *(x, y)* is either a foreground pixel or a background pixel.

For the definition of the Distance Transform, we can say: the Distance Transform can be represented by the function, $D(x, y)$ where $D(x, y) \in \{0,...,1\}$. The set $\{0,...,1\}$ is the distance to the nearest foreground pixel, the range of this set may vary depending on implementation, convention and preference. For example $D(x, y) \in \{1,...,imagesize\}$

In this paper we will refer to the input image and the image to be compared, as $I_1(x, y)$ and $I_2(x, y)$ respectively. For each comparison two Distance Transformations are required, one for all the distances to the nearest object, $D_O(x, y)$ and the other all the distances to the nearest background pixel $D_B(x, y)$. These

transformations are only done for one of the input images; however both are done on the same input image.

3. Implementation

In this section we will discuss the implementations of the main components of the papers, namely the Distance Transformations and then the comparison algorithm.

3.1 Distance Transform Implementation

Initially our distance map was approximated using the concept of a local distance map. The distances were calculated around each pixel, but only for a small region or window as implemented by É. Baudier et al using the Hausdorff distance [1]. However, our implementation used a circular window around each point and the Euclidian distance between each pixel in the window and the centre of the window.

The 4-connected distance transform is implemented by selecting the minimum value between a pixel's four surrounding values (above, below, left and right) and storing them into an interim distance map. This interim distance map is then passed back and is recursively processed until all the distances have been computed [6].

3.2 Example of the distance transform

For the purpose of clarity the colours of the images have been inverted, i.e. black represents the foreground and white represents the background as opposed to the norm where white represents the foreground (features) and black the background (non-features). Figure 1 shows a binary image containing two objects, where figure 2 represents figure 1's distance transformation. In figure 2, the darker the colour, the further away from the object the point is.

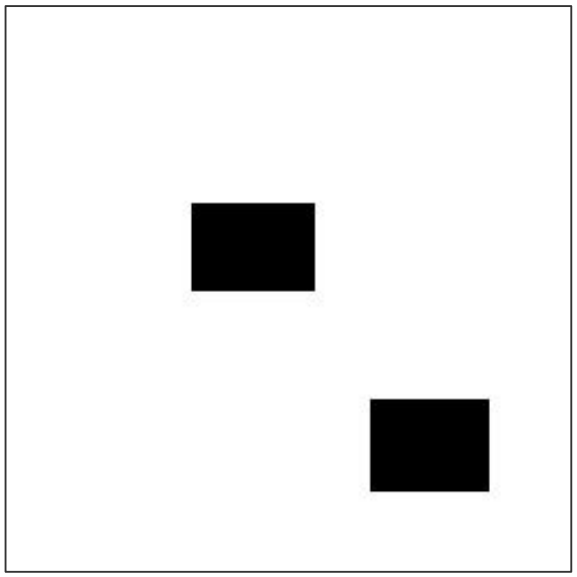


Figure 1. Binary image containing 2 objects represented as black pixels.

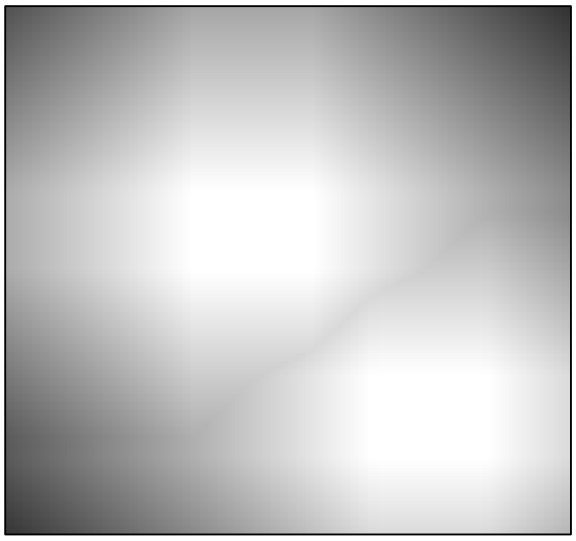


Figure 2. Figure 1's Distance Transform. Darker colours are far from the objects.

3.3 Image Comparison Implementation

The proposed algorithm is as follows: in order to compare the two images the distance maps of the first image, $I_1(x, y)$, have to be computed; with respect to both background pixels and foreground pixels i.e. two distance maps are created. One containing distances to the nearest white feature, $D_O(x, y)$ and another containing distances to the nearest black feature, $D_B(x, y)$.

Once these distance maps have been acquired a pixel at point (x, y) from the second image, $I_2(x, y)$, is compared to the 2 distance maps. If the pixel at the *current* point is black the corresponding distance value in the nearest-to-black map, $D_B(x, y)$, is returned. If the pixel is white the corresponding value is returned from the nearest-to-white map, $D_O(x, y)$. The output of the algorithm then represents the differences in the image, or rather how far a point is to its closest feature. Figure 3 shows a graphical representation of the algorithm where D_O, D_B and I_2 are the input textures.

In terms of the GPU implementation of the algorithm; OpenGL fragment programs were coded to generate the two distance maps of the first image, I_1. The distance maps are stored in the GPU's memory as a texture (or image). This is done to avoid losing the GPU's performance advantage by passing information back and forth between the GPU and CPU. The second distance map is done using the same algorithm as the first. However, the inverse of the first image is used as an input. The inversion is also implemented on the GPU. A separate fragment program was created in order to do the comparison on the GPU. The result of the comparison is then stored as a texture and then displayed on screen.

The pseudo code below is the algorithm for comparing the 2 images as implemented in the comparison fragment program. The value `current_Pixel` is the current texture value from the second image, I_2.The value `current_Distance` is the texture value from either one of the two distance maps at the

current *(x, y)* position; the same position where `current_Pixel` was obtained. The `current_Distance` is returned to a new texture in order to make the result graphically viewable.

```
current_Pixel<- current_Texture from I2

if current_Pixel = black then
   current_Distance<- value from DB
else
   current_Distance<- value from DO

return current_Distance
```

From the algorithm it is easy to see that a threshold can be added which can be used to make decisions based on the result, for example to discard any differences that are not intense enough and only keep the differences that are clear enough.

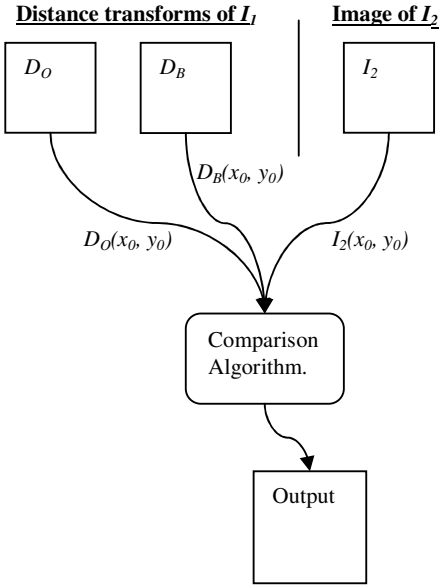


Figure 3. Graphical representation of the algorithm

4. Experimental Setup

The algorithm was tested on two different systems; both systems had Windows XP Professional 32Bit Service Pack 3 as operating systems. The main specifications of the two systems are as follows:

	System A	System B
CPU	AMD Athlon X2 4200+	AMD Athlon 3200+
GPU	8800GTX	6800GE
RAM	2048 MB	2048 MB

Table 1. System used in the performance test of the algorithm

The systems were chosen as they are from two different eras in terms of performance, System A being a lot more powerful than System B especially in terms of graphics processing capabilities.

	8800GTX	6800
Pixel Shaders	128	16
Core Clock (MHz)	575	350
Memory (MB)	768	256
Memory Clock (MHz)	900 (DDR3)	500 (DDR3)
Shader Model	4.0	3.0

Table 2. GPU specifications of the test systems

The algorithm was initially written and implemented in RenderMonkey (version 1.81) to test and verify the OpenGL syntax. Once verified, the OpenGL was implemented in C++ in order to do more accurate performance tests and comparisons between the two systems.

5. Results

The results of the tests will be discussed in the following section. Firstly we will look at the results of the image comparisons followed by the performance results

5.1 Comparison Results

The algorithm was tested on various images. One of the tests was done on a "spot the difference" game containing eight differences. The results are discussed below.

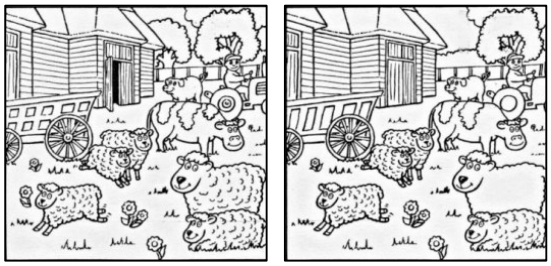


Figure 4. Input images. Spot the difference game containing 8 differences [10].

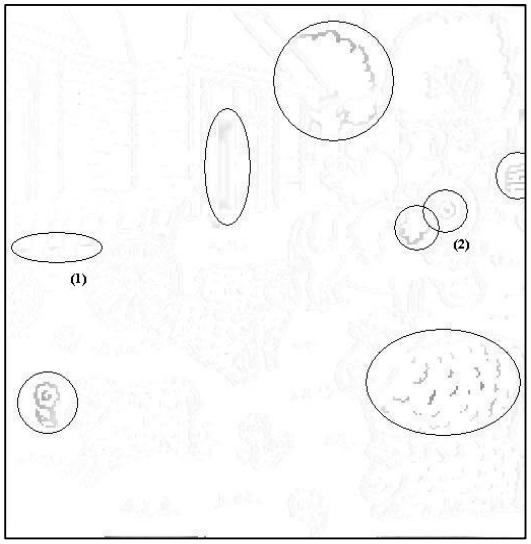


Figure 5. Highlighted differences between the images in figure 4.

Figure 5 (above) highlights the differences between the images in figure 4. Circles were placed round all eight of the differences. The comparison seems to fail in regions where it is difficult to compute distances as the differences are only subtle changes in shape, see points 1 and 2 on figure 5. The fact that these appear as light grey, shows that the algorithm is only recognizing a minor difference. The grey outlines of the images above are due to the fact that the images are not perfectly aligned for demonstration purposes; showing the invariance property of the algorithm.

Further tests were done with regards to more practical applications such as template matching and character recognition. Figure 6 a and b (the numbers "3" and "8") were compared. The result of the comparison can be seen in figure 7.

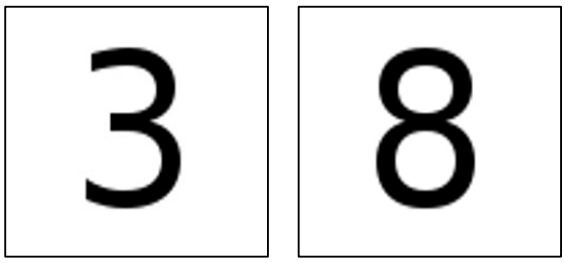


Figure 6 a and b. The second part of the algorithm test demonstrating possible uses in template matching and character recognition.

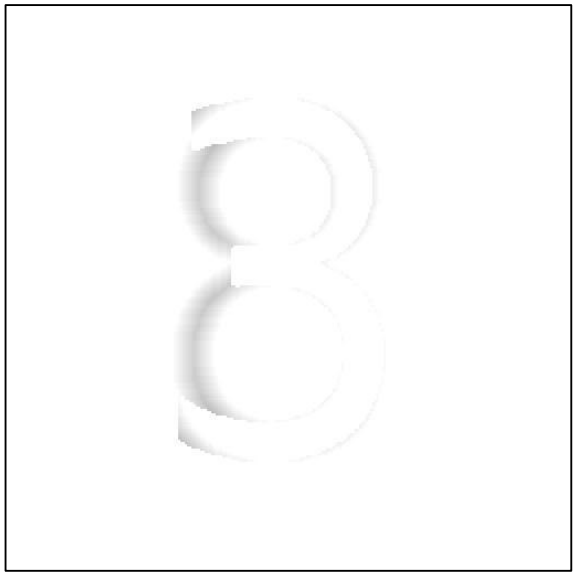


Figure 7. Comparison results between figure 6 a and b

When comparing the image (example figure 6 a) to a slightly rotated version of itself (figure 8 a) using the proposed technique, only minor differences are highlighted (see figure 8 b). These changes can easily be discarded. However, when comparing our results to an XOR comparison, the rotation is clearly visible in the output (see figure 8 c). Rotating the image further, still only highlights minor changes when using our technique. Again the XOR comparison reveals very clear changes due to the rotation (see figure 8 d, e and f).

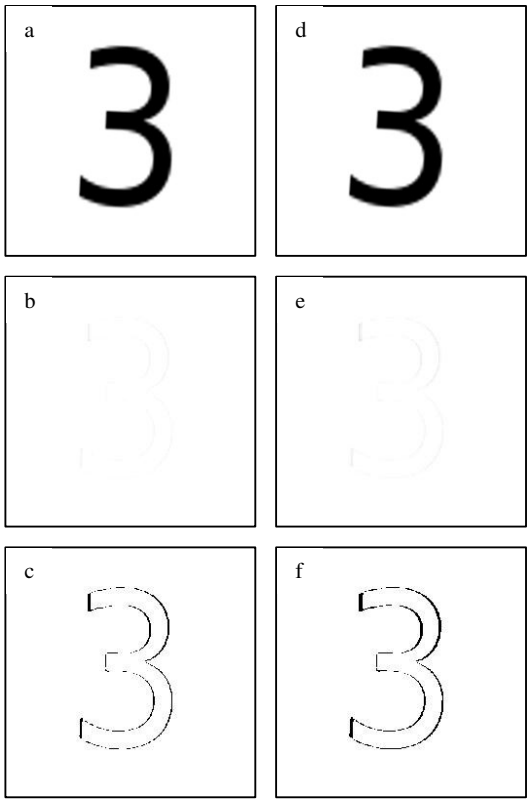


Figure 8 a – f. Comparison of images with rotation using the proposed algorithm and XOR comparisons.

Figure 9 demonstrates a situation where the comparison will begin to fail. The rotation of the image is much greater than the previous examples. However it will still be possible to threshold out and discard many of the errors, but in such an extreme case it leaves a lot of room for error.

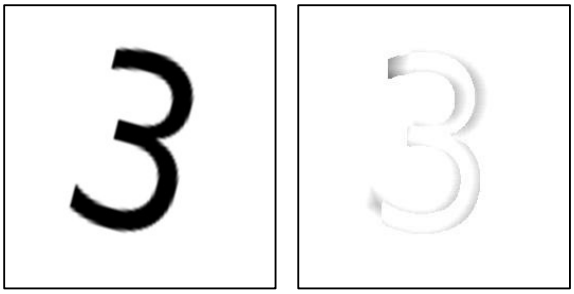
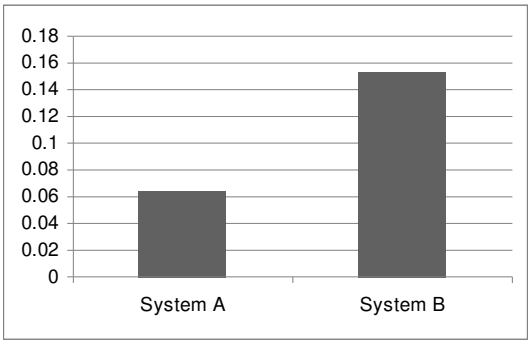


Figure 9. Comparison demonstrating a larger rotation, where inevitably the test will begin to fail.

5.2 Performance test results

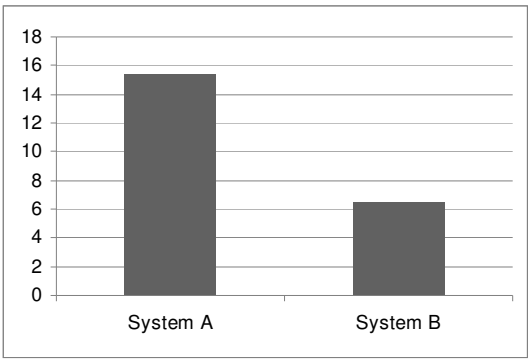
The performance was tested in terms of processing time and frames per second (fps). When we refer to frames per second we are referring to actual renders per second which is the inverse of the processing time. The tests on both systems were done using 128 iterations to calculate each Distance Transformations. Images of size 1024 by 1024 were used in the tests. The performance results have been summarized as follows.

Processing time (s)



Graph 1 Processing time (s)

Frames per second (fps)



Graph 2 Frames per second (fps)

The above results show that the performance of the algorithm is sufficient even on older systems. Considering the fact that number of calculations done to perform just one of the Distance Transformations is a staggering 134,217,728 iterations ($1024 \times 1024 \times 128 = 134217728$).

6. Conclusion

We proposed a technique to comparing images using the concept of distance maps. The entire algorithm was implemented on the GPU in OpenGL to take maximum advantage of the performance advantage of the GPU has over traditional desktop processors.

The comparisons were invariant to slight rotation and offset as seen in the comparison results. This invariance makes the technique useful in the fields of template matching and character recognition.

7. References

[1.] Baudrier É, Nicoler F, Millon G and Ruan S, "Binary-image comparison with local dissimilarity mapping", Pattern Recognition, Vol. 41, pp. 1461-1478, 2008

[2.] Kim HY and Araújo SA, "Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast," IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science, vol. 4872, pp. 100-113, 2007.

[3.] Gavrila DM, "Multi-Feature Hierarchical Template Matching Using Distance Transforms", Proceedings of 14th International Conference on Pattern Recognition (ICPR'98), Vol. 1, pp 439, 1998

[4.] Felzenszwalb PF, Huttenlocher DP, "Distance Transforms of Sampled Functions", Cornell, Computing and Information Science, 2004

[5.] Saude AV, Couprie M, De Alencar Lotufo R, "Distance Transform to seeds: computation and application", Proceedings of 8th International Symposium on Mathematical Morphology, Vol. 2, pp 15-16, 2007

[6.] Bailey DG, "An Efficient Euclidean Distance Transform", International conference on combinatorial image analysis, IWCIA, Vol. 3322, pp. 394-408, 2004

[7.] Owens J, Davis UC, "GPU Architecture Overview" Proceedings of International Conference on Computer Graphics and Interactive Techniques, Article 2, 2007

[8.] Owens J, Luebke D, Govindaraju N, Harris M, Krüger J, Lefohn A and Purcell T. "A Survey of General-Purpose Computation on Graphics Hardware". Proceedings in Eurographics, State of the Art Reports, pp. 21-51, 2005

[9.] Rong G, Tiow-Seng T, "Jump Flooding in GPU with Applications to Voronoi Diagrams and Distance Transform", Proceedings in ACM Symposium on Interactive 3D Graphics and Games, pp. 109-116, 2006

[10.] "Wimphole Home Farm", "spot the difference image" found at http://www.wimpole.org/spot.html, 25 September 2008

[11.] Fabbri R, JC Torelli, Bruno OM, "2D Euclidean Distance Transform Algorithms: A Comparative Survey", ACM Computing Surveys, Vol. 40, No. 1, Article 2, 2008.

Data requirements for speaker independent acoustic models

Jacob A. C. Badenhorst[1], Marelie H. Davel[2]

[1]School of Electrical, Electronic and Computer Engineering,
North-West University, Potchefstroom, South Africa
[2]HLT Research Group, Meraka Institute, CSIR, South Africa
jbadenhorst@csir.co.za, mdavel@csir.co.za

Abstract

When developing speech recognition systems in resource-constrained environments, careful design of the training corpus can play an important role in compensating for data scarcity. One of the factors to consider relates to the speaker composition of a corpus: finding the appropriate balance between the number of speakers and number of speaker-specific utterances. We define a model stability measure based on the Bhattacharyya bound and apply this to analyse inter- and intra-speaker variability of a training corpus. We find that the different phone groups exhibit significantly different behaviour across groups, but within groups similar trends are observed. We demonstrate that, at a predictable point, additional data from one speaker does not contribute further to modelling accuracy and demonstrate the trends that can be expected when additional speakers are added. We also note that inter- and intra-speaker variability are independent effects, with some phone groups requiring more speaker-specific data, and others more cross-speaker data. More complex models require more training data, but exhibit similar overall trends to a simple Gaussian model.

1. Introduction

When building speech recognition systems for the languages of the developing world, it is often necessary to create new speech recognition corpora with limited resources. It is therefore important to design a speech corpus carefully in order to compensate to the extent possible for the scarcity of data. For example, even though the Lwazi corpus [1] is currently the most comprehensive speech recognition corpus available for South African languages, it contains only approximately 2 hours of annotated audio for each of the 11 languages – significantly less resources than typically used in the construction of a speech recognition system.

When designing a speech corpus, we are interested in the interplay between the number of speakers and number of utterances per speaker on the estimation accuracy of acoustic models for different phone types. Adding additional utterances from one speaker is more cost-efficient than adding additional speakers. How should the variety of speakers and utterances per speaker be balanced? Can we estimate whether the cost of adding additional data will be justified?

In this paper, we address these questions in the context of standard Gaussian Mixture Models (GMMs) as employed in a Hidden Markov Model (HMM) based speech recognition system. Specifically, we utilise a Monte Carlo estimation of the Bhattacharyya bound to characterise the similarity of two models, and use the stability of this measure when estimated for different subsets of the same data set to characterise the estimation accuracy that can be obtained with a specific type of acoustic model, using that data set. The effect of an increasing number of speakers and utterances is then analysed using this technique for acoustic models of different types of phones, and some interesting trends are observed.

The similarity technique we define here also allows us to understand the similarity between different phones, for example, the same nominal phone across languages. This is useful when combining training data across languages in order to compensate for a lack of sufficient training data, a useful strategy in resource-scarce environments. By evaluating model stability we can better understand whether the measured differences between models stem from an actual variance in the data, or from variability introduced by estimation errors, and also estimate whether different models are similar enough to support data sharing.

The paper is structured as follows: In Section 2 we discuss related work and provide some background on the Bhattacharyya bound. In Section 3 we describe the general technique we use for the analysis of model similarity and stability. In Section 4 we use this technique to analyse our data set, specifically with regard to the effect of an increasing number of speakers and utterances for different types of phones and types of acoustic models, and discuss the trends observed. Section 5 contains some concluding remarks.

2. Background

Data selection strategies for speech recognition purposes typically focus on selecting informative subsets of data from large corpora, with the smaller subset yielding comparable results [2]; or the use of active learning to improve the accuracy of existing speech recognition systems [3]. Both techniques provide a perspective on the sources of variation inherent in a speech corpus, and the effect of this variation on speech recognition accuracy.

In [2], Principle Component Analysis (PCA) is used to cluster data acoustically. These clusters then serve as a starting point for selecting the optimal utterances from a training database. As a consequence of the clustering technique, it is possible to characterise some of the acoustic properties of the data being analysed, and to obtain an understanding of the major sources of variation, such as different speakers and genders. Interestingly, the effect of utterance length has also been analysed as a main source of variation [3].

Active and unsupervised learning methods can be combined to circumvent the need for transcribing massive amounts of data [3]. The most informative untranscribed data is selected for a human to label, based on acoustic evidence of a partially and iteratively trained ASR system. From such work, it soon becomes evident that the optimisation of the amount of variation

inherent to training data is needed, since randomly selected additional data does not necessarily improve recognition accuracy. By focusing on the selection (based on existing transcriptions) of a uniform distribution across different speech units such as words and phonemes, improvements are obtained [4].

In the current work, the separability of two probability density functions is measured by a widely-used upper bound of the Bayes error, namely the Bhattacharyya bound [5]. If the Bayes error is given by

$$\epsilon = \int min[P_1 p_1(X), P_2 p_2(X)] dX \quad (1)$$

(with P_i and $p_i(X)$ denoting the prior probability and class-conditional density function for class i, respectively), the upper bound of the integrand can be determined by making use of the fact that

$$min[a, b] \leq a^s b^{1-s} \quad 0 \leq s \leq 1 \quad (2)$$

and is called the Chernoff bound, with s a parameter to be estimated through optimisation (Eq. 2 states that the geometric mean of two positive numbers is larger than the smaller one). If the condition for selection of an optimal s is relaxed by choosing $s = 0.5$, this simplified bound is referred to as the Bhattacharyya bound:

$$\epsilon = \sqrt{P_1 P_2} \int \sqrt{p_1(X) p_2(X)} dX \quad (3)$$

When both density functions are Gaussian with mean M_i and covariance matrix Σ_i, integration of ϵ leads to a closed-form expression for ϵ

$$\epsilon = \sqrt{P_1 P_2} e^{\mu(1/2)} \quad (4)$$

where

$$\mu(1/2) = \frac{1}{8}(M_2 - M_1)^T \Big[\frac{\Sigma_1 + \Sigma_2}{2}\Big]^{-1} (M_2 - M_1) + \frac{1}{2} ln \frac{\big|\frac{\Sigma_1 + \Sigma_2}{2}\big|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (5)$$

is referred to as the Bhattacharyya distance. For complex distributions, the Bhattacharyya bound can be estimated via Monte Carlo simulation.

3. Approach

We approach the task of analysing model estimation accuracy by first defining an appropriate similarity measure [6] and then defining a measure of model estimation stability based on this similarity measure. These two techniques are described below.

3.1. Measuring model similarity

From Eq. 3 and using the sample value of the expectation of the integral, we derive an estimator for the Bhattacharyya bound of two Gaussian Mixture Models. In practice we calculate:

$$\epsilon = \sqrt{P_1 P_2} \frac{1}{n_1 + n_2} \Big[\sum_{i=1}^{n_1} \sqrt{\frac{p_2(x_i)}{p_1(x_i)}} + \sum_{i=1}^{n_2} \sqrt{\frac{p_1(x_i)}{p_2(x_i)}} \Big] \quad (6)$$

where x_i are the actual samples and both n_1 and n_2 are the number of samples with regard to each of the two probability densities respectively. For our purpose we assume that the prior values $P_1 = P_2 = 0.5$ and utilise equal numbers of samples drawn from each distribution. We ensure that we utilise a sufficient number of samples by first selecting a set of model pairs that cover a range of similarity values, and then evaluating the variance observed in the estimated bound between these model pairs over various runs (initiated with different sampling seed values) using an increasing number of samples per run. The number of samples is then selected where the variance across different runs falls below an acceptable threshold.

Note that the ϵ error bound can easily be converted to a distance measure using Eq. 4 but we find it more intuitive to work with the bound directly.

3.2. Measuring model estimation stability

In order to estimate the stability of an acoustic model, we separate the training data for that model into a number of disjoint subsets. All subsets are selected to be mutually exclusive with respect to the speakers they contain. For each subset, a separate acoustic model can be trained, and the Bhattacharyya bound between each pair of models is calculated. By calculating both the mean of this bound and the standard deviation of this measure across the various model pairs, a statistically sound measure of model estimation stability is obtained.

4. Analysis and results

4.1. Data and experimental setup

We use the November 1992 ARPA Continuous Speech Recognition Wall Street Journal Corpus as training data for our analysis. The dataset consists of 102 speakers recorded over the same channel. This enables us to experiment with up to 20 hours of data, which is comparable to the amount of data contained in the Lwazi corpus. In order to be able to control the number of phone observations used to train our acoustic models, we first train a speech recognition system and then use forced alignment to label all of the utterances.

We perform speech recognition using standard HMMs with three emitting states, tied across models, each containing up to 12 GMMs trained on 39-feature MFCC-based vectors (13 MFCCs, deltas and double-deltas with cepstral mean subtraction). Similar feature vectors are utilised in our analysis.

Using the process discussed in section 3.1, we estimate the number of samples required for our Bhattacharyya estimator and find that 20,000 samples are sufficient for our purpose. Table 1 summarises the number of samples required to keep standard deviations below a threshold of 0.0100 for the various model comparisons. With 20,000 samples, the standard deviation among different estimations of bounds between GMMs containing up to 6 mixtures are below 0.0020 for very similar phones and below 0.0061 for quite dissimilar phones. (Model pairs with Bhattacharyya bounds of approximately 0.5 and 0.1 respectively). We also find that with 20,000 samples and a single GMM, these estimates are within 0.0002 and 0.0020 from the corresponding analytically calculated values. We separate our data set into 5 disjoint subsets and estimate the mean of the 10 distances obtained between the various model pairs.

4.2. Initial analysis

During our initial analysis we develop speaker-and-utterance three-dimensional plots for acoustic models of different phone types at two levels of model complexity: a simple single Gaussian model (GMM with 1 mixture) and a complex 6-mixture

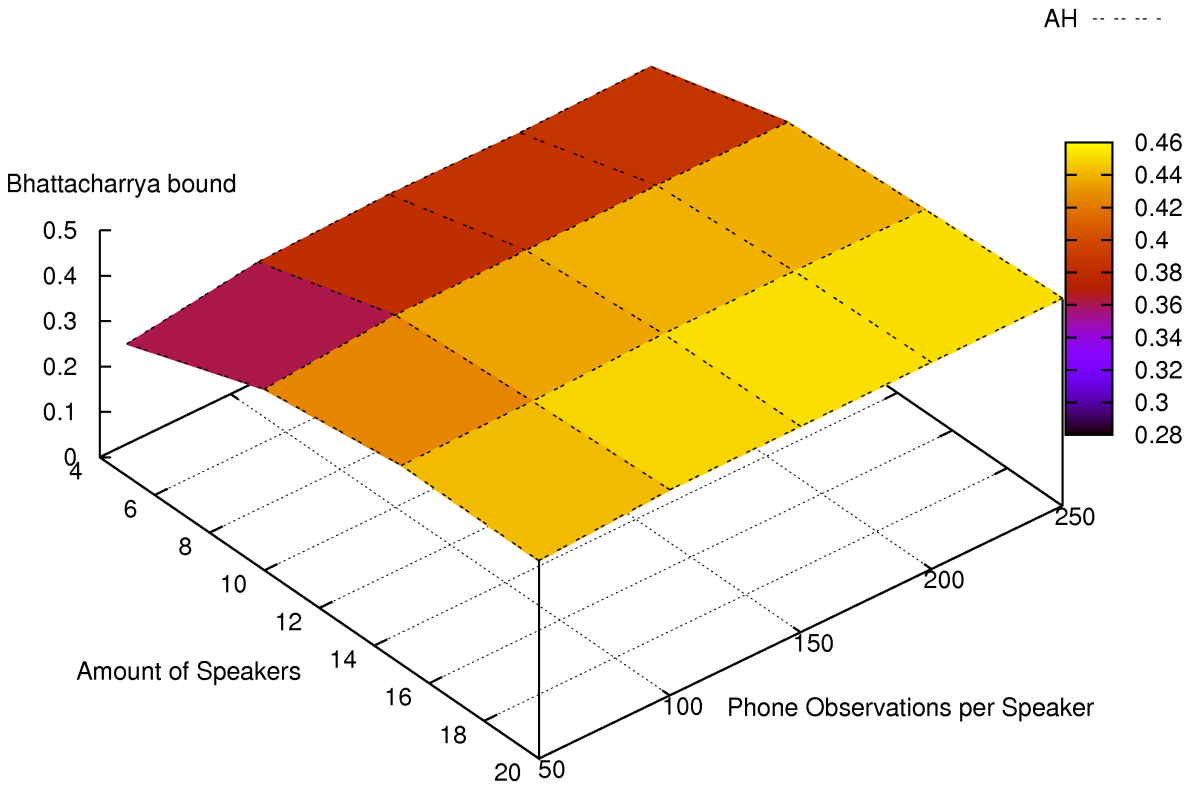


Figure 1: *Speaker-and-utterance three-dimensional plot for the phone /ah/*

Num mixtures	Samples required	ϵ	σ_s
1	5,000	0.1	0.0100
		0.3	0.0073
		0.5	0.0031
2	10,000	0.1	0.0062
		0.3	0.0060
		0.5	0.0018
4	20,000	0.1	0.0045
		0.3	0.0020
		0.5	0.0017
6	20,000	0.1	0.0061
		0.3	0.0045
		0.5	0.0020

Table 1: *Number of samples required for accurate estimation of bounds.*

GMM. (The choice to utilise a 6-mixture GMM was made to balance high speech recognition accuracy for our data set with computational requirements during bound estimation.) Each plot indicates the value of the Bhattacharyya mean, as described in Section 3.2, as a function of both the number of speakers in the training corpus and the number of phone occurrences per speaker. As the mean value shown is an estimate of the Bhattacharyya bound, this value should approach 0.5 once a model is fully trained on an optimal set of data. An example of such a plot for the phone /ah/ is shown in Figure 1.

From this analysis the following was observed: (1) A specific number of speakers and phone occurrences result in significantly different results for the different phones. (2) While phones from the different broad phone categories (such as vowels, plosives or fricatives) exhibit varying learning behaviour, phones within a specific phone group follow remarkably similar trends. (3) Similar trends are observed when utilising either the more simple or the more complex acoustic model.

These initial observations are explored further in the following sections for a number of broad phone categories. For each broad category, a number of representatives are selected to illustrate the trends observed. Specifically, the following phones are selected: /ah/ and /ih/ (vowels), /n/ (nasals), /l/ and /r/ (liquids), /d/ (voiced plosives), /t/ and /p/ (unvoiced plosives), /z/ (voiced fricatives) and /s/ (unvoiced fricatives), after verifying that these phones are indeed representative of the larger groups. Given (3) above, the next two sections first discuss trends obtained using the simpler model, before the effect of moving towards a more complex model is discussed in Section 4.5

4.3. Number of phone occurrences required per speaker

In this section we aim to understand whether a saturation point is reached after which additional examples of phones by a specific speaker no longer improve the accuracy of the speaker independent acoustic model for that phone. We therefore take a

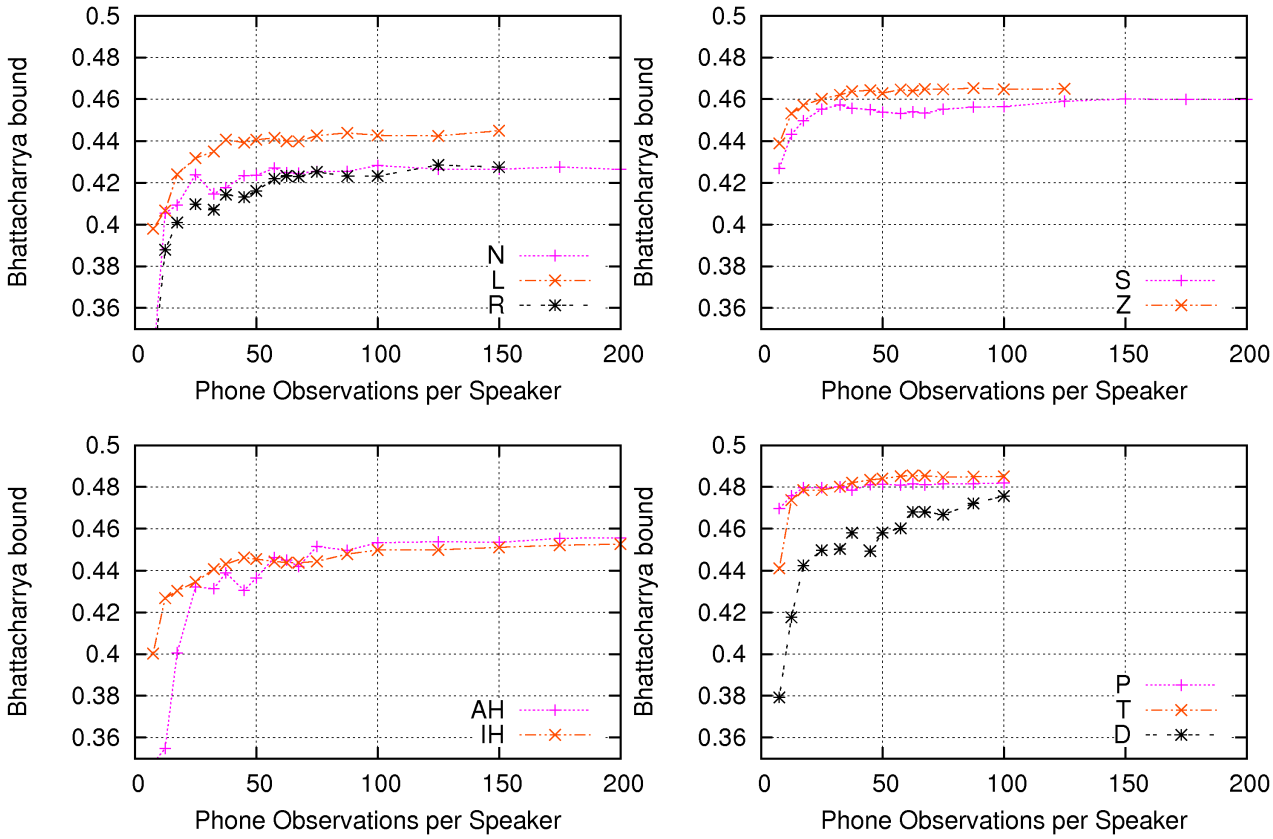


Figure 2: *Effect of number of phone utterances per speaker on mean of Bhattacharyya bound for different phone groups using data from 20 speakers*

cross-section of the 3-D plot in Figure 1, for a specific number of speakers (20) and evaluate the effect of increasing the phone observations per speaker. As clearly demonstrated in Figure 2, the means all reach an asymptote quite quickly and for 20 speakers, this asymptote does not yet approach the ideal 0.5 level for most of the phone types. When this experiment is repeated with 50 speakers, even fewer phone observations per speaker are required to reach the asymptote, and all the asymptotes are also nearer to the ideal level of 0.5. Interestingly though, the total numbers of phone observations necessary for the model of a phone to reach the asymptote are comparable for the 20 and 50 speaker cases. [1]

For the different phone types we observe that vowels are the slowest to reach the saturation point (at approximately 100 phone observations per speaker in the 20-speaker case) while unvoiced plosives and fricatives stabilise the most quickly, reaching this point at only 35 phone observations for /s/, 45 phone observations for /z/ and 25 phone observations for /p/ or /t/. There is a clear difference between the unvoiced and voiced versions of the plosives, with voiced versions taking significantly longer to stabilise (compare /d/ at 85 phone observations with /t/ at 25 phone observations). For most phones, those that saturate more quickly achieve a higher bound (closer to the ideal 0.5). However for some phones, such as /d/, a large number of phone occurrences are required per speaker, but the higher bound indicates that fewer speakers are required to obtain an accurate estimate. Similarly, the fricatives (/s/ and /z/) reach their asymptote very quickly, but this asymptote is fairly low, indicating low intra-speaker but high inter-speaker variability for this phone.

[1]Note that in order to be able to evaluate the effect at 50 speakers, only 2 models could be trained and 1 distance estimated, in comparison to the 5 models and 10 distances possible at the 20-speaker level.

4.4. Number of speakers required per phone

In this section we aim to understand the effect of adding additional speakers to a training corpus during acoustic model construction. We select a number of phone observations per speaker (100) where the asymptote has already been reached for all phones if 20 training speakers are employed. We construct a training set where we systematically add 100 observations for each new speaker. The results of this experiment are shown in Figure 3.

This time, the asymptote is not reached, and it is clear that additional speakers would improve the modelling accuracy for all phone types. On theoretical grounds we expect that the means should in all cases approach 0.5, and this expectation is supported by the observed trends. Again we observe that the unvoiced plosives and fricatives quickly achieve high values for the bound (close to the ideal 0.5). Low inter-speaker variability for the phone /d/ is also confirmed with high bound values. The high inter-speaker variability of the fricative phones

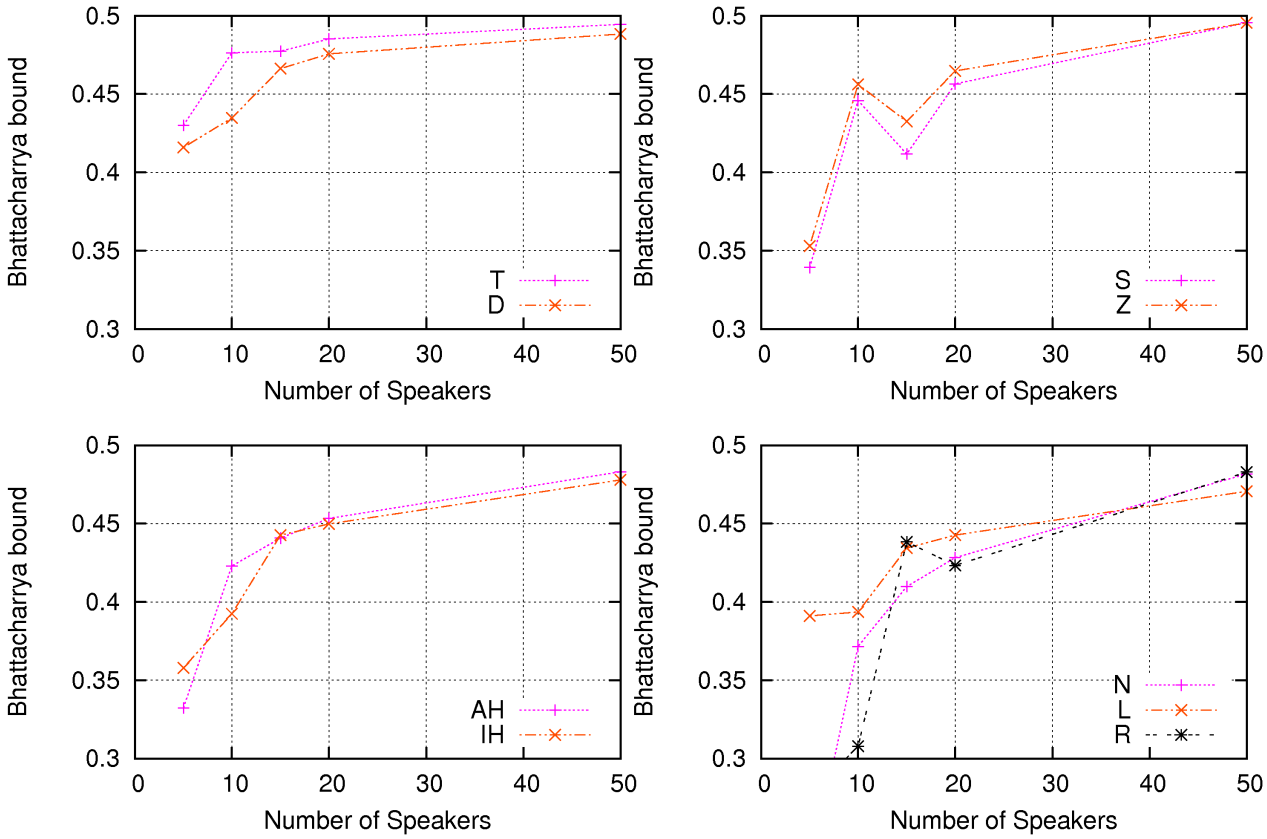


Figure 3: *Effect of number of speakers on mean of Bhattacharyya bound for different phone groups using 100 utterances per speaker*

are apparent in the unstable behaviour they exhibit (varying between 0.4 to 0.45 up to 20 speakers). Interestingly the vowels are not the slowest to reach large bound values as the speakers are increased: the phones /n/ (nasals) and /r/ (liquids) converge more slowly, signalling a higher inter-speaker variability for this group.

These results confirm the results obtained in Section 4.3 and comparative behaviour for the different phone types is summarised in Table 2.

Phone type	Inter-speaker variability	Intra-speaker variability
Unvoiced plosives	low	low
Voiced plosives	low	high
Unvoiced fricatives	medium	low
Voiced fricatives	medium	low
Vowels	medium	high
Nasals	high	medium
Liquids	high	medium

Table 2: *Comparative inter- and intra-speaker variability for different phone types.*

4.5. Effect of model complexity

The numerical values of the Bhattacharyya bound for different model types cannot be compared directly, since factors such as the existence of local minima during training increase the apparent variability of more complex models. We therefore compare such models by studying the observed bound values as a fraction of the observed asymptotic values. While the more complex model requires additional samples before the asymptote is reached, the same trends across phone groups are observed for more complex models. This is illustrated in Figure 4 where this fraction is shown, as the number of phone occurrences per speaker is increased. In these figures, data from 20 speakers is shown for both the simple single Gaussian model as well as the more complex 6-mixture GMM.

5. Conclusions

We have introduced a systematic approach that enables us to study the resource requirements for speech-recognition systems, based on the mean Bhattacharyya bound between models trained on different subsets of the data. We find that the different broad categories of phones have significantly different data requirements: whereas as few as 20 speakers and fewer than 50 samples per speaker are sufficient for the plosives /t/ and /d/, even 100 samples per speaker from each of 50 speakers do not describe the vowels, liquids or nasals adequately. Overall, the number of speakers for even a basic speaker-independent re-

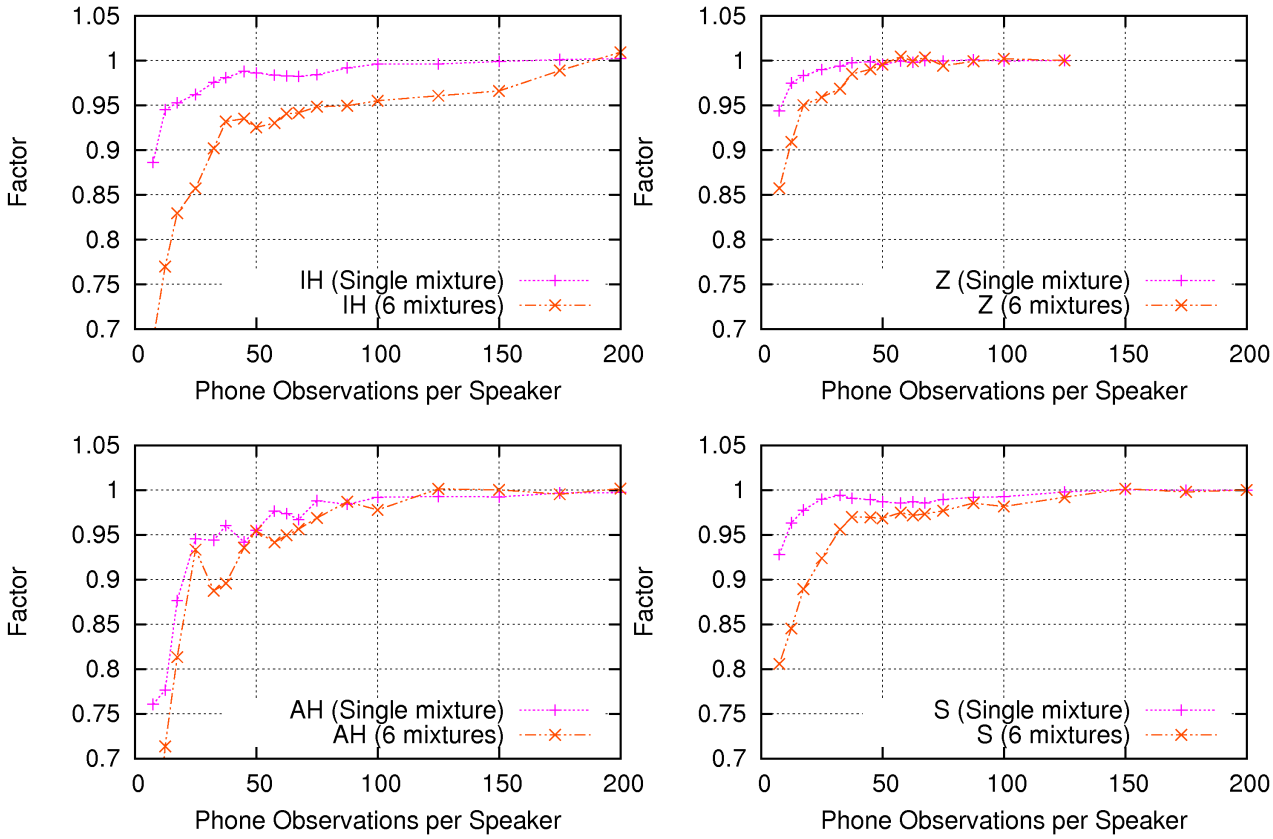


Figure 4: *Comparing the effect of model complexity on the relative distance to asymptote for two phone groups*

source collection therefore needs to contain significantly more than 50 speakers. (We are not able to suggest a reasonable lower bound based on the data used in this study.)

We found similar trends for simple and more complex models, with the more complex models requiring somewhat more speakers and phone occurrences to stabilise. Our work has focused on simple models, and can be extended in various directions. It would be interesting to see whether robust asymptotes are achieved as the number of speakers is increased; other variables, such as gender or speaking style should also be studied along with more complex models (e.g. context-specific models, multistate models such as HMMs and more complex density estimators). In our current work we are also investigating how the measurements described here relate to actual speech recognition performance obtained.

These insights are likely to play an increasingly important role as the reach of speech processing systems extends beyond the major languages of the world.

6. References

[1] "Lwazi ASR corpus," 2008, http://www.meraka.org.za/lwazi.

[2] A. Nagroski, L. Boves, and H. Steeneken, "In search of optimal data selection for training of automatic speech recognition systems," *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, pp. 67–72, 30 Nov. - 3 Des 2003.

[3] G. Riccardi and D. Hakkani-Tur, "Active and unsupervised learning for automatic speech recognition," in *Proceedings of EUROSPEECH*, Geneva, Switzerland, 2003.

[4] Y. Wu, R. Zhang, and A. Rudnicky, "Data selection for speech recognition," *Automatic Speech Recognition and Understanding, 2007. ASRU. IEEE Workshop on*, pp. 562–565, 9 - 13 Des 2007.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc., 2nd edition, 1990.

[6] P.A. Olsen and J.R. Hershey, "Bhattacharyya error and divergence using variational importance sampling," in *Proceedings of Interspeech*, Antwerp, Belgium, 2007, pp. 46–49.

Acoustic analysis of diphthongs in Standard South African English

Olga Martirosian [1] and Marelie Davel [2]

[1] School of Electrical, Electronic and Computer Engineering,
North-West University, Potchefstroom, South Africa /
[2] Human Language Technologies Research Group, Meraka Institute, CSIR
omartirosian@csir.co.za, mdavel@csir.co.za

Abstract

Diphthongs typically form an integral part of the phone sets used in English ASR systems. Because diphthongs can be represented using smaller units (that are already part of the vowel system) this representation may be inefficient. We evaluate the need for diphthongs in a Standard South African English (SSAE) ASR system by replacing them with selected variants and analysing the system results. We define a systematic process to identify and evaluate replacement options for diphthongs and find that removing all diphthongs completely does not have a significant detrimental effect on the performance of the ASR system, even though the size of the phone set is reduced significantly. These results provide linguistic insights into the pronunciation of diphthongs in SSAE and simplifies further analysis of the acoustic properties of an SSAE ASR system.

1. Introduction

The pronunciation of a particular phoneme is influenced by various factors, including the anatomy of the speakers, whether they have speech impediments or disabilities, how they need to accommodate their listener, their accent, the dialect they are using, their mother tongue, the level of formality of their speech, the amount and importance of the information they are conveying, their environment (Lombard effect) and even their emotional state [1].

The nativity of a person's speech describes the combination of the effects of their mother tongue, the dialect that they are speaking, their accent and their proficiency in the language that they are speaking. If an automatic speech recognition (ASR) system uses speech and a lexicon associated with a certain nativity, non-native speech causes consistently poor system performance [2]. For every different dialect of a language, additional speech recordings are typically required, and lexicon adjustments may also be necessary.

Standard South African English (SSAE) is an English dialect which is influenced by three main South African English (SAE) variants: White SAE, Black SAE, Indian SAE and Cape Flats English. These names are ethnically motivated, but because each ethnicity is significantly related to a specific variant of SAE, they are seen as accurately descriptive [3]. Each variety will be made up of South African English as influenced specifically by the different languages and dialects thereof spoken in South Africa. It should be noted that these variants include extreme, strongly accented English variants that are not included in SSAE, and not referred to in this paper.

This analysis focuses on the use of diphthongs in SSAE. This is an interesting and challenging starting point to an acoustic analysis of SSAE. We are specifically interested in diphthongs since some of these sounds (such as /OY/ and /UA/, using ARPABET notation) are fairly rare and large corpora are required to include sufficient samples of these sounds.

A diphthong is a sound that begins with one vowel and ends with another. Because the transition between the vowels is smooth, it is modelled as a single phoneme. However, since it would also have been possible to construct a diphthong using smaller units that are already part of the vowel system, this may be an inefficient representation.

In this paper we evaluate the need for diphthongs in a lexicon by systematically replacing them with selected variants and analysing the system results. One way to analyse the phonemic variations in a speech corpus is to use an ASR system [4]. A detailed error analysis can be used to identify possible phonemic variations [1]. Once possible variations are identified, they can be filtered using forced alignment [4].

Some studies have found that using multiple pronunciations in a lexicon is better for system performance [5], while others have found that a single pronunciation lexicon outperforms a multiple pronunciation lexicon [6]. The argument can therefore be made for representing the frequent pronunciations in the data, but being careful not to over-customise the dictionary - if acoustic models are trained on transcriptions that are too accurate, they do not develop robustness to variation and therefore contribute to a decline in the recognition performance of the system [7].

In this paper we analyse diphthong necessity systematically in the context of an SSAE ASR system. The paper is structured as follows: In Section 2 we describe a general approach to identify possible replacement options for a specific diphthong, and to evaluate the effect of such replacement. In Section 3 we first perform a systematic analysis of four frequently occurring diphthongs individually, before replacing all diphthongs in a single experiment and reporting on results. Section 4 summarises our conclusions.

2. Approach

In this section we describe a general approach to first suggest alternatives for a specific diphthong and then to evaluate the effectiveness of these alternatives.

2.1. Automatic suggestion of variants

In order to identify possible alternatives (or variants) for a single diphthong, we propose the following process:

1. An ASR system is trained as described in more detail in Section 3.1.3. The system is trained using all the data available and a default dictionary containing the original diphthongs.

2. The default dictionary is expanded: variant pronunciations are added to words containing the diphthong in question by replacing the diphthong with all vowels and combinations of two vowels. Two glides (the sounds */W/* and */Y/*) are considered as part of the vowel set for the purpose of this experiment.
3. The original diphthong is removed completely, so that the dictionary only contains possible substitutions. The order of the substitutions is randomised in every word. This ensures that the speech that would represent the diphthong is not consistently labelled as one of the possible substitutions and the training process therefore biased in a certain direction.
4. The ASR system is used to force align the data using the options provided by the new dictionary. (Since the diphthong has been removed, the system now has to select the best of the alternatives that remain.)
5. The forced alignment using the expanded dictionary (alignment B) is compared to the forced alignment using the default dictionary (alignment A):
    - Each time the diphthong in question is found in alignment A, it and its surrounding phonemes are compared to the phonemes recognised at the same time interval in alignment B. The phonemes in alignment B that align with the diphthong in alignment A are noted as possible alternatives to the specific diphthong.
    - The alternatives are counted and sorted by order of frequency.
6. The frequency sorted list is perused and three to five possible replacements for the diphthong are selected by a human verifier from the top candidates. The human verifier is required to assist the system because they are equipped with SSAE and general linguistic knowledge, and are thus able to select replacement candidates that contain vowels or vowel combinations that are most likely to be replacements for the diphthong in question.

Once this process is completed, a list of possible replacements is produced. This list is based on a combination of system suggestion and human selection. For example, as a diphthong typically consists of two or more vowels linked together, it is quite likely that the best alternative to a diphthong is a combination of two vowels (diphone). Even though an ASR system may not initially lean towards such a double vowel replacement, including such an alternative may be forced by the human verifier. Also, knowledge-based linguistically motivated choices may be introduced at this stage. These choices are motivated by linguistic definitions of diphthongs as well as SAE variant definitions supplied in [3]. This process is described in more detail when discussing the process with regard to specific diphthongs below.

2.2. Evaluating replacement options

Once a list of three to five possible replacements has been selected for each diphthong, these replacements can be evaluated for their ability to replace the diphthong in question. Per diphthong, the following process is followed:

1. The default dictionary is expanded to include the selected alternatives as variants for the diphthong in question. The pronunciation with the diphthong is removed and the alternative pronunciations are randomised in order not to bias the system towards one pronunciation (as again, the system initially trains on the first occurring pronunciation of every word).
2. Each time the diphthong is replaced by an alternative, a list is kept of all words and pronunciations added.
3. An ASR system is trained on all the data using the expanded dictionary, and the alignments produced during training are analysed.
4. The pronunciations in the forced alignment are compared to each of the lists of added alternatives in turn, calculating the number of times the predicted pronunciation is used in the forced alignment, resulting in an occurrence percentage for each possible replacement.
5. Using these occurrence percentages, the top performing alternatives are selected. The number of selections is not specified, but rather, the ratio between the occurrence percentages of the alternatives is used to select the most appropriate candidates for the next round.
6. This process is repeated until only a single alternative remains, or no significant distinction can be made between two alternatives.
7. After each iteration of this process, the ASR phoneme and word accuracies are monitored.

3. Experimental Results

3.1. The baseline ASR system

In this section we define the baseline ASR system used in our experiments. We describe the dictionary used, the speech corpus and provide details with regard to system implementation.

3.1.1. Pronunciation Dictionary

The pronunciation dictionary consists of a combination of the British English Example Dictionary (BEEP) [8] and a supplementary pronunciation dictionary that has words contained in the speech corpus but not transcribed in BEEP. (This includes SAE specific words and names of places). The 44-phoneme BEEP ARPABET set is used. The dictionary was put through a verification process [9] but also manually verified to eliminate highly irregular pronunciations. The dictionary has 1 500 entries, 1 319 of which are unique words. The average number of pronunciations per word is 1.14 and the number of words with more than one pronunciation is 181. In further experimentation, this dictionary is referred to as the *default dictionary*.

3.1.2. Speech Corpus

The speech corpus consists of speech recorded using existing interactive voice response systems. The recordings consist of single words and short sentences. There are 19 259 recordings made from 7 329 telephone calls, each of which is expected to contain a different speaker. The sampling rate is 8 kHz and the total length of the calls is 9 hours and 2 minutes. It total, 1319 words are present in the corpus, but the corpus is rather specialised, with the top 20% of words making up over 90% of the corpus. For cross validation of the data, all the utterances of a single speaker were grouped in either the training or the test data, and not allowed to appear in both. The relevant phoneme counts are given in Table 1.

Table 1: *Selected phoneme counts for the speech corpus. Counts are calculated using forced alignment with the speech corpus and default dictionary. Diphthongs are shown in bold.*

Phoneme	Occurrences	Phoneme	Occurrences
/AX/	14 282	/UW/	3 151
/IY/	9 634	/AO/	3 106
/IH/	9 084	/Y/	2 743
/AY/	**6 561**	**/EA/**	**2 566**
/EH/	6 158	/ER/	2 499
/AE/	5 470	/AA/	2 097
/EY/	**4 509**	**/AW/**	**2 037**
/W/	4 293	/UH/	1 324
/AH/	3 883	**/IA/**	**1 014**
/OW/	**3 442**	**/UA/**	**455**
/OH/	3 232	**/OY/**	**39**

3.1.3. System Particulars

A fairly standard ASR implementation is used: context dependent triphone acoustic models, trained using Cepstral Mean Normalised 39-dimensional MFCCs. The optimal number of Gaussian Mixtures per state in the acoustic models was experimentally determined to be 8. The system makes use of a flat word based language model and was optimised to achieve a baseline phoneme accuracy of 79.57% and a corresponding word accuracy of 64.50%. As a measure of statistical significance, the standard deviation of the mean is calculated across the 10 cross-validations, resulting in 0.07% and 0.13% for phoneme and word accuracy respectively. The system was implemented using the ASR-Builder software [10].

3.2. Systematic replacement of individual diphthongs

In this section we provide results when analysing a number of diphthongs individually according to the process described in the previous section (Section 2).

Since training the full system outlined in Section 3.1.3 is highly time consuming, a first experiment was performed to determine whether a monophone-based system is sufficient to use during the process to identify and evaluate replacement options. For each diphthong investigated, a dictionary was compiled as described in Section 2.1, a full system was trained using this dictionary, and its forced alignment output when using monophone models was compared with its forced alignment output when using triphone models with 8 mixtures. This comparison always resulted in an equivalence of more than 95%. Therefore, from here onwards, only monophone alignment is used for decision making, while final accuracies, or selection rates, are reported on using the full triphone system.

3.2.1. Diphthong Analysis: /AY/

The AY diphthong was first to be analysed. The results of the analysis are summarised in Table 2. Each line represents one experiment. For each experiment, the accuracies of each of the included alternatives are noted, as well as the cross validated phoneme and word accuracies of the full ASR system.

The progression of this experiment is outlined below:

- In the first iteration, the alternatives /AH/, /AH IH/ and /AA/ achieve the highest accuracies and are selected for the next round. /AH/ achieves the highest selection rate overall.
- In the second iteration, the alternatives /AH/ and /AA/ achieve the highest accuracies and are selected for the next round. Again, /AH/ has the highest selection rate. All diphones have now been eliminated.
- In the third iteration, /AH/ has the highest selection rate and is therefore selected as the final and best alternative for /AY/.
- In the fourth iteration, /AH/ is tested as a replacement of /AY/. Phoneme accuracy rises to its highest, however, word accuracy suffers. As phoneme accuracy in influenced by the change in number of phonemes (from one experiment to another), word accuracy is the more reliable measure for this experiment.
- The diphone theory, detailed in Section 2.1, suggests that, because diphthongs are made up of two sounds, their replacement must also consist of two sounds in order to have the capacity to model them accurately. In order to test this theory, an iteration is run with /AH/ and /AH IH/ as the alternatives for /AY/. The ASR system still selects the /AH/ alternative over the /AH IH/ alternative. However, the word accuracy increases at this iteration, implying that perhaps having /AH IH/ as an alternative pronunciation for /AY/ fits the acoustic data better than only having /AH/.
- A final iteration is run with the knowledge-based linguistically motivated choice "/AH IH/" as the replacement of /AY/. Both the phoneme and word accuracy rise to their highest values with this replacement. This shows that the linguistically predicted /AH IH/ is indeed the best replacement for /AY/.

3.2.2. Diphthong Analysis: /EY/

The /EY/ diphthong is analysed using the technique outlined in Section 2. The results are summarised in Table 3. In the first iteration, /AE/ and /EH/ are clearly the better candidates, but the diphone (double vowel) scores were lower and very similar. Thus, for the second iteration, all diphones are cut and only /AE/ and /EH/ are tested. But for the third iteration, testing the necessity of including a diphone, two of the diphones were brought back to be tested again. It should be noted that the highest word accuracy achieved for the suggested variants was achieved in the third iteration, suggesting that diphones are indeed necessary when attempting to replace a diphthong. Again, the highest accuracy achieved overall is for the knowledge-based linguistically suggested alternative /EH IH/.

3.2.3. Diphthong Analysis: /EA/

The /EA/ diphthong is now analysed. The results of the experiment are summarised in Table 5. These results behave quite differently compared to the other diphthong experiments. The first iteration, where all 3 of the variant options are included, achieves the highest word accuracy, even higher than the iteration which makes use of linguistic knowledge. The phoneme accuracy however, increases with every iteration, reaching its peak with the use of the linguistic replacement. Again, this may be related to the change in number of phones (in words causing errors) which makes word accuracy a more reliable measure. The knowledge-based linguistic replacement performs very well, achieving the second highest word accuracy overall.

Table 2: *Results of the experiments for the diphthong /AY/*

	/AH/	/AA/	/AH IH/	/AE IY/	/AH IY/	P Acc	W Acc
1	0.46	0.20	0.18	0.08	0.07	78.51%	63.88%
2	0.46	0.36	0.17	N/A	N/A	78.75%	64.06%
3	0.56	0.43	N/A	N/A	N/A	79.14%	64.17%
4	1	N/A	N/A	N/A	N/A	79.56%	64.03%
5	0.62	N/A	0.38	N/A	N/A	79.19%	64.13%
6	N/A	N/A	1	N/A	N/A	79.77%	64.30%

Table 3: *Results of the experiments for the diphthong /EY/*

	/AE/	/EH/	/AE IY/	/AE IH/	/EH IY/	/EH IH/	P Acc	W Acc
1	0.24	0.25	0.17	0.17	0.16	N/A	78.97%	64.27%
2	0.59	0.41	N/A	N/A	N/A	N/A	79.30%	64.03%
3	0.48	N/A	0.26	0.27	N/A	N/A	79.36%	64.41%
4	1	N/A	N/A	N/A	N/A	N/A	79.64%	64.04%
5	N/A	N/A	N/A	N/A	N/A	1	79.78%	64.43%

Table 4: *Results of the experiments for the diphthong /OW/*

	/OH/	/ER/	/ER UW/	/AE/	/AE UW/	/AX UH/	P Acc	W Acc
1	0.29	0.36	0.14	0.13	0.08	N/A	79.53%	64.33%
2	0.52	0.48	N/A	N/A	N/A	N/A	79.57%	64.41%
3	0.59	N/A	0.41	N/A	N/A	N/A	79.53%	64.48%
4	1	N/A	N/A	N/A	N/A	N/A	79.60%	64.45%
5	N/A	N/A	N/A	N/A	N/A	1	79.63%	64.48%

Table 5: *Results of the experiments for the diphthong /EA/*

	/EH/	/IH EH/	/AE/	/EH AX/	P Acc	W Acc
1	0.51	0.34	0.15	N/A	79.22%	64.49%
2	0.72	0.28	N/A	N/A	79.51%	64.43%
3	1	N/A	N/A	N/A	79.65%	64.21%
4	N/A	N/A	N/A	1	79.73%	64.30%

Table 6: *IPA based diphthong replacements*

Diphthong	Diphone	Diphthong	Diphone
/AY/	/AH IH/	**/OY/**	/OH IH/
/EY/	/EH IH/	**/AW/**	/AH UH/
/EA/	/EH AX/	**/IA/**	/IH AX/
/OW/	/AX UH/	**/UA/**	/UH AX/

3.2.4. Diphthong Analysis: /OW/

The experiment is repeated for the diphthong /OW/. The results for the experiment are outlined in Table 4. The phoneme accuracy follows a similar pattern to the earlier experiments. The word accuracy is highest at both iteration 3, where a diphone is included and iteration 5, where the linguistic knowledge-based replacement is implemented. The knowledge-based linguistic replacement once again achieves the highest phoneme and word accuracies.

3.3. Systematic replacement of all diphthongs

Given the results achieved in the earlier experiments, a final experiment is run where all the diphthongs are replaced using a systematic system based on the linguistic definitions of the individual diphthongs.

Two ASR systems are used, designed as described in Section 3.1.3. These two systems differ only with regard to their dictionary. One system (system A) uses the baseline dictionary, in the other (system B), the diphthongs in the baseline dictionary are all replaced with their diphone definitions, using British English definitions defined in Table 6.

All results are cross-validated and the two systems are compared using their word accuracies. Interestingly word accuracy decreases only very slightly: from 64.53% for system A to 64.35% for system B. The removal of 8 diphthongs is therefore not harmful to the accuracy of the system. This is an interesting result, especially as the detailed analysis was only performed for 4 of the diphthongs and further optimisation may be possible.

4. Discussion

The aim of this study was to gain insight into the use of diphthongs in SSAE. We defined a data-driven process through which diphthongs could automatically be replaced with optimal phonemes or phoneme combinations. To complement this process, a knowledge-based experiment was set up using linguistic data for British English. Although the data-driven method was partially successful in finding the best replacement for diphthongs, the knowledge-based method was superior. However, the increase in accuracy from the knowledge-based method is small enough that if knowledge is not available, the data-driven technique can be used quite effectively.

It is interesting to consider the South African English variants that are described in [3]. The variants described here or ones close to them always appear on the list of the top candidates of the data-driven selection. This in itself is an interesting observation from a linguistic perspective.

From a linguistic perspective, the fact that a diphthong can successfully be modelled as separate phonemes provides an insight into SSAE pronunciation.

From a technical perspective, the removal of diphthongs simplifies further analysis of SSAE vowels. Our initial investigations were complicated by the confusability between diphthongs and vowel pairs, and this effect can now be circumvented without compromising the precision of the results.

Ongoing research includes further analysis of SSAE phonemes with the aim to craft a pronunciation lexicon better suited to South African English (in comparison with the British or American versions commonly available). In addition, similar techniques will be used to evaluate the importance of other types of phonemes, for example the large number of affricates in Bantu language.

5. References

[1] Strik H. and Cucchiarini C., "Modeling pronunciation variation in ASR: A survey of the literature," *Speech Communication*, vol. 29, pp. 225–246, 1999.

[2] Wang Z., Schultz T., and Waibel A., "Comparison of acoustic model adaptation techniques of non-native speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003, vol. 1, pp. 540 – 543.

[3] Kortmann B. and Schneider E.W., *A Handbook of Varieties of English*, vol. 1, Mouton de Gruyter New York, 2004.

[4] Adda-Decker M. and Lamel L., "Pronunciation variants across system configuration, language and speaking style," *Speech Communication*, vol. 29, pp. 83–98, 1999.

[5] Wester M., Kessens J.M., and Strik H., "Improving the performance of a dutch csr by modelling pronunciation variation," in *Proceedings of the Workshop Modeling Pronunciation Variation for Automatic Speech Recognition*, Rolduc, The Netherlands, May 1998, pp. 145–150.

[6] Hain T., "Implicit modelling of pronunciation variation in automatic speech recognition," *Speech communication*, vol. 46, no. 2, pp. 171–188, 2005.

[7] Saraclar M., Nock H., and Khudanpur S., "Pronunciation modeling by sharing gaussian densities across phonetic models," in *Sixth European Conference on Speech Communication and Technology*, Budapest, Hungary, September 1999, ISCA.

[8] BEEP, "The british english example pronunciation (beep) dictionary," `ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries`.

[9] Martirosian O.M. and Davel M., "Error analysis of a public domain pronunciation dictionary," in *PRASA 2007: Eighteenth Annual Symposium of the Pattern Recognition Association of South Africa*, Pietermaritzburg, South Africa, November 2007, pp. 13–18.

[10] Mark Zsilavecz, "ASR-Builder," January 2008, `http://sourceforge.net/projects/asr-builder`.

The origin of Afrikaans pronunciation: a comparison to west Germanic languages and Dutch dialects

Wilbert Heeringa, Febe de Wet

Meertens Institute, Variationist Linguistics
wilbert.heeringa@meertens.knaw.nl
Stellenbosch University, Centre for Language and Speech Technology
fdw@sun.ac.za

Abstract

This paper aims to find the origin of the Afrikaans pronunciation with the use of dialectometry. First, Afrikaans was compared to Standard Dutch, Standard Frisian and Standard German. Pronunciation distances were measured by means of Levenshtein distances. Afrikaans was found to be closest to Standard Dutch. Second, the Afrikaans pronunciation was compared to 361 Dutch dialect varieties in the Netherlands and North-Belgium. Material from the *Reeks Nederlandse Dialectatlassen* was used. Afrikaans was found to be closest to the South Holland variety of Zoetermeer, which largely agrees with Kloeke (1950, *Herkomst en Groei van het Afrikaans*).

1. Introduction

Afrikaans is a daughter language of Dutch and is mainly spoken in South Africa and Namibia. Reenen & Coetzee [1] briefly describe the origin of Afrikaans. Nearly 350 years ago, in 1652, Jan van Riebeeck founded a refreshment station at the Cape of Good Hope on the way to the Indies and introduced a Dutch variety. He and the group around him came from the southern part of the Dutch province of South-Holland. Van Reenen & Coetzee refer to Kloeke [2] who claims that Jan van Riebeeck's group is the most important source of today's Afrikaans language. Kloeke writes extensively about the origin of Afrikaans in his *Herkomst en Groei van het Afrikaans* 'Origin and growth of Afrikaans'. Van Reenen & Coetzee also refer to Scholtz [3, p. 254] who does not agree with Kloeke but wonders whether Afrikaans is derived from a common Hollandish language, the Hollandish norm of the second half of the 17th century. However, Van Reenen & Coetzee doubt whether a common Hollandish language already existed in that period.

The South African constitution recognizes 11 official languages. According to the 2001 census data, Zulu is the most widely spoken mother-tongue in South Africa, followed by Xhosa and Afrikaans, with the latter constituting 13.3% of the population. This percentage is lower than the value reported in the 1996 census, when 14.4% of the population indicated that Afrikaans was their first language [4]. This observation can probably be explained by a decline in population growth as well as the fact that many Afrikaans people emigrated during that period. Although English is most often used as the lingua franca in the country, Afrikaans is more frequently used than English in some provinces of South Africa and Namibia.

As explained above, Afrikaans is seen historically as a daughter of Dutch. This paper shows that Afrikaans is linguistically still a daughter of Dutch. In order to prove this, the Afrikaans pronunciation is compared to the pronunciation of the languages in the west Germanic language group: Standard Dutch, Standard Frisian and Standard German. Pronunciation distances are measured with Levenshtein distance, a string edit distance measure. Kessler [5] was the first to use Levenshtein distance for measuring linguistic distances. He applied Levenshtein distance to transcriptions of Irish Gaelic dialect varieties. Later Levenshtein distances was applied to Dutch dialects by Nerbonne et al. [6] (more detailed results are given by Heeringa [7], to Norwegian by Gooskens & Heeringa [8] and to several other dialect families.

The Levenshtein distance corresponds to the distance between the transcriptions of two pronunciations of the same concept corresponding to two different varieties. The distance is equal to the minimum number of insertions, deletions and substitutions of phonetic segments needed to transform one transcription into another. The distance between two varieties is based on several pronunciation pairs, in our case 125. The corresponding Levenshtein distances are averaged. This paper aims to answer the following question: which of these standard languages is closest to Afrikaans? Afrikaans is also compared to 361 Dutch varieties, found in the Dutch dialect area. This area comprises the Netherlands and North-Belgium. Material from the *Reeks Nederlandse Dialectatlassen* is used. We determine which dialect variety (or dialect region) is closest to Afrikaans. Again pronunciation differences are measured with Levenshtein distance. We also distinguish between vowel and consonant differences.

The aim of this study is twofold. Firstly, this investigation sheds light on the linguistic relationship between Afrikaans and the west Germanic languages, and between Afrikaans and the Dutch dialects in particular. Secondly, the results of this study will provide useful guidelines for the development of speech technology applications for Afrikaans. Human language technology (HLT) is still a relatively new field in South Africa and most of the South African languages are severely underresourced in terms of the data and software required to develop HLT applications such as automatic speech recognition engines, speech synthesis systems, etc. Development can be accelerated if existing resources from closely related languages can be used. We are specifically interested in constructing a large vocabulary continuous speech recognition system for Afrikaans. This requires large quantities of annotated audio data. Given that very little Afrikaans data is currently available, we would like to investigate the possibility of using data from closely related languages.

2. Data source

2.1. Dutch dialects

In order to study the relationship between Afrikaans and Dutch dialect varieties, it would be preferable to use data from about 1652, because that time period would coincide with Jan van Riebeeck's influence on the Afrikaans language. Of course, we do not have phonetic transcriptions from that time. The oldest available source containing phonetic transcriptions of a dense sample of dialect locations is the *Reeks Nederlandse Dialectatlassen* (RND), a series of Dutch dialect atlases which were edited by Blancquaert and Pée [9] in the period 1925–1982. The atlases cover the Dutch dialect area, which comprises the Netherlands, the northern part of Belgium, a smaller northwestern part of France and the German county Bentheim.

In the RND, the same 141 sentences are translated and transcribed in phonetic script for each dialect. Blancquaert mentions that the questionnaire was conceived as a range of sentences with words that illustrate particular sounds. The design saw to it that, for example, possible changes of old-Germanic vowels, diphthongs and consonants are represented in the questionnaire. Since digitizing the phonetic texts is time-consuming and the material was intended to be processed by the word-based Levenshtein distance, a set of only 125 words was selected from the text (Heeringa [10]). The words are selected more or less randomly and may be considered as a random sample. The transcriptions of the 125 word pronunciations were digitized for each dialect. The words represent (nearly) all vowels (monophthongs and diphthongs) and consonants. The consonant combination [sx] is also represented, which is pronounced as [sk] in some dialects and as [ʃ] in some other dialects.

The RND contains transcriptions of 1956 Dutch varieties. Since it would be very time-consuming to digitize all transcriptions, a selection of 361 dialects has been made (see Heeringa [10]). When selecting the dialects, the goal was to get a net of evenly scattered dialect locations. A denser sampling resulted in the areas of Friesland and Groningen, and in the area in and around Bentheim. In Friesland the town Frisian dialect islands were added to the set of varieties which belong to the (rural) Frisian dialect continuum. In Groningen, some extra localities were added because of personal interest. In the area in and around Bentheim extra varieties were added because of a detailed investigation in which the relationship among dialects at both sides of the border was studied. Besides the relationship to Standard Dutch and Standard German was studied (see Heeringa et al. [10]).

In the RND, the transcriptions are noted in some predecessor of IPA. The transcriptions were digitized using a computer phonetic alphabet which might be considered as a dialect of X-SAMPA. The data is freely available at `http://www.let.rug.nl/~heeringa/dialectology/atlas/rnd/`.

2.2. Languages

In this paper, Dutch dialects are compared to Afrikaans. The 125 words, selected from the RND sentences, were therefore translated into Afrikaans and pronounced by an old male and a young female, both native speakers of Afrikaans. Old males are known to be conservative speakers while young females are usually innovative speakers [11]. In our measurements below we always take the average of the two speakers when we compare Dutch dialects to Afrikaans. The pronunciations of the two speakers were transcribed consistently with the RND transcriptions.

Afrikaans is also compared to Standard Dutch, Standard Frisian and Standard German. To ensure consistency with the existing RND transcriptions, the Standard Dutch transcription is based on the *Tekstboekje* of Blancquaert [12]. However, words such as *komen*, *rozen* and *open* are transcribed as [koˑməˑ], [roːzəˑ] and [oˑpəˑ]. In the *Tekstboekje* of Blancquaert these words would end on an [n], as suggested by the spelling. For more details see Heeringa [10].

The RND transcription of the Frisian variety of Grouw is used as Standard Frisian. Standard Frisian is known to be close to the variety of Grouw.

The Standard German word transcriptions are based on *Wörterbuch der deutschen Aussprache* [13]. However, the transcriptions were adapted so that they are consistent with the RND data. In the dictionary the <r> is always noted as [r], never as [ʁ]. Because in German both realizations are allowed, for each pronunciation containing one or more <r>'s two variants are noted, one in which the [r] is pronounced, and another in which the [ʁ] is pronounced. More details are given by Heeringa *et al.* [14]. In the measurements below, both realizations will be taken into account.

3. Measuring pronunciation distances

Pronunciation differences are measured with Levenshtein distance. Pronunciation variation includes variation in sound components and morphology. The items to be compared should have the same meaning and they should be cognates.

3.1. Algorithm

Using the Levenshtein distance, two varieties are compared by measuring the pronunciation of words in the first variety against the pronunciation of the same words in the second [15]. We determine how one pronunciation might be transformed into the other by inserting, deleting or substituting sounds. In this way *distances* between the transcriptions of the pronunciations are calculated. Weights are assigned to these three operations. In the simplest form of the algorithm, all operations have the same cost, e.g., 1. Assume the Standard Dutch word *hart* 'heart' is pronounced as [hɑrt] in Afrikaans and as [ærtə] in the East Flemish dialect of Nazareth (Belgium). Changing one pronunciation into the other can be done as follows:

hɑrt	delete h	1
ɑrt	replace ɑ by æ	1
ært	insert ə	1
ærtə		
		3

In fact many string operations map [hɑrt] to [ærtə]. The power of the Levenshtein algorithm is that it always finds the least costly mapping. To deal with syllabification in words, the Levenshtein algorithm is adapted so that only a vowel may match with a vowel, a consonant with a consonant, the [j] or [w] with a vowel (and vice versa), the [i] or [u] with a consonant (and vice versa), and a central vowel (in our research only the schwa) with a sonorant (and vice versa). In this way unlikely matches (e.g. a [p] with an [a]) are prevented. The longest alignment has the greatest number of matches. In our example we thus have the following alignment:

h	ɑ	r	t	
	æ	r	t	ə
1	1			1

3.2. Operations weights

The simplest versions of this method are based on a notion of phonetic distance in which phonetic overlap is binary: non-identical phones contribute to phonetic distance, identical ones do not. Thus the pair [i,ɒ] counts as different to the same degree as [i,ɪ]. The version of the Levenshtein algorithm used in this paper is based on the comparison of spectrograms of the sounds. Since a spectrogram is the visual representation of the acoustical signal, the visual differences between the spectrograms are reflections of the acoustical differences. The spectrograms were made on the basis of recordings of the sounds of the International Phonetic Alphabet as pronounced by John Wells and Jill House on the cassette *The Sounds of the International Phonetic Alphabet* from 1995 [16]. The different sounds were isolated from the recordings and monotonized at the mean pitch of each of the two speakers with the program PRAAT [17]. Next, for each sound a spectrogram was made with PRAAT using the so-called Barkfilter, a perceptually oriented model. On the basis of the Barkfilter representation, segment distances were calculated. Inserted or deleted segments are compared to silence, and silence is represented as a spectrogram in which all intensities of all frequencies are equal to 0. The [ʔ] was found closest to silence and the [a] was found most distant. This approach is described extensively in Heeringa [7, pp. 79–119]. In perception, small differences in pronunciation may play a relatively strong role in comparison to larger differences. Therefore logarithmic segment distances are used. The effect of using logarithmic distances is that small distances are weighted relatively more heavily than large distances. The weights will vary between 0 and 1. In a validation study, Heeringa [7, pp. 178–195] found that among several alternative distances obtained with the Levenshtein distance measure, using logarithmic Bark filter segment distances gives results which most closely approximates dialect distances as perceived by the speakers themselves.

3.3. Vowels and consonants

Besides calculating Levenshtein distances on the basis of all segments (full pronunciation distance) we also calculated distances on the basis of only vowel and consonant substitutions. If distances are calculated solely on the basis of vowels, initially the full phonetic strings are compared to each other using Levenshtein distance. Once the optimal alignment is found, the distances are based on the alignment slots which represent vowel substitutions. Consonant substitutions are calculated mutatis mutandis.

3.4. Processing RND data

The RND transcribers use slightly different notations. In order to minimize the effect of these differences, we normalized their data. The consistency problems and the way we solved them are extensively discussed by Heeringa [10][7]. For the same reason only a part of the diacritics found in the RND is used.

As in earlier studies, we processed diacritics for length (extra short, half long, long), syllabicity (syllabic), voice (voiced, voiceless) and nasality (nasal) (see Heeringa [7, pp. 109–111]). In this study the diacritic for rounding (rounded, partly rounded, unrounded, partly unrounded) is used. The distance between for example [a] and rounded [i] is calculated as the distance between [a] and [y]. The distance between [a] and partly rounded [i] is equal to the average of the distance between [a] and [i] and the distance between [a] and [y]. The diacritic for rounding is important in our analysis since the [ɯ] and [ʏ] are not included in the phonetic transcription system of the RND, but transcribed as unrounded [u] and [o] respectively.

	Afrikaans	Dutch	Frisian	German
Afrikaans		3.2	4.1	5.1
Dutch			3.8	4.2
Frisian				4.8
German				

Table 1: Average Levenshtein distances between four standard languages

The distance between a monophthong and a diphthong is calculated as the mean of the distance between the monophthong and the first element of the diphthong and the distance between the monophthong and the second element of the diphthong. The distance between two diphthongs is calculated as the mean of the distance between the first elements and the distance between the second elements. Details are given by Heeringa [7, p. 108].

4. Results

4.1. Afrikaans versus Dutch, Frisian and German

The Levenshtein distance enables us to compare Afrikaans to other language varieties. Since we selected 125 words, the distance between a variety and Afrikaans is equal to the average of the distances of 125 word pairs. In Table 1 the average Levenshtein distances between Standard Afrikaans, Standard Dutch, Standard Frisian and Standard German are given. The distances represent the average Levenshtein distances, regardless of the length of the alignments the distances are based on. The table shows that Afrikaans is most closely related to Standard Dutch. This confirms that Afrikaans is a daughter of Dutch, as suggested by Kloeke[2], Van Reenen[1] and others. Furthermore, we found Afrikaans closer to Standard Frisian than to Standard German.

4.2. Afrikaans versus Dutch dialects

With the use of Levenshtein pronunciation distances between Afrikaans and 361 Dutch dialect varieties are calculated. The results are shown in Figure 1. In the map the varieties are represented by polygons, geographic dialect islands are represented by colored dots, and linguistic dialect islands are represented by diamonds. Lighter polygons, dots or diamonds represent dialects which are close to Afrikaans and darker ones represent the varieties which are more distant. The distances in the legend represent the average Levenshtein distances.

The closest varieties are found in the province of South-Holland. Some close varieties are also found in the provinces of North-Holland and Utrecht. The dialect variety of Zoetermeer is closest to Afrikaans. Kloeke[2] claimed that the dialect of the first settlers was the main source of Afrikaans. These settlers came from southern part of the Dutch province of South-Holland, the area around Rotterdam and Schiedam. Zoetermeer is slightly north of these two locations. The Limburg variety of Raeren is furthest away.

4.2.1. Vowels

Distances between Dutch dialects and Afrikaans based solely on vowel substitutions are shown in Figure 2. The map is

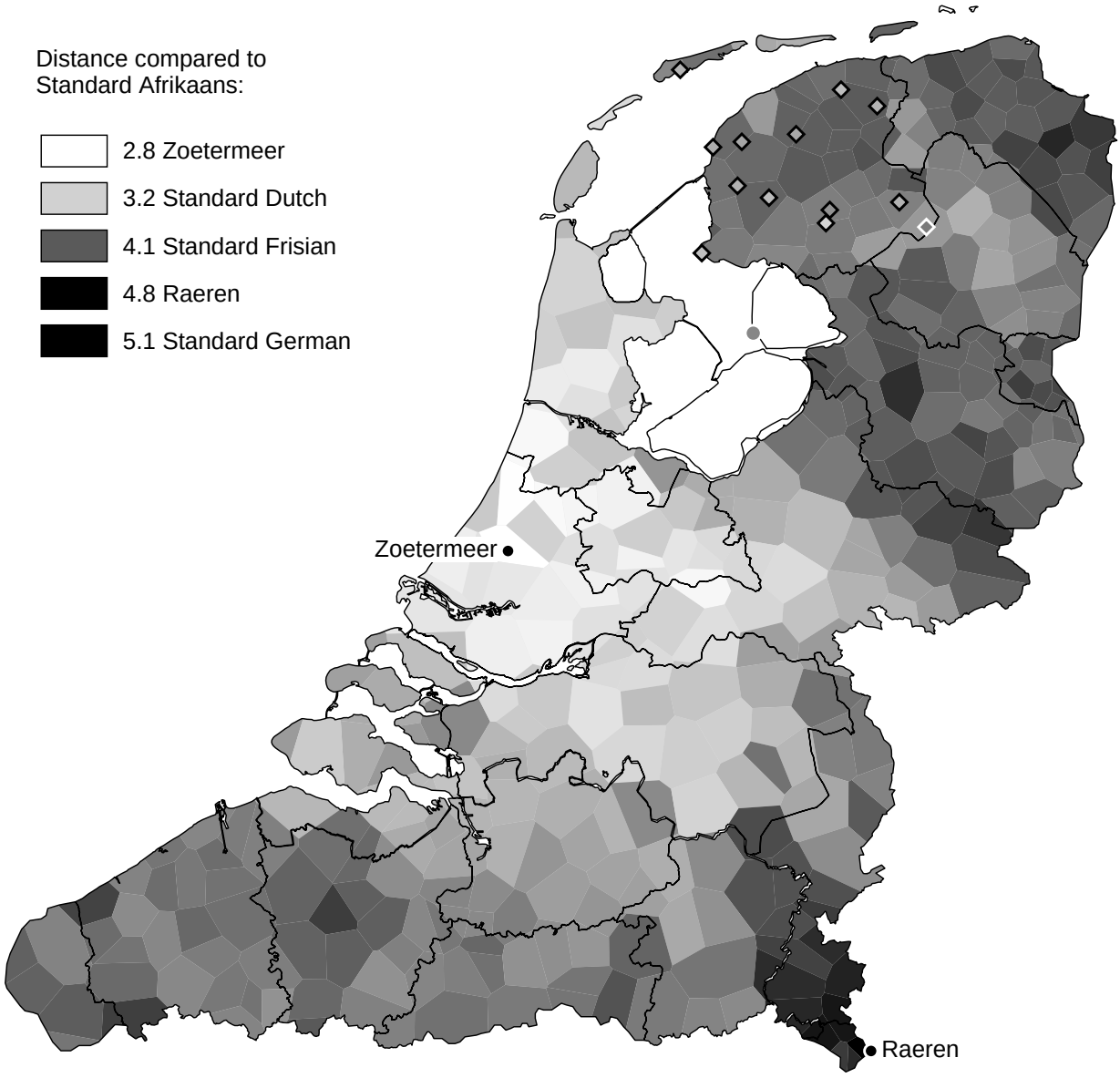


Figure 1: Distances of 361 Dutch dialect varieties compared to Afrikaans. The varieties are represented by polygons, geographic dialect islands are represented by colored dots, and linguistic dialect islands are represented by diamonds. Lighter polygons, dots or diamonds represent dialects which are closest to Afrikaans and darker ones represent the varieties which are most distant. Note that the variety of Zoetermeer is closest to Afrikaans. The IJsselmeer polders (Wieringermeerpolder, Noordoostpolder and Flevopolder) are not under consideration, so they are left white.

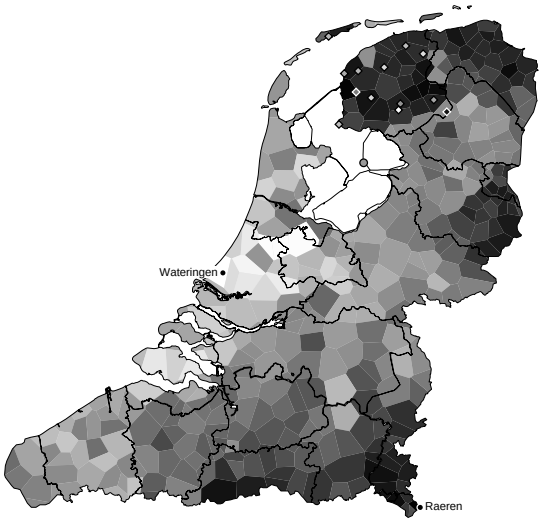


Figure 2: Vowel substitution distances of 361 Dutch dialect varieties compared to Afrikaans. Note that the variety of Wateringen is closest to Afrikaans, and the variety of Raeren is most distant.

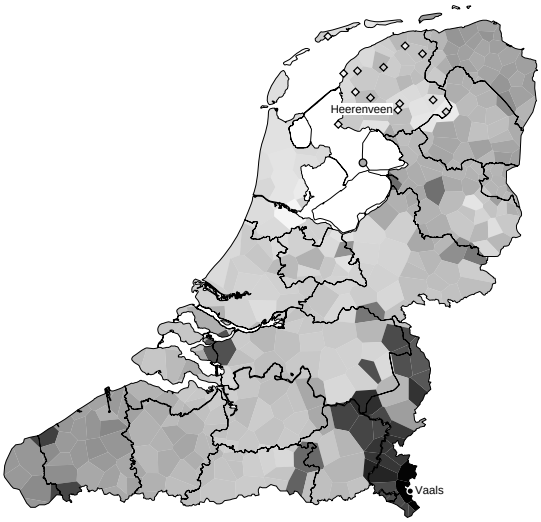


Figure 3: Consonant substitution distances of 361 Dutch dialect varieties compared to Afrikaans. Note that the variety of Heerenveen is closest to Afrikaans, and the variety of Vaals is most distant.

relatively similar to the map in Figure 1. Again the South-Hollandish varieties are close and the southern Limburg varieties are distant. The dialect of Wateringen is closest, and the dialect of Raeren is the most distant. The Frisian varieties and the core Low Saxon varieties found in Groningen and Twente are more distant than in Figure 1. The varieties close to the Dutch/French border in the Belgian province of Brabant are also relatively distant.

Our findings agree with Kloeke [2]. In the summary of his book (p. 262–263) he writes:

> The two chief sources of Afrikaans, the old dialects of South Holland on the one hand and the "High" Dutch on the other, are reflected in the vocal system. In some respect Afrikaans is of a pronounced conservative "Holland" dialectal character, still more conservative than the dialects of Holland itself, which are gradually disappearing.

Although the Holland dialects *are* disappearing, the relationship with the South-Holland varieties is still found when we use the RND data.

4.2.2. Consonants

When consonant substitution distances between the Dutch dialects and Afrikaans are calculated, a completely different picture is obtained, as can be seen in Figure 3. Closest is the town Frisian variety of Heerenveen. Other Town Frisian varieties (Harlingen, Staveren, Bolsward, Midsland and Dokkum), the dialect of Oost-Vlieland and the dialect of Amsterdam are also found among the eight closest varieties. The map shows that the Limburg varieties are again distant.

The strong relationship with the Town Frisian dialects may be explained by the fact that both in Afrikaans and in Town Frisian the initial consonant cluster in words like *schip* 'ship' and *school* 'school' is pronounced as [sk], while most other dialects and Standard Dutch have [sx]. Another shared feature is that the initial consonant in words like *vinger* 'finger' and *vijf* 'five' is a voiceless [f] and the initial consonant in words like *zee* 'sea' and *zes* 'six' is a voiceless [s]. Most other dialects and Standard Dutch have initial [v] and [z] respectively, although there may be a current tendency to increasingly unvoice these fricatives.

The relationship of Afrikaans with Town Frisian may be an unexpected outcome at first glance. According to Kloeke, Frisian did not have any significant influence on Afrikaans. But he stresses the assumption that once the [sk] pronunciation was used in the whole Dutch dialect area. Relics are presently still found in Frisia, the islands, North-Holland, Overijssel and Gelderland, but also in Noordwijk and Katwijk in South-Holland. He also suggests the possibility that, in the 17th century, there may have been large relic areas in South-Holland (see p. 225–226).

As to the unvoiced fricatives, this phenomenon is partly found in the RND transcription of the South-Hollandish variety of Zoetermeer, but not to the same extent as in the Heerenveen transcription. A similar reasoning as for the [sk] pronunciation may also apply here.

5. Conclusions

In this paper, Afrikaans was compared to the west Germanic standard languages (Dutch, Frisian and German). Afrikaans was found to be most related to Dutch. Van Reenen and Coetzee[1] rightly refer to Afrikaans as a daughter of Dutch. When Afrikaans is compared to 361 Dutch dialects, the South-Hollandish varieties were found to be closest to Afrikaans. According to Kloeke[2] the southern varieties in the province of South Holland are the main source of Afrikaans. However, our closest variety – the dialect of Zoetermeer – is found in the center of the province. We did not specifically find the southern South-Hollandish varieties to be closest. It is likely that the South-Hollandish dialect area has changed since 1652. The strong relationship between Afrikaans and the South-

Hollandish varieties can be explained by their vowels. As regards the consonants, the Town Frisian varieties are most closely related to Afrikaans, probably since they still maintain features which were lost in the South-Hollandish dialects. The southern Limburg varieties are most distant to Afrikaans, both when looking at vowel differences and when considering consonant differences.

The results of this study indicate that, for the development of automatic speech recognition systems for Afrikaans, Standard Dutch is probably the best language to "borrow" acoustic data from. The use of acoustic data of the South-Hollandish dialects would be even better, but will probably not be available, since developers of automatic speech systems focus on (accents of) standard languages rather than on dialects.

6. Acknowledgments

We thank Peter Kleiweg for the program which we used for the visualization of the maps in this paper. The program is part of the R*u*G/L^{04} package which is freely available at `http://www.let.rug.nl/~kleiweg/L04`. This research is partially supported by an NRF Focus Area grant for research on *HLT Resources for Closely-Related Languages*.

7. References

[1] P. Reenen and A. Coetzee, "Afrikaans, a daughter of Dutch," in *The Origins and Development of Emigrant Languages; Proceedings from the Second Rasmus Colloquium, Odense University, November 1994*, H. F. Nielsen and Lene Schøsler, Eds. 1996, pp. 71–101, Odense University Press.

[2] G.C. Kloeke, *Herkomst en groei van het Afrikaans*, Universitaire Pers, Leiden, 1950.

[3] J. du P. Scholtz, *Taalhistoriese Opstelle*, J.L. van Schaaik, Pretoria, 1963.

[4] Statistics South Africa, "Census 2001: Key results," Tech. Rep., Statistics South Africa, Pretoria, 2001, Available as: http://www.statssa.gov.za/PublicationsHTML/Report-03-02-012001/html/Report-03-02-012001.html.

[5] B. Kessler, "Computational dialectology in Irish Gaelic," in *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, 1995, pp. 60–67, EACL.

[6] J. Nerbonne, W. Heeringa, E. Van den Hout, P. van der Kooi, S. Otten, and W. van de Vis, "Phonetic distance between Dutch dialects," in *CLIN VI, Papers from the sixth CLIN meeting*, G. Durieux, W. Daelemans, and S. Gillis, Eds., Antwerp, 1996, pp. 185–202, University of Antwerp, Center for Dutch Language and Speech (UIA).

[7] W. J. Heeringa, *Measuring Dialect Pronunciation Differences using Levenshtein Distance*, Ph.D. thesis, Rijksuniversiteit Groningen, Groningen, 2004.

[8] Ch. Gooskens and W. Heeringa, "The position of Frisian in the Germanic language area," in *On the Boundaries of Phonology and Phonetics*, D. Gilbers, M. Schreuder, and N. Knevel, Eds., pp. 61–87. Center for Linguistics and Cognition, Groningen, Groningen, 2004.

[9] E. Blancquaert and W. Peé, Eds., *Reeks Nederlands(ch)e Dialectatlassen*, De Sikkel, Antwerpen, 1925–1982.

[10] W. Heeringa, "De selectie en digitalisatie van dialecten en woorden uit de Reeks Nederlandse Dialectatlassen," *TABU: Bulletin voor taalwetenschap*, vol. 31, no. 1/2, pp. 61–103, 2001.

[11] F. Hinskens, P. Auer, and P. Kerswill, "The study of dialect convergence and divergence: conceptual and methodological considerations," in *Dialect change. The convergence and divergence of dialects in contemporary societies*, P. Auer, F. Hinskens, and P. Kerswill, Eds., pp. 1–48. Cambridge University Press, Cambridge, 2005.

[12] E. Blancquaert, *Tekstboekje*, De Sikkel, Antwerpen, 2nd edition, 1939, Nederlandse Fonoplaten van Blancquaert en van der Plaetse, Eerste Reeks.

[13] H. Krech and U. Stötzer, *Wörterbuch der deutschen Aussprache*, Max Hueber Verlag, München, 1969.

[14] W. Heeringa, J. Nerbonne, H. Niebaum, R. Nieuweboer, and P. Kleiweg, "Dutch-German contact in and around Bentheim," in *Languages in Contact. Studies in Slavic and General Linguistics*, D. Gilbers, J. Nerbonne, and J. Schaeken, Eds., vol. 28, pp. 145–156. Rodopi, Amsterdam and Atlanta GA, 2000.

[15] J. B. Kruskal, "An overview of sequence comparison," in *Time Warps, String edits, and Macromolecules. The Theory and Practice of Sequence Comparison*, D. Sankoff and J. Kruskal, Eds., pp. 1–44. CSLI, Stanford, 2nd edition, 1999, 1st edition appeared in 1983.

[16] IPA, *The Sounds of the International Phonetic Alphabet*, Department of Phonetics and Linguistics, University College London, London, 1995, Available as audio cassette or CD.

[17] P. Boersma and D. Weenink, *Praat: doing phonetics by computer*, Institute of Phonetic Sciences, Amsterdam, 2002, Available at: `http://www.praat.org`.

Speect: a multilingual text-to-speech system

J.A.Louw

Human Language Technologies Research Group
Meraka Institute, Pretoria, South Africa
jalouw@csir.co.za

Abstract

This paper introduces a new multilingual text-to-speech system, which we call *Speect* (**Spe**ech synthesis with **e**xtensible ar**c**hitec**t**ure), aiming to address the shortcomings of using Festival as a research system and Flite as a deployment system in a multilingual development environment. Speect is implemented in C with a modular object oriented approach and a plugin architecture, aiming to separate the linguistic and acoustic dependencies from the run-time environment. A scripting language interface is provided for research and rapid development of new languages and voices. This paper discusses the motivation for a new text-to-speech system as well as the design architecture and implementation of the system. We also discuss what is still required in the development to make the new system a viable alternative to the Festival - Flite tool-chain.

1. Introduction

Text-to-speech (TTS) synthesis introduces a multitude of communication possibilities, which are especially important in developing countries for cheap and effective conveyance of information. Multilingual text-to-speech is especially important in countries with more than one official language as is the case in South Africa. Multilingual text-to-speech, as used in this paper, refers to *simple multilingual speech synthesis* [1] where language switching is usually accompanied by voice switching. There are many high-quality commercial text-to-speech systems available for the major spoken languages, but not so for languages with a small geographical distribution or a small number of speakers relative to the major languages. Development of these technologies is a daunting task, and in multilingual environments even more so.

Text-to-speech synthesis is the automated process of mapping a textual representation of an utterance into a sequence of numbers representing the samples of synthesized speech [2]. This conversion is achieved in two stages as depicted in figure 1.

- *Natural Language Processing (NLP):* Converting the textual representation of an utterance into symbolic linguistic units.
- *Digital Signal Processing (DSP):* Mapping the symbolic linguistic units into samples of synthesized speech.

The *Natural Language Processing* stage consists of the following major modules:

- Text *pre-processing* involves the transformation of the textual input into a format suitable for the phonetization module. The specifics of this task is dependent on the type of textual input given to the system and includes utterance chunking and text normalization.
- The normalized text of the pre-processing module is converted into a phonetic representation by the *phonetization* block.
- *Prosody generation* involves the generation of intonation and duration targets through some form of prosody models.

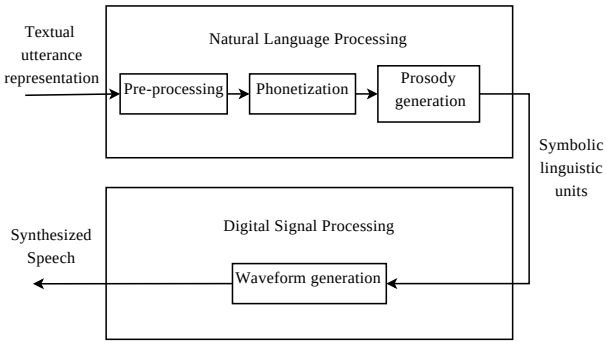


Figure 1: Functional blocks of a text-to-speech synthesizer.

The data generated by the NLP stage represents the *symbolic linguistic units*, which are then converted into synthetic speech by the *Digital Signal Processing* stage. The DSP stage can be realized by means of unit selection [3], statistical parametric synthesis [4], formant synthesis [5], or some other type of synthesizer technology. Each of the modules in the two stages adds some type of information to the initial given utterance which enables the final module, *waveform generation*, to generate synthetic speech based on this information.

The NLP stage is language dependent, whereas the DSP stage is dependent on the synthesizer technology of the implemented synthetic voice. Therefore, a multilingual text-to-speech system must be able to apply different NLP and DSP modules for different synthetic voices based on the language and synthesizer technology of the specific voice.

The next section discusses the motivation behind the need for a new speech synthesizer, followed by the design and implementation. We then conclude with a discussion.

2. Motivation

Over the last decade, the Festival speech synthesis system [6] has become the de facto standard free toolkit for speech synthesis research [7]. Festival provides a modular architecture whereby it is possible to modify each of the sub-tasks involved in the NLP and DSP stages in a text-to-speech conver-

sion process. Festival is implemented in two languages, C++ and Scheme (a lisp dialect), providing an integrated interpreted language for run-time manipulation. Festival, together with the Festvox project [8], aims to make the building of text-to-speech voices a structured and well defined task.

While being a fine example of a research system there are drawbacks to using Festival as a component within a speech enabled technology solution such as an *integrated voice response* (IVR). Festival has a large memory footprint and is relatively slow as a result of having a self contained interpreted language. A Festival compatible alternative is the Flite [9] synthesis engine, and while having a similar modular architecture and utterance structure representation, it provides improvements with regards to [9];

- speed,
- portability,
- maintenance,
- code size,
- data size, and
- thread safety.

Flite was written in ANSI C and has no interpreted language. In Festival a synthetic voice is loaded into internal data structures into memory, while in Flite all voice data is represented in C code. Therefore one still needs to use Festival and the Festvox toolkit for research and development of new voices, and then convert these voices with appropriate scripts into a Flite compatible version. The process of building a new voice in a new language (a language where the NLP modules do not exists in either Festival or Flite) will require one to first develop the NLP modules in C++ and/or Scheme in Festival and then rewrite these modules in C code for use in Flite. This is time consuming and requires expert knowledge of the Festival and Flite code base.

As a result of our experience with multilingual text-to-speech development we decided to design and implement a new text-to-speech system that combines the best features of the existing Festival and Flite synthesis engines while also addressing the shortcomings of these systems with regards to our requirements. The most important requirements for the new system, which we call *Speect* (**Spe**ech synthesis with **e**xtensible ar**c**hitec**t**ure), can be summarized as follows:

- **A single synthesis engine**: Having one synthesis engine reduces the code base and will eliminate any discrepancies between a development system and deployment system. This also leads to less maintenance.
- **Extensible architecture**: It should be easy to extend and modify the system with regards to the NLP as well as DSP stages of the text-to-speech conversion process.

3. Design

A synthetic voice in a TTS system can be seen as a combination of two parts

- **linguistic component**: providing language models and data for the NLP stage of the synthesis process.
- **acoustic component**: the acoustic models and data required by the DSP stage for waveform generation.

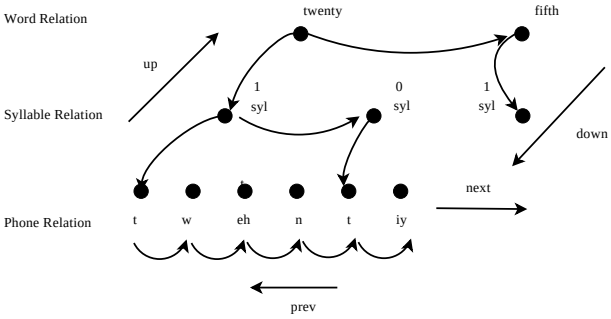


Figure 2: An example representation of an utterance structure using a heterogeneous relation graph.

The linguistic component is language dependent and can be shared by voices of the same language while the acoustic component is unique to a specific voice. Speect aims to provide control of the synthesis process and its design is intended to be independent of the underlying linguistic or acoustic models and data. Speect is not meant to replace speech processing tool-kits, the linguistic and acoustic models and data still needs to be generated by packages such as Edinburgh Speech Tools [10], Festvox and the Speech Signal Processing Toolkit [11].

To allow existing linguistic and acoustic Festival models and data to be reused, the internal representation of an utterance follows the same formalism as used in Festival and Flite. The *utterance structure* is represented internally as a *Heterogeneous Relation Graph* [12] (HRG), which consists of a set of relations, where each relation contains some items (the items need not be unique to a relation). The relations represent structures such as words, syllables, phonemes or even duration targets and the items are the content of these structures. Figure 2 shows an example representation of an utterance structure using a HRG with three relations and their items.

The individual NLP and DSP modules of figure 1 are called *utterance processors*. Utterance processors create relations in the utterance structure and add information (items with features) to the relations based on the linguistic and acoustics models and data. For example in figure 2 the *syllable relation* of the utterance has three items, with syllable stress as a feature of the items.

Speect has an object oriented design which allows the same modular approach to text-to-speech as Festival and Flite. A plugin architecture is used for the utterance processors, thereby restricting the language dependencies within the data and resources of the specific voice implementation and not in the synthesis platform. This plugin architecture allows different implementations of the same voice and/or language to be used during run-time, as the voice and language specifications load the required plugins.

4. Implementation

Speect is implemented in ANSI C to provide maximum portability and speed. The implementation of an object oriented paradigm in C requires more discipline from the programmer, but allows for code reuse and a modular design. Figure 3 shows the implementation architecture of Speect.

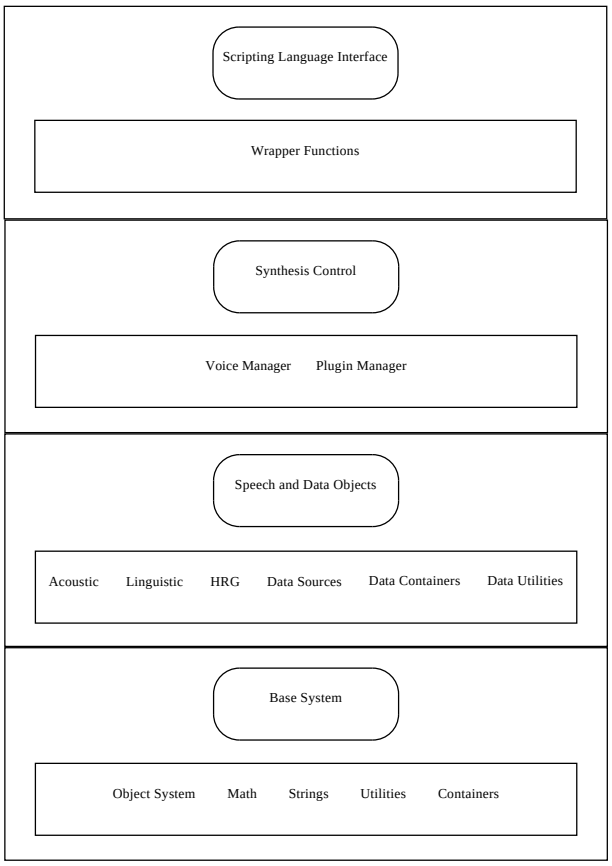


Figure 3: The Speect architecture.

The Speect architecture is divided into 4 major sections

- The **base system** provides a library of basic functions that are used by the upper levels of the system.
  - *object system*: the objects system implements a object oriented paradigm in C, whereby an object is described by two structures, one for it's data members and one for it's methods. The object system provides basic encapsulation, polymorphism, and inheritance.
  - *math routines*: basic mathematical routines.
  - *utility functions*: memory allocation and logging utilities.
  - *string functions*: basic string functions and UTF 8 support.
  - *basic containers*: doubly-linked lists and a hash table as basic data containers.
- **Speech and Data Objects** offer higher level objects specific to speech synthesis and data handling.
  - *acoustic objects*: provides *interfaces* to waveforms, data tracks, etc. Interfaces are implemented by plugins, therefore removing data dependencies from the synthesis system.
  - *linguistic objects*: provides *interfaces* to phoneset, lexicon, etc. Plugins implement the linguistic interfaces.
  - *HRG objects*: the utterance structure implementation. Follows the implementation of Festival and Flite for representing utterances.
  - *data sources*: objects and interfaces for reading and writing data from/to files and memory. An *Extensible Binary Meta Language [13]* protocol is implemented as the standard format for reading/writing to files.
  - *data containers*: Abstract objects that encapsulate the use of the base system containers.
  - *data utilities*: the basic data object used in the HRG system. All objects that inherit from this object can be used as a feature in the utterance structure.
- **Synthesis Control** is provides the top level control of voices.
  - *plugin manager*: handles requests for specific plugin implementations. Dynamically loads and unloads plugins as required by the system.
  - *voice manager*: loads and unloads voices and handles synthesis requests.
- **Scripting language interface** connects interpreted scripting languages to the Speect library.
  - *wrapper functions*: the connection between the Speect library and scripting languages through SWIG (Simplified Wrapper and Interface Generator) [14].

The scripting language interface enables one to use Speect in an interpreted language setting, therefore speeding up research and development of new voices and languages. The speed of the Speect library is not influenced by the scripting language as it is external to the library implementation.

The work-flow of Speect is as follows: a synthesis request must be accompanied by the desired voice. The voice specification, which consists of a list of linguistic and acoustic utterance processors and associated data, is loaded by the *voice manager*. The desired utterance processor plugins are loaded dynamically by the *plugin manager* on request from the voice manager. The voice manager then proceeds to execute each of the utterance processors on the textual utterance representation, building an utterance structure. The utterance structure is synthesized and the synthetic speech returned.

5. Discussion

The Festival speech synthesis system provides a research and development platform for building synthetic voices in different languages. However, it is challenging to use in a real world deployment environment because of it's size and speed. Flite aims to correct these deficiencies with a much smaller and more efficient implementation, but lacks the development environment and suffers from language dependencies in the data and resources. Therefore, to develop synthetic voices for deployment one needs to create the voice in Festival and

convert it to a Flite suitable format. This is a complicated task, especially for new languages and requires extensive knowledge of the Festival and Flite code base.

Speect aims to be an alternative to the Festival - Flite tool-chain by providing a single speech synthesis engine for research, development and deployment in multilingual environments. This is achieved by a modular object oriented design with a plugin architecture, thereby separating the synthesis engine from the linguistic and acoustic dependencies. The improvements of the proposed Speect synthesis system with regards to the Festival - Flite tool-chain can be summarized as follows:

- The research, development and deployment cycle is done with one synthesis engine, reducing the size of the code base as well as the required maintenance. Therefore, implementation of new NLP or DSP plugins requires expert knowledge of just one synthesis engine.
- Run-time performance comparable with that of Flite, while retaining the research and development advantages of the Festival design, without the speed and size penalties associated with the integrated interpreted language because of the separation of the core library and the interpreted language.
- Footprint size comparable to Flite due to plugin architecture, therefore only the required modules for a particular voice are loaded.

The modular object oriented design combined with the SWIG interface enables the use of the Speect library through native calls from multiple scripting languages, and other languages such as Java, C#, Scheme and Ocaml, while encapsulating the underlying implementation through the use of the plugin architecture.

The Speect system has been completed up to a stage where utterance processor plugins can be loaded and run on basic input text and a concatenative unit selection method as described in [7], but to be a viable alternative to the current system the following still needs to be addressed:

- SWIG interface files for Python,
- Python scripts for the creation of unit selection voices,
- NLP modules for different languages,
- complete documentation on the implementation,
- manual for writing and extending plugins,
- documentation for building voices, and
- scripts for converting existing Festival voices into a Speect format.

6. References

[1] Traber, C., Huber, K., et al. “From multilingual to polyglot speech synthesis”, In Proceedings of Eurospeech, pp. 835-838, Budapest, Hungary, September, 1999.

[2] Stylianou, Y., “Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification”, Ph.D. Thesis, Ecole Nationale Superieure des Telecommunications, Paris, France, 1996.

[3] Hunt, A. and Black, A. “Unit selection in a concatenative speech synthesis system using a large speech database”, In Proceedings of ICASSP, vol 1, pp. 373-376, Atlanta, Georgia, 1996.

[4] Black, A., Zen, H., and Tokuda, K. “Statistical Parametric Synthesis”, Proceedings of ICASSP, pp. 1229–1232, Hawaii, 2007.

[5] Högberg, J. “Data driven formant synthesis”, In Proceedings of Eurospeech, pp. 565-568, Greece, 1997.

[6] Taylor, P., Black, A.W., and Caley, R. “The architecture of the Festival Speech Synthesis System”, 3rd ESCA Workshop on Speech Synthesis, pp. 147-151, Jenolan Caves, Australia, 1998.

[7] Clark, R.A.J., Richmond, K., and King, S. “Multisyn: Open-domain unit selection for the Festival speech synthesis system.”, Speech Communication 49:317-330, 2007.

[8] Black, A. and Lenzo, K. “Building Voices in the Festival Speech Synthesis System”, http://www.festvox.org/festvox/bsv.ps.gz, 2003.

[9] Black, A. and Lenzo, K. “Flite: a small fast run-time synthesis engine”, 4th ISCA Speech Synthesis Workshop, pp. 157-162, Scotland, 2001.

[10] The Centre for Speech Technology Research, The University of Edinburgh, The Edinburgh Speech Tools Library, http://www.cstr.ed.ac.uk/projects/speech_tools/.

[11] Department of Computer Science, Nagoya Institute of Technology, Speech Signal Processing Toolkit, SPTK 3.1. Reference manual, http://downloads.sourceforge.net/sptk/SPTKref-3.1.pdf.

[12] Taylor, P., Black, A.W., and Caley, R. “Heterogeneous relation graphs as a mechanism for representing linguistic information”, Speech Communication 33:153-174, 2001.

[13] Extensible Binary Meta Language, http://ebml.sourceforge.net/.

[14] Beazley, D., “Swig: An easy to use tool for integrating scripting languages with c and c++”, Presented at the 4th Tcl/Tk Workshop, Monterey, California, 1996.

Afrikaans Homophone Disambiguation

Hendrik J. Groenewald and Marissa van Rooyen

Centre for Text Technology
North-West University (Potchefstroom Campus), South-Africa
handre.groenewald@nwu.ac.za; 13017527@student.nwu.ac.za

Abstract

Homophone disambiguation is a very important natural language processing task in any language. It is for example an essential prerequisite for effective machine translation and an important component of a grammar checker. In this paper we describe two different approaches to homophone disambiguation in Afrikaans, namely a frequency-based and a decision tree-based approach. We describe the data requirements and operation of the two methods. We also compare the two methods on the basis of the results obtained from evaluation on the same evaluation data set. We conclude that the frequency-based approach is currently more suitable for implementation in a grammar checker, despite the fact that the decision tree-based approach obtains a higher recall figure.

1. Introduction

"Waterkloof se eerste span het die lood [sic] gewen en eerste gekolf. (Waterkloof's first team won the lead [*sic*] and batted first.) The above sentence was obtained from the Afrikaans Newspaper, *"Beeld"*, of 16 September 2008 [1]. This is an example of a sentence that contains a homophone that has been used incorrectly, since the word *"lood"* (lead) has not been used in the correct context. The correct word choice would have been to use the word *"loot"* (toss) which means to flip coins in order to decide about an issue [2].

Incorrect usage of homophones is unfortunately a very common problem in written work and is not only restricted to Afrikaans. The origin of the problem is that homophone words like *"loot"* and *"lood"* sound very similar when pronounced and have similar spelling, although the meanings and origin of the two words are completely different.

A spelling checker for Afrikaans, like that of CTexT [3], will however not be able to detect homophone errors. The reason for this is that all the words in the sentence are spelt correctly, despite the fact that the homophone word is not used in the correct context. The spelling checker only evaluates surface forms of words and cannot flag grammatical errors [4].

CTexT is currently developing a grammar checker for Afrikaans that will attempt to address this problem. Incorrect usage of homophones is just one of the many grammatical errors that we want the grammar checker to identify and correct. Obtaining an accurate method for Afrikaans homophone disambiguation for purposes of implementation in the grammar checker motivated this research. This is also a first step in the process of constructing a fully-fledged word sense disambiguator and a context-sensitive spelling checker for Afrikaans.

The remainder of this paper is structured as follows: Section 2 introduces the frequency-based approach to homophone disambiguation and further provides detailed information about the implementation and evaluation of the method. Section 3 focuses on an alternative decision tree-based approach. We conclude with a comparison of the two approaches and some directions for future work in Section 4.

2. Method 1: Frequency-Based

2.1. Background

The frequency-based approach is based on the assumption that the existence of certain words that co-occur with homophone words in a sentence contributes to the process of homophone disambiguation [5]. It is for example highly unlikely that a sentence containing the word *"lood"* (lead) will also contain the words "win", "lose", "heads", "tails" and "referee", since all these words refer to entities or concepts that are associated with tossing before a sporting game. For the same reason it is also unlikely that the word *"loot"* (toss) will co-occur with the words "ammunition", "weapons", "poison", "element" etc. This approach is also revered to as a bag-of-words approach.

2.2. Approach

The first step of this approach was to compile a list of the homophone pairs that exist in Afrikaans. This resulted in a list of 469 homophone pairs. The next step was to extract all sentences containing homophones in our list of homophone pairs from the Media24 Corpus [6]. For some of the homophone pairs we experienced large differences between the numbers of sentences that were extracted for each homophone word. This can be attributed to the vast differences in frequency of use of some of the homophone words. The homophone *"dit"* (this) has for example a frequency of 577,864 in the Media24 Corpus [6], while *"dut"* (nap or snooze) has a significantly lower frequency of 50. For purposes of this research we only considered 50 homophone pairs in total. 25 of the homophone pairs were selected where the homophone words have more or less the same frequency, while the remaining 25 were selected from pairs where large differences between the frequencies of the homophone words exist.

These extracted sentences were used to create a word list of all the words (together with their frequency counts) that co-occur with each of the 100 homophones (50 pairs). These word lists are the so-called constraint words, words that are not likely to occur with the other homophone in the pair.

All function words were removed from these word lists, since function words have little ambiguous meaning and would therefore not contribute to the disambiguation process. For the same reason we also removed all named entities, abbreviations, foreign words and words that contain spelling errors with the aid of the *"Afrikaanse Speltoetser"* [3].

The next step was to normalise the frequency counts of all

the constraint words in the different word lists. This was calculated by dividing the frequency count of the word, by the sum of the frequency counts of all the words in the list. The constraint lists with the normalised frequencies form the basis of our implementation of the frequency-based disambiguation algorithm.

2.3. Process

The process of homophone disambiguation with the frequency based approach starts with a document that needs to be checked for homophones that have been used in the wrong context. The document is firstly sentencised and the process continues on a per-sentence basis. The sentence is then checked to determine if it contains a homophone word. If it does not, the process continues with the next sentence. If a sentence does contain a homophone, the sentence is passed to the summation module. The summation module compares the words in the sentence with the constraint words of the involved homophone. The normalised frequency value of all the constraint words that are found in the sentence is summed. If the value of the sum is above a certain threshold value, the homophone is flagged and the other homophone in the pair is suggested. The process ends when all the sentences in the document have been checked. Figure 1 shows a flow diagram of the entire process.

2.4. Evaluation

The evaluation of both methods was performed on a separate evaluation set from the test set, originating from the PUKProtea Corpus. The evaluation data was created by extracting sentences containing the 50 homophone pairs. 25 of these pairs had similar amounts of data and 25 showed large differences, as discussed in Section 2.2. The results for these two groups are given separately in Table 1 and 2 to indicate the influence of the amount of data collected on the different metrics.

The evaluation data consisted of 150 sentences containing homophone words that were used in the correct context and 150 sentences containing homophone words that were incorrectly used. These sentences were created artificially to contain either the correct or incorrect homophone. The reason for evaluating with data containing no errors is that we are (for purposes of the grammar checker) not only interested in detecting homophone errors, we also want to make sure that we do not flag correct words as incorrect. Table 1 indicates the results obtained with the frequency-based approach.

Table 1: *Results obtained with Method 1.*

	Recall	Precsion	F-Score
Equal	0.355	0.862	0.503
Unequal	0.205	0.914	0.335
Average	0.28	0.888	0.419

Table 1 indicates that Method 1 has a relatively low recall figure, but high precision. The system performs better on homophone pairs that have equal frequencies in the Media24 Corpus [6]. The low recall figures are not ideal for implementation in a grammar checker.

The threshold value can however be adjusted to improve the recall figure, but this will be at the expense of precision. We decided against adjusting the threshold value for increased recall, since we believe that in the context of a grammar checker it is better to flag a low number of errors (low recall) with high precision. The disappointing results obtained with Method 1

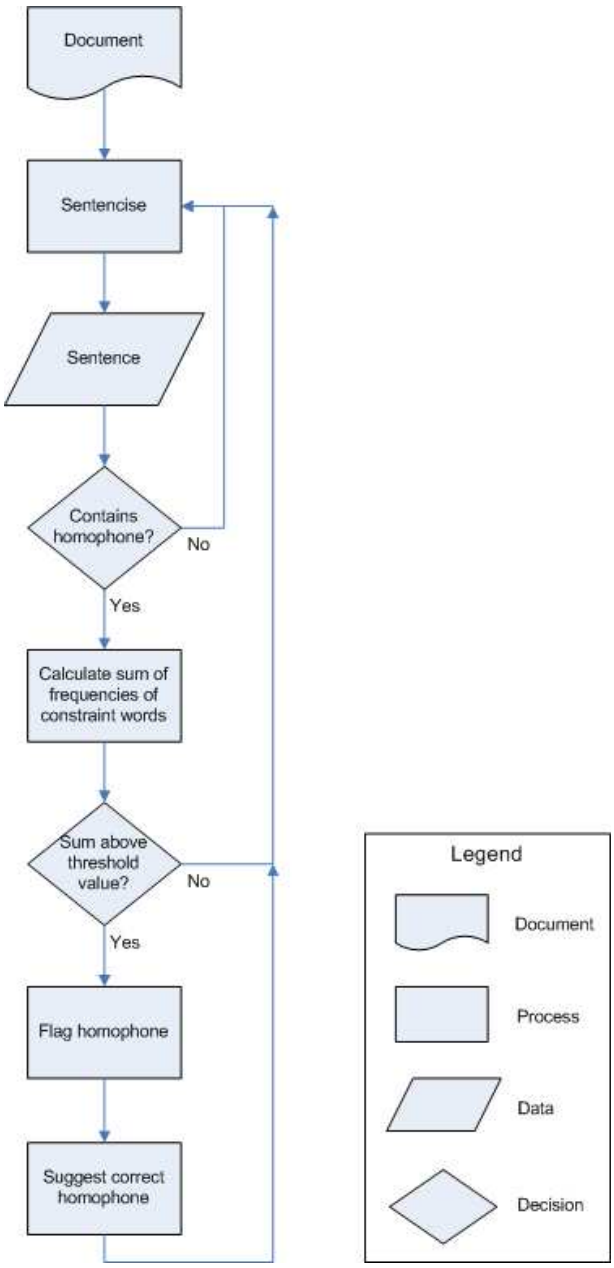


Figure 1: *Flow Diagram of Method 1.*

prompted us to consider alternative methods for homophone disambiguation. Method 1 thus serves as benchmark for all future work on the subject in Afrikaans, particularly at CTexT.

3. Method 2: Desicion Tree-Based

3.1. Background

The second part of this study is based on a similar experiment by Daelemans and Van Den Bosch - they studied the effect of part of speech tags and word frequencies on word sense disambiguation for Dutch [7]. According to their study a human makes the choice between two homophones based on one of two criteria: Firstly, a person explicitly learns when to use which form from an early age. This knowledge builds in a trial and error fashion;

a person uses whatever form comes naturally until a teacher or parent corrects him/her. From this the person learns that certain words are only used in restricted contexts (the basis for the first module discussed). If a person has never encountered a certain form, he/she reverts to the second method: using the most frequent word. For most pairs of homophones, one is used much more often than the other, as can be seen from the differences in the frequency of use of some homophone words discussed earlier. The most frequent form is thus considered the most prototypical. According to Daelemans and Van den Bosch [7], this second method is 98% accurate. Such a one-sided approach is however not suitable for implementation in a grammar checker.

Using this knowledge, and the first method as baseline, we constructed a second module an adaptation of Daelemans and Van Den Boschs work [7], while incorporating ideas from Wilkss earlier work in the field [8]. This method uses decision trees instead of the frequency-based method to construct a classifier that can indicate whether or not a sentence contains a homophone that has been used in the wrong context.

3.2. Approach

The machine learning algorithm that was used to create the classifier is a decision tree algorithm called IGTree. IGTree is one of the algorithms contained in the Tilburg Memory-Based Learner (TIMBL) [9], a program that implements several memory-based machine learning techniques and algorithms. IGTree was chosen because of its fast speed of classification and relative high accuracy. We decided to construct a separate decision tree for each homophone pair, since we want to prevent the construction of large, complicated decision trees that slows down the classification process.

The high precision obtained by Method 1 proved that the context words are indeed a good indicator of the meaning and usage of a homophone. The features in the training data consist of five context words to the left of the homophone, and five to the right. In an attempt to improve the low recall of Method 1, we decided to add part-of-speech tags to this experiment. Daelemans and Van den Bosch [7] also obtained very good results with the use of part-of-speech tags, since they obtained a 5% increase in accuracy. The TnT Tagger [10] was used for this part of the experiment, with a tag set developed for Afrikaans at CTexT [11]. The purpose of the classifier is to classify a sentence by explicitly indicating the correct homophone word.

A flow diagram of the resulting module is shown in Figure 2. The process once again starts with a document that needs to be checked for homophones that have been used in the wrong context. The document is also sentencised and the process continues on a per-sentence basis. The sentence is then checked to determine if it contains a homophone word. If the sentence does contain a homophone, it is tagged and then windowed. This step is necessary to ensure that the sentence conforms to the format of the training data. The windowed sentence is then classified. If the awarded class is the same as the original homophone, the process continues with the next sentence. If the awarded class does not match the original homophone, the homophone is flagged and the other homophone in the pair is suggested.

3.3. Process

The steps for compilation of this second module are as follows:

1. Compile a list of homophones to be used (in this case, 50 pairs were selected manually for their usage in Afrikaans, and the amount of data available).

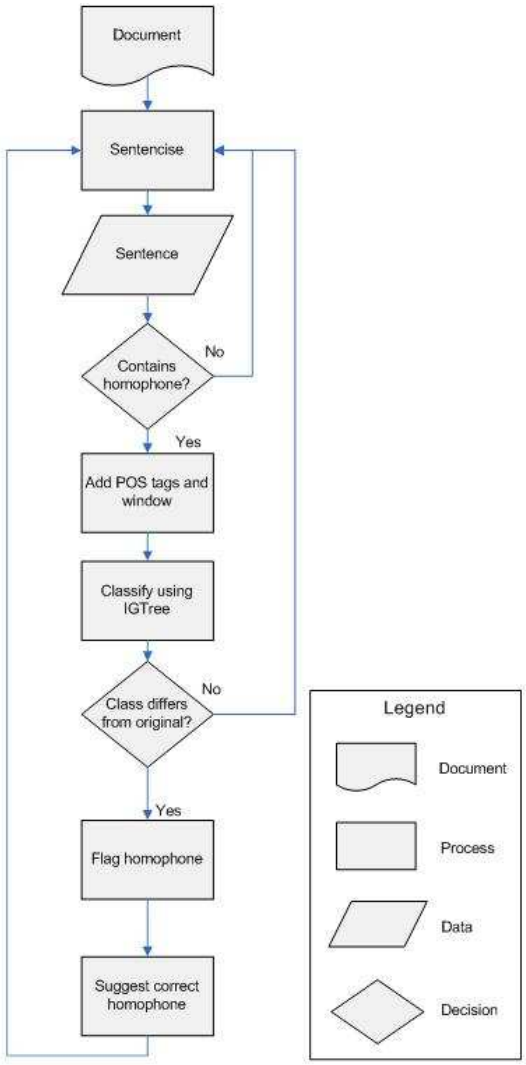


Figure 2: *Flow Diagram of Method 2.*

2. Extract sentences containing these homophones. The sentences are then tokenised for the next step.
3. Add POS-tags to all words in the sentence.
4. Add all components to each other and window the input for IGTree. A line will then contain 10 context words and their tags. If there are not enough words to either side of the homophone, empty features were added so that each instance held 20 features.
5. Train the trees with IGTree in TiMBL (no special weights or other parameters were added or adjusted).
6. Evaluate the module using TiMBL and data extracted from the PUKProtea Corpus in the same way as the training data.
7. Calculate the recall, precision and F-score for each tree separately, as well as a mean for the entire set.

3.4. Evaluation

Table 2 displays the results obtained with Method 2 by using the same evaluation data as Method 1:

Table 2: *Results obtained with Method 2.*

	Recall	Precsion	F-Score
Equal	0.724	0.753	0.738
Unequal	0.608	0.757	0.674
Average	0.665	0.755	0.708

The above results were derived from a very limited amount of training data. For some pairs as little as 500 kb of data could be extracted from the Media24 Corpus [5].

For a number of pairs, a recall and precision figure of 100% was obtained. This was very often the case for pairs with relatively large data sets for both words in the pair and very diverse meanings (e.g. *"buur"* (neighbour) vs. *"bier"* (alcoholic drink). From Table 2 it is also clear that the precision stayed roughly the same for equal or unequal amounts of data. It is only in recall that the module suffers. This can be adjusted by adding more data for the scarcer form. Another interesting phenomenon is that the decision tree chose the more frequent form rather than the other in ambiguous cases, just as a human would. This happened in all of the relevant instances.

4. Conclusion

A comparison of the results of the two methods shows that the second method obtains a higher overall recall figure than the frequency-based method. The higher recall can be attributed to the use of part-of-speech tags. Unfortunately, the decision tree-based methods obtains lower precision than that of the first method. This lower precision figure currently makes the decision tree-based approach unsuitable for implementation in a grammar checker for Afrikaans. Such a low precision figure will result in a large number of false positive classifications that might be very frustrating to the user of the grammar checker.

Our future work will however focus on improving the precision of the decision tree-based approach, since we believe that this approach is more likely to deliver better results than the frequency-based approach. It is a well-known fact that decision tree-based algorithms like IGTree, require large amounts of training data. This was also evident from the very good results obtained with the decision tree-based approach for the homophone pairs where both homophone words had high frequencies in the Media24 Corpus [6]. We should therefore try to increase the training data of the homophones with a low frequency. This can be done by using larger corpora for the extraction of the training data. Another possible solution might be to use a web crawler to automatically obtain more corpora from the web.

Another option that might also have a positive effect on the precision of the decision tree-based approach is to add more additional information than only part-of-speech tags to the training data. An example of such information that can be included is the lemmas of the context words. Improving the accuracy of the part-of-speech tagger may also improve the accuracy of the classifier.

5. References

[1] Grobler, R. *"Menlo klop Klofies weer"*. IN Beeld, 16 September 2008.

[2] The Free Dictionary, http://www.thefreedictionary.com/toss, Date of use: 3 October 2008.

[3] *"SkryfGoed 2008"*, Centre for Text Technology, North-West University, 2008.

[4] Verberne, S., *"Context-sensitive spell checking based on word trigram probabilities"*, University of Nijmegem, 2002.

[5] Jurafski, D and Martin, J.H. *"Speech and Language Processing"*. Upper Saddle River, NJ: Prentice Hall. Chapter 17, pp. 647, 2000.

[6] Pharos Dictionaries, Media24 Corpus, Cape Town, N.d.

[7] Daelemans, W. and Van Den Bosch, A.,*"Dat gebeurd mei niet: Computationele modellen voor verwarbare homofonen"*,In: Dominiek, S., Rymenans, R., Cuvelier, P., and Van Petegem, P. (Eds), *"Tussen taal, spelling en onderwijs: Essays bij het emeritaat van Frans Daems."* Academica Press, Gent, pp. 199-210, 2007.

[8] Ide, N and Veronis, J. *"Introduction to the special issue on word sense disambiguation: The state of the art."*, Computational Linguistics, 24(1):140, 1998.

[9] Daelemans, W., Zavrel, J., Van Der Sloot, K., and Van Den Bosch, A., *"TiMBL: Tillburg Memory-Based Learner Version 6.0."*, Tillburg, 2007.

[10] Brants, T., *"TnT - A statistical Part-of-Speech Tagger Version 6.0."*, Saarland University Saarbrucken, 2000.

[11] Pilon, S. *"Outomatiese Afrikaanse Woordsoortetiketteering"*. Potchefstroom: North-West University. 2005.

POSTER ABSTRACTS

Improving Iris-based Personal Identification using Maximum Rectangular Region Detection
Serestina Viriri and Jules-R Tapamo

Iris recognition is proving to be one of the most reliable biometric traits for personal identification. In fact, iris patterns have stable, invariant and distinctive features for personal identification. In this paper, we propose a new algorithm that detects the largest non-occluded rectangular part of the iris as region of interest (ROI). Thereafter, a cumulative-sum-based grey change analysis algorithm is applied to the ROI to extract features for recognition.

This method could possibly be utilized for partial iris recognition since it relaxes the requirement of using the whole part of the iris to produce an iris template. Preliminary experimental results carried on a CASIA iris database, show that the approach is promisingly effective and efficient.

Impact Assessment for Data Imputation using Computational Intelligence Techniques
F. A. Netshiongolwe, J. Mistry, F. V. Nelwamondo, and T. Marwala

In this paper, the statistical properties and accuracy levels of estimating missing data using computational intelligence techniques are evaluated. Autoencouders and conventional feedforward neural network architectures that use genetic algorithm optimization have been implemented in imputing missing features from an antenatal survey conducted in South Africa in 2001. The use of autoencoders results in outcomes that have considerably high accuracies and this also results in outcomes that preserve the variability of the data. The developed models show that the computationally predicted values preserve the mean of the original data to within 5% and 15% of its value during single feature imputation and simultaneous imputation of three missing features respectively.

The Kernel Fisher Discriminant for learning bioinformatic data sets
Hugh Murrell

Support Vector Machines have long been used as machine learning tools for bioinformatic data sets. The trick is to make use of a string based kernel. In this article we introduce a simpler kernel machine, the *Kernel Fisher Discriminant*. A *Mathematica* package *MathKFD* is described for carrying out Kernel Fisher Discrimination on bioinformatic data sets.

Evaluating techniques to binarize historic cosmic-ray data
Tjaard Du Plessis and Gunther Drevin

Two adaptive image binarization techniques are evaluated to find the algorithm best suited for the binarization of historic cosmic-ray data. The two techniques are implemented and their parameters are manipulated to find an optimal binarization for each of them. They are then compared to each other in order to choose the best suited technique.

Inductive Reasoning in Description Logics
Ken Halland and Katarina Britz

Inductive reasoning is a form of inconclusive reasoning for making generalisations based on observations. In the field of pattern recognition, inductive reasoning is often called learning[4], where general rules are derived from empirical data in the context of some background knowledge.

We restrict our attention to what we call qualitative inductive generalisations. In other words, we attach no statistical or probabilistic values to observations or inferences from them. For example, we consider arguments of the form “All observed Fs are Gs, therefore all Fs are Gs” rather than “X percent of all observed Fs are Gs, therefore X percent of all Fs are Gs”.

Inductive generalisations are by their nature ampliative and defeasible. They are ampliative in that they allow the inference of knowledge beyond what is observed. They are defeasible in that they are vulnerable to counter-examples, in which case they may need to be retracted or refined in some way.

Description logics (DLs) are a family of logics for knowledge representation. DLs are used for specifying classes of objects (or concepts) and the relationships between them [1].

The observations involved in inductive reasoning typically consist of examples, and the generalizations capture the commonalities between the examples. Description logics, with their division of a knowledge base into an ABox consisting of assertional knowledge about individuals and a TBox consisting of axiomatic knowledge about groups or classes (of individuals), provide an ideal formalism for specifying generalisations over examples. in other words, from the assertional knowledge about individuals we can make inductive generalisations in the form of axiomatic statements about groups.

In this paper we identify four different kinds of inductive reasoning that can be performed in description logics: Concept induction (or concept learning [7]) involves giving a definition of a new concept in terms of existing concepts in the knowledge base, based on the characteristics of a sample of individuals.

ABox induction involves the inference of relationships between existing classes based on the knowledge about all individuals that belong to some class.

TBox induction involves the inference of a TBox axiom which summarises or generalises a number of other TBox axioms. For example, if the TBox contains a number of axioms stating that various concepts are subclasses of a particular concept, but are also disjoint from some other particular concept. Then we can inductively infer that the two (particular) concepts are disjoint.

Finally, knowledge base induction is an abstraction of the other three forms of induction, allowing the inference of a generalization over a combination of ABox and TBox statements.

An optimised parametric speech synthesis model based on Linear prediction (LP) and the Harmonic plus noise model (HNM)
Allen Mamombe, Beatrys Lacquet, and Ms Shuma-Iwisi

Linear predictive speech synthesis plays an important role in acoustic verification and analysis. This is because system parameters can be tuned to account for prosody and intonation. The quality and intelligence of speech produced from such parametric synthesisers however falls short of many people expectations. In this paper we discuss a parametric speech model based on Linear Prediction (LP) and Harmonic plus Noise Model (HNM). We investigate ways of optimising our LP parameters and window lengths. We describe a mathematical model for LP and HNM speech synthesis. Mean opinion score (MOS) and transcription tests were then carried out on English phonemes and words synthesised using our model and renowned LP models i.e Rosenburg-Klatt (R-K) and Unit impulse. The test sample was composed of 20 native South African English listeners. The results of both tests favoured speech synthesised with our LP/HNM model when compared with renowned LP models based on the R-K and Unit impulse.

Segmentation of Candidate Bacillus Objects in Ziehl Neelsen Stained Sputum Images Using Deformable Models
Ronald Dendere, Sriram Krishnan, Andrew Whitelaw, Konstantinos Veropoulos, Genevieve Learmonth, and Tania S. Douglas

The process of automating the detection of tuberculosis (TB) in Ziehl-Neelsen (ZN) stained sputum samples seeks to address the issue of physical demand on technicians and to achieve faster diagnosis to cope with the rising number of TB cases. We explore the use of parametric and geometric deformable models for segmentation of TB bacilli in ZN stained sputum images for an automated TB diagnostic method.

A GPU-customized visual hull reconstruction algorithm for real-time applications
Yuko Roodt and Willem A. Clark

In this paper we present a Graphics Processing Unit (GPU)-based method for reconstruction of a volumetric scene taken from known but arbitrarily distributed camera views-points. This novel approach allows for efficient parallelization and distributed processing of the reconstruction algorithm. We further extend this implementation by calculating the reconstructed hull's volume. The Space carving algorithm is evaluated for accuracy and speed.

A Shader-based GPU Implementation of the Fast Fourier Transform
Philip E Robinson and Willem A Clarke

Image processing technology is maturing at a rapid rate, but the classic processing platforms available on which to perform image processing are still not powerful enough to allow for the real-world implementation of many of these techniques. Technologies like the Field-Programmable Gate Arrays (FPGA) are expensive and difficult to develop for and as such do not provide a practical solution to this problem as yet. GPU's however are quickly outstripping the more standard CPU architecture as powerful parallel processors which are well suited to image processing techniques. This paper describes the implementation of the Fast Fourier Transform, a fundamental building block of many image processing algorithms, on the GPU by making use of shader technology. A performance comparison between various GPU's and CPU implementations of the Fast Fourier Transforms is also provided.

A Readability Formula for Afrikaans
Cindy A. McKellar

This paper is about the development of a Readability formula for Afrikaans. It discusses the collection and processing of the data needed to calculate the formula coefficients. A number of different formulas were developed to see which of the variables gave the best indication of the readability of a document. It was found that sentence length, word length measured in syllables, the number of familiar words (words found in a wordlist), the number of brackets and the amount of symbols and numbers gave the best formula. This formula was then evaluated and compared to two existing readability formulas for Afrikaans.

Assessing the impact of missing data using computational intelligence and decision forest
Donghyun Moon and Tshilidzi Marwala

Autoencoder Neural Network is implemented to estimate the missing data. Genetic Algorithm (GA) is implemented for network optimization and estimating the missing data. Missing data is treated as Missing At Random (MAR) by implementing maximum likelihood algorithm. The network performance is determined by calculating the network's Mean Square Error (MSE). The network is further optimized by implementing Decision Forest (DF). The impact of missing data is then investigated on both ANN-GA and ANN-GA-DF network.

Effects of the Type of Missingness of Data on Artifical Intelligence Predition
D. A. Braude

In surveys data often goes missing. While many techniques exist to combat this problem, a recent proposal was for a system that is composed of a neural network and a genetic algorithm has been suggested. A design example for a prediction based on this system is given. It uses the results of an HIV survey. The effect of the choice of activation function and the type of missing data was examined. The tests show that Gaussian activation functions are the best choice for radial basis function neural networks. The type of missing data has little impact on the accuracy of the prediction.