Fast Region-based Object Detection and Tracking using Correlation of Features

Fred Senekal
Mobile Intelligent Autonomous Systems (MIAS)
Council for Scientific and Industrial Research (CSIR)
Pretoria, South Africa
Email: fsenekal@csir.co.za

***Abstract*—A new method for object detection using region-based characteristics is proposed. The method uses correlation between features over a region as a descriptor for the region. It is shown that this region descriptor can be successfully applied to object detection and tracking problems. An attractive property of the method is that region characterisation, region matching and localisation can be done sufficiently fast to use the method in a real-time system.**

I. Introduction

Visual target tracking refers to the ability of a system to detect and track a target object (or objects) over a series of digital images. Visual target tracking can be accomplished by feature-based or region-based approaches.

In *feature-based approaches*, interest points are calculated in a digital image, and a local region descriptor is calculated at each interest point. Once a set of descriptors is calculated for a particular object, the object can be tracked over various images by comparing and matching descriptors calculated over the images. Methods such as the Scale-invariant Feature Transform (SIFT) [1] and Speeded Up Robust Features (SURF) [2] are popular and are successfully applied to the problem. Although these methods are successful, they suffer from being computationally expensive, impacting negatively on their suitability for implementation in real-time systems. They are also affected by motion blur, which make it difficult to reliably extract interest points for further computation.

In *region-based approaches*, an image is segmented into different regions, typically corresponding to different surfaces of an object. A feature is calculated based on the region-based characteristics of surfaces associated with the target object. These features are then matched across different images, to determine the regions corresponding to the target object. Region-based approaches will be explored further in this article as an alternative or complementary approach to feature-based approaches.

Any region-based tracking method generally relies on the specification of three components:

- *Region characterisation.* This is the way in which a given region is abstracted mathematically.
- *Region matching.* The similarity or dissimilarity between two region characterisations needs to be quantified in order to determine the best-matching region and/or whether a positive detection could be made.
- *Localisation.* Even if a region could be characterised and matched to other regions, a way is sought to select the best-matching region from the large number of region configurations in an arbitrary image in a computationally feasible way.

A. Background

The essence of any region-based detection algorithm is the way in which the region is characterised. A mathematical description or model is sought that describes the visual characteristics of the region to the extent that it can be used for higher-level purposes such as tracking or recognition.

A natural starting point is to use the basic region statistics such as colour, intensity or gradient information ([3], pp. 90-99). These approaches work well in cases where the target object or region can be engineered to exhibit certain characteristics. These approaches are commonly found in manufacturing-type environments where the objects being manufactured typically have certain visual characteristics and where the environmental variables such as lighting and camera position can be controlled as well. In the work conducted here, a method is sought that can be applied in arbitrary situations. In such situations, there may be considerable variation in the visual characteristics of the object that should be tracked and in the environmental conditions.

In a general situation, the object that should be tracked might have variations in the colour, intensity, gradient or other low-level features. To characterise these variations, histogram methods are often used [4]. In histogram-based approaches, the range of every feature variable is divided into several bins. The feature vector associated with every pixel is associated with a set of indices into the bins. An object is characterised by calculating the number of times a particular bin-combination is obtained. These count values are often normalised, in which case the histogram is a nonparametric estimation of the joint distribution of the features. Methods such as integral histograms [5] have been devised to speed up computation of histograms over regions. As noted in [6], one problem with histogram-based methods is that they are computationally exponential in the number of features.

Another way in which regions are often characterised is by expressing them through their texture properties. The most common approach for calculating region texture is through the

use of filter banks ([7], pp. 191-196). An image is convolved with a set of filters, often sensitive to local structures such as spots or bars at different scales. The set of filter responses associated with every pixel is then used as a feature vector associated with that pixel. Various filter banks have been designed, such as the Leung-Malik filters [8], Schmid filters [9] and Maximum-Response filters [10]. The same approaches used for other low-level features can then be applied to the texture features. A common way to proceed is to build a universal dictionary of different types of texture responses by clustering together similar texture responses [11]. A region is then characterised by calculating a histogram over the visual dictionary.

Tuzel et al [6] presents a method to characterise a region based on region covariance. In the method, a set of features is calculated for each pixel. The covariance of the features over a region is used as a descriptor for the region. The authors found that the method outperforms methods based on the calculation of histograms of features. The method presented in this paper builds on Tuzel's approach.

II. Correlation-based Detection

The new method based on correlation-based detection is introduced in this section.

A. Region Characterisation

To characterise a region, a d-dimensional feature vector is calculated for every pixel in the region. Following [6], a 9-dimensional feature vector composed of the x and y coordinates of the pixel, the three colour components (red, green and blue) and the first and second order derivatives of the intensity of the pixel in both the x and y dimensions is calculated. Let $\overline{f}(x,y)$ be the feature vector associated with pixel position (x, y). Then $\overline{f}(x,y) = [x, y, r(x,y), g(x,y), b(x,y), \frac{di(x,y)}{dx}, \frac{di(x,y)}{dy}, \frac{d^2i(x,y)}{dx^2}, \frac{d^2i(x,y)}{dx^2}]$, where r, g, b and i denote the red, green, blue and intensity values of the pixel. More specifically, the first order derivatives are calculated by convolving the image intensities with a filter with kernel [-1 0 1] in both the x and y dimensions. The second order derivatives are calculated by convolving the image intensities with a filter with kernel [-1 2 -1] in both the x and y dimensions. Although this specific feature vector is used in the experiments, the technique can be applied to any arbitrary d-dimensional feature vector.

Once the feature vectors are calculated for every pixel in the region, the region is characterised by calculating the $d \times d$ correlation matrix P_R over the feature vectors corresponding to the region R. The $(i,j)^{th}$ entry of P_R is given by the Pearson product-moment correlation coefficient, calculated as

$$\rho_{i,j} = \frac{\frac{1}{N}\sum_{(x,y)\in R}(f_i(x,y)-\mu_i)(f_j(x,y)-\mu_j)}{\sigma_i\sigma_j}, \quad (1)$$

where

$$\mu_i = \frac{1}{N}\sum_{(x,y)\in R} f_i(x,y) \quad (2)$$

is the mean of feature i over the region,

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{(x,y)\in R}(f_i(x,y)-\mu_i)^2} \quad (3)$$

is the standard deviation of feature i over the region and N is the number of pixels in the region.

Since the correlation matrix is symmetric ($\rho_{i,j} = \rho_{j,i}$) and the diagonal entries equal to one ($\rho_{i,i} = 1$), only the non-diagonal upper-triangular entries of the matrix are necessary to characterise the region. Thus, only $\frac{d^2-d}{2}$ values need to be calculated for the region descriptor. In the case that $d = 9$, only 36 values are calculated.

It is useful to compare the above approach with that of Tuzel et al [6]. In Tuzel's method, the covariance matrix C_R calculated over the features in a region is used as the region descriptor. The covariance values are related to the correlation values through

$$\rho_{i,j} = \frac{c_{i,j}}{\sigma_i\sigma_j} = \frac{c_{i,j}}{\sqrt{c_{i,i}c_{j,j}}}. \quad (4)$$

The covariance values are unbounded and could have arbitrarily large or small values depending on the range of the features used. The correlation values are restricted to the interval [-1, 1]. The correlation values can be viewed as a normalisation of the covariance values through the product of the standard deviations. This normalisation makes direct comparison between correlation values possible, which may not be the case for the covariance values (consider for example the case where the coordinates of the pixel are used as features and where regions in different parts of the image are considered). It is also noted that Tuzel's method requires $\frac{d^2+d}{2}$ values to characterise a region. The proposed method based on correlation requires d values less.

Similarly to Tuzel's method, the correlation matrix do not retain information pertaining to the ordering and number of pixels, which implies a certain scale and rotation invariance (depending on the design of the feature vector).

B. Region Matching

Given that a region can be characterised using correlation matrices, a method is sought by which two such characterisations can be compared. The simmilarity or dissimilarity between two regions is expressed by the use of a distance function. The distance function is used to determine the best-matching region to a target region and also to determine whether a positive detection can be made.

Since the correlation values are normalised, corresponding correlation values in two matrices can be directly compared. Given two correlation matrices P_{R_1} and P_{R_2} over regions R_1 and R_2 respectively, the Euclidean distance given by

$$dist(P_{R_1}, P_{R_2}) = \sqrt{\sum_{i=1}^{d}\sum_{j=i+1}^{d}(\rho_{1,i,j}-\rho_{2,i,j})^2} \quad (5)$$

is a reasonable choice. Note that only the non-diagonal upper-triangular values are compared, since the matrices are symmetric and the differences between diagonal entries would be zero.

The distance function could be further modified by introducing a weight $w_{i,j}$ associated with the $(i,j)^{th}$ entries in the correlation matrices. These weights could be optimised to express which of the product-moment terms are more important relative to others. In the experiments conducted here, no such weights were used.

In Tuzel's method [6] the corresponding covariance values cannot be directly compared. The method requires calculation of the generalised eigenvalues between the two covariance matrices being compared. A distance measure proposed in [12] is then used to compare the dissimilarity of the covariance matrices. The distance measure is given by

$$dist(C_{R_1}, C_{R_2}) = \sqrt{\sum_{i=1}^{d} ln^2 \lambda_i(C_{R_1}, C_{R_2})}, \tag{6}$$

where the $\lambda_i(C_{R_1}, C_{R_2})_{i=1..d}$ are the generalised eigenvalues of C_{R_1} and C_{R_2}. The computational complexity lies in calculating the generalised eigenvalues λ_i. Algorithms such as the QZ algorithm [13] can be used to solve the generalised eigenvalue problem in $O(d^3)$ arithmetic computations using numerical methods. The iterative nature of the QZ (and similar) algorithms is however a practical drawback and is further compounded by the fact that a number of region comparisons need to be performed per image. The use of the correlation rather than covariance values makes it possible to avoid the computationally expensive generalised eigenvalue calculations.

C. Localisation

Although a region can be characterised and region characterisations meaningfully compared, there still remains the problem of determining which regions to compare to the target region. An object could have any shape and thus have any arbitrarily-shaped region projection in an image. Ideally all region configurations should be evaluated. The computational complexity in selecting and evaluating all arbitrarily-shaped regions in an image make such an approach infeasible.

The standard approach is to restrict regions to be evaluated to rectangular regions. A "moving window" is then applied accross the image at different scales and a brute-force search is performed. The standard approach is adopted here; however, a significant computational speed increase is achieved through the application of integral images [14], which make it possible to compute the correlation in any rectangular region in constant time.

Given a rectangular arrangement of values (such as an image), an integral image is simply the sum of the values in the rectangle bounded by the upper left corner and the coordinate of interest. More precisely, given values $I(x,y)$, the integral image $II(x,y)$ at position (x',y') is given by

$$II(x',y') = \sum_{x<x',y<y'} I(x,y). \tag{7}$$

An integral image can be computed in a single pass through the matrix of values, as shown in [14]. Given a rectangle with upper left coordinate (x_1,y_1) and lower right coordinate (x_2,y_2), the sum of the values in the rectangle can be computed in constant time by using the integral image (the operator L is introduced here as a shorthand notation for the sum over the rectangle):

$$\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} I(x,y) = L(II, x_1, y_1, x_2, y_2)$$
$$= II(x_2,y_2) + II(x_1,y_1) - II(x_2,y_1) - II(x_1,y_2). \tag{8}$$

To speed up the brute force search through an image, d feature integral images and $\frac{d^2+d}{2}$ product-of-feature integral images are pre-computed for the image. Given that features $\overline{f}(x,y)$ have been calculated for the image, the feature integral images are calculated as

$$IF_i(x',y') = \sum_{x<x',y<y'} f_i(x,y), \tag{9}$$

for $i \in [1,d]$. The product-of-feature integral images are calculated as

$$IP_{i,j}(x',y') = \sum_{x<x',y<y'} f_i(x,y) \times f_j(x,y), \tag{10}$$

for $i \in [1,d]$ and $j \in [i,d]$.

Calculation of the correlation matrix over a rectangular region with upper left coordinate (x_1,y_1) and lower right coordinate (x_2,y_2) proceeds as follows. First, the sum of the individual feature values over the region is calculated:

$$F_i = L(IF, x_1, y_1, x_2, y_2), \forall i \in [1,d]. \tag{11}$$

Thereafter, the sum of the product-of-feature values over the region is calculated:

$$P_{i,j} = L(IP, x_1, y_1, x_2, y_2), \forall i \in [1,d], j \in [i,d]. \tag{12}$$

The covariance terms in the region is calculated as

$$c_{i,j} = \frac{1}{N_R}(P_{i,j} - \frac{1}{N_R}F_iF_j), \forall i \in [1,d], j \in [i,d]. \tag{13}$$

where $N_R = (x_2 - x_1 + 1)(y_2 - y_1 + 1)$ is the number of pixels in the region. Finally, the correlation terms are calculated through (4).

III. Tracking System

To test the correlation-based detection method introduced in this paper, it was implemented as part of a tracking system as part of the CSIR MULE robot project developed at MIAS. A brief overview of the tracking system is given in this section. The design of the tracking system is shown in Figure 1.

First, the image dimensions are reduced from $W \times H$ pixels to $w \times h$ pixels through bilinear interpolation. The resampling

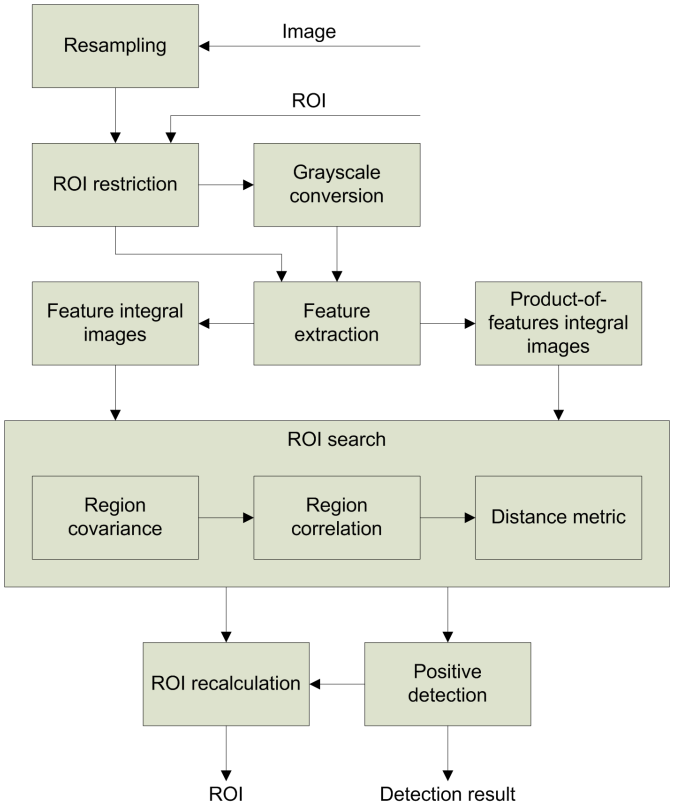


Fig. 1: Region-based visual target tracking system block diagram

is done to limit the computational requirements for further processing, the idea being that the parametres w and h could be adjusted up or down to achieve real-time performance based on the computational power that is available on a specific platform.

To further reduce the computational requirements, processing is restricted to the region of interest (ROI). The region of interest is provided as an input parameter to the system. The region of interest is specified as a rectangular region. The coordinates of the ROI in the resampled image are calculated and all subsequent processing restricted to the ROI in the resampled image.

The intensity values of pixels in the ROI are calculated to facilitate computation of the first and second order derivatives. Thereafter the feature vectors are extracted over the pixels in the ROI. Feature and product-of-features integral images are calculated based on the extracted features. Note that to further increase computational speed, the features and integral images can be calculated and stored as integer values.

A brute-force search over the region of interest is now applied using the computed integral images. A moving window is applied over the ROI at various scales. The aspect ratio of the window is kept the same as the aspect ratio of the target region. For each window, the region covariance and region correlation are calculated. The distance to the target region is calculated. The region with the lowest distance score is kept as a potential candidate for the target object. Since only a minimum distance is sought, the square root in the distance metric never need to be explicitly evaluated.

Finally, if the minimum distance value is within some threshold, a positive detection was made; otherwise, the system indicates that no detection was made. If a positive detection was made, an output ROI is calculated. In the absence of a model of the dynamic behaviour of the target object, the output ROI is centred on and has twice the width and height of the search window with the best score. The width and height is adapted to fit into the boundaries of the window, in the case the left, right, top or bottom of the ROI would overflow the boundaries of the image. If no detection is made, the output ROI is set equal to boundaries of the image.

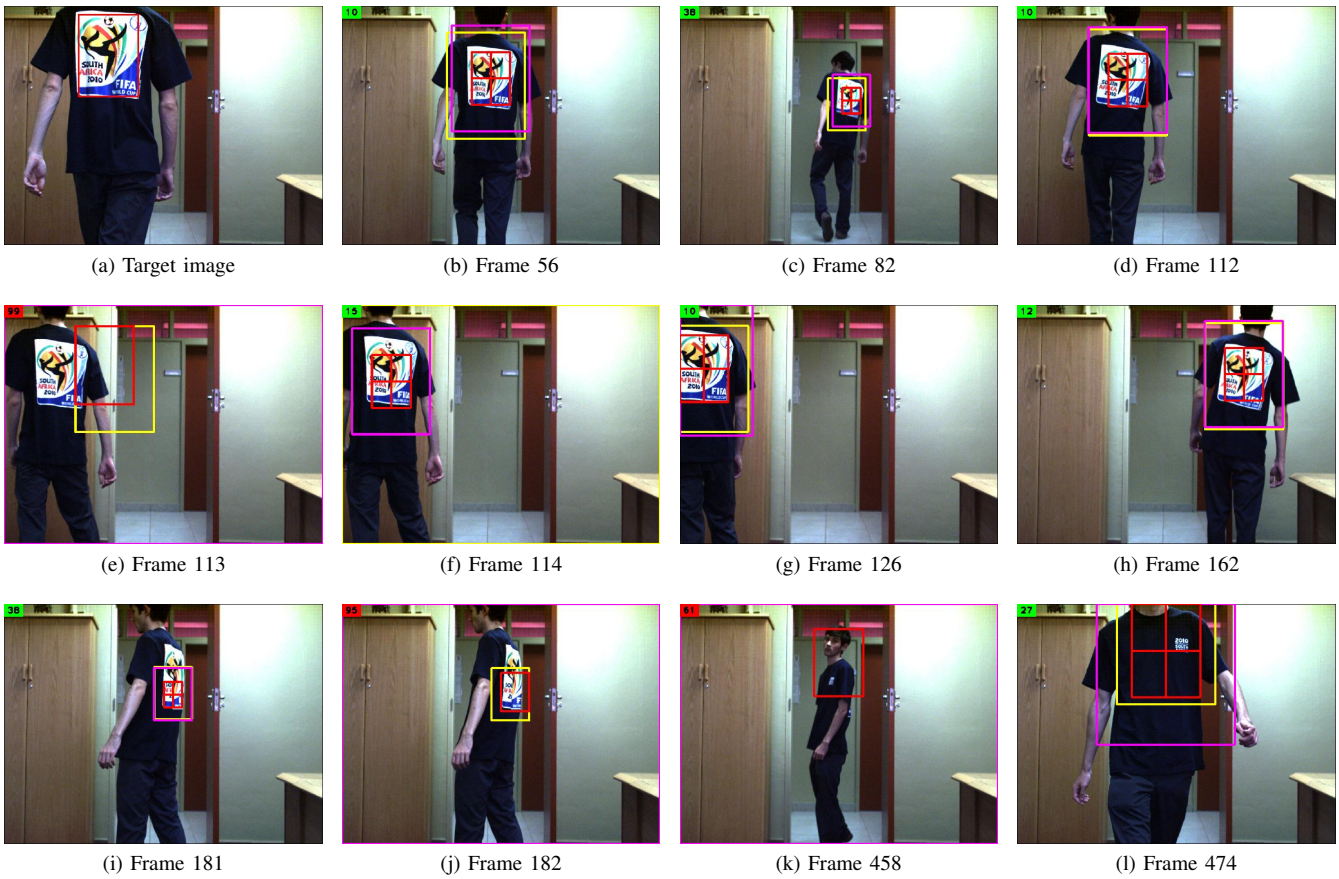
IV. Results and Discussion

There is often difficulty in quantifying the results obtained by tracking algorithms when applied to video sequences. Firstly, such video sequences are not readily available. Secondly, the success of a particular tracking algorithm is often dependent on the application for which it is designed, which could bring into question the worth of the measure when applied to a video sequence pertaining to a different application. Thirdly, the actual quantification depends to a large extent on the specific video sequence. In many cases, the quantified success of the algorithm (such as positive detection rates) could be artificially improved by including more video footage for which the algorithm performs well.

For the above reasons, the results obtained by the algorithm will be discussed from a qualitative perspective based on two typical tracking scenaries encountered in the CSIR MULE project.

Fig. 2 shows selected frames from a video sequence where the objective was to track the insignia on the back of a shirt. In this experiment, the output ROI of one frame is used as the input ROI for a subsequent frame. Fig. 2a shows the target image that need to be tracked. The detection is successful (Fig. 2b), even under severe scale changes (Fig. 2c), partial occlusions (Fig. 2g), slight rotation (Fig. 2h) and aspect ratio modification (Fig. 2i). The output ROI from Fig. 2d is used as the input ROI for Fig. 2e; however, the target has moved out of the ROI and the system indicates that no detection is made. The ROI is reset to the entire frame and the system is able to resume detection in Fig. 2f. Fig. 2j shows an example of a false negative for a challenging image. Fig. 2k shows an example of a true negative detection and Fig. 2l an example of a false positive detection.

Fig.3 shows selected frames from a video sequence where to objective was to track a cereal box. The cereal box was moved about erratically, to introduce motion blur. ROI feedback was not used for this experiment. Fig. 3a shows the target image that need to be tracked. The detection is successful (Fig. 3b), even with motion blur (Fig. 3c and Fig. 3e), partial occlusions (Fig. 3d), slight rotations (Fig. 3f) and out-of-plane rotations (Fig. 3g and Fig. 3j). There are however some instances where the algorithm fails, such as the out-of-plane rotation in Fig. 3h,



(a) Target image (b) Frame 56 (c) Frame 82 (d) Frame 112

(e) Frame 113 (f) Frame 114 (g) Frame 126 (h) Frame 162

(i) Frame 181 (j) Frame 182 (k) Frame 458 (l) Frame 474

Fig. 2: Results of the tracking algorithm on the shirt video sequence. Legend: Yellow - input ROI, Magenta - output ROI, Red - best-matching region, Green/Red Box - detection/non-detection with associated score. (a) Target image, (b) True positive detection, (c) True positive detection under large scale change, (d) True positive detection (note output ROI), (e) False negative failure since the target has moved out of the ROI predicted in (d), (f) The failure in (e) reset the ROI and detection succeeded again, (g) True positive detection under partial occlusion, (h) True positive detection under slight rotation, (i) True positive detection under aspect ratio modification, (j) False negative failure due to severe aspect ratio modification, (k) True negative detection, (l) False positive detection.

partial occlusion in Fig. 3i and the false positive detection in Fig. 3k. Fig. 3l shows an example of a true negative detection.

The method is robust against scale changes. It is robust against small changes in rotation, but fails under larger rotation due to changes in the correlation coefficients. Failures under out-of-plane-rotations can be explained as the result of the system trying to maintain the original aspect ratio of the enrolled target image, which becomes severely distorted. The occurance of false positives can be reduced by dynamically adjusting the detection threshold.

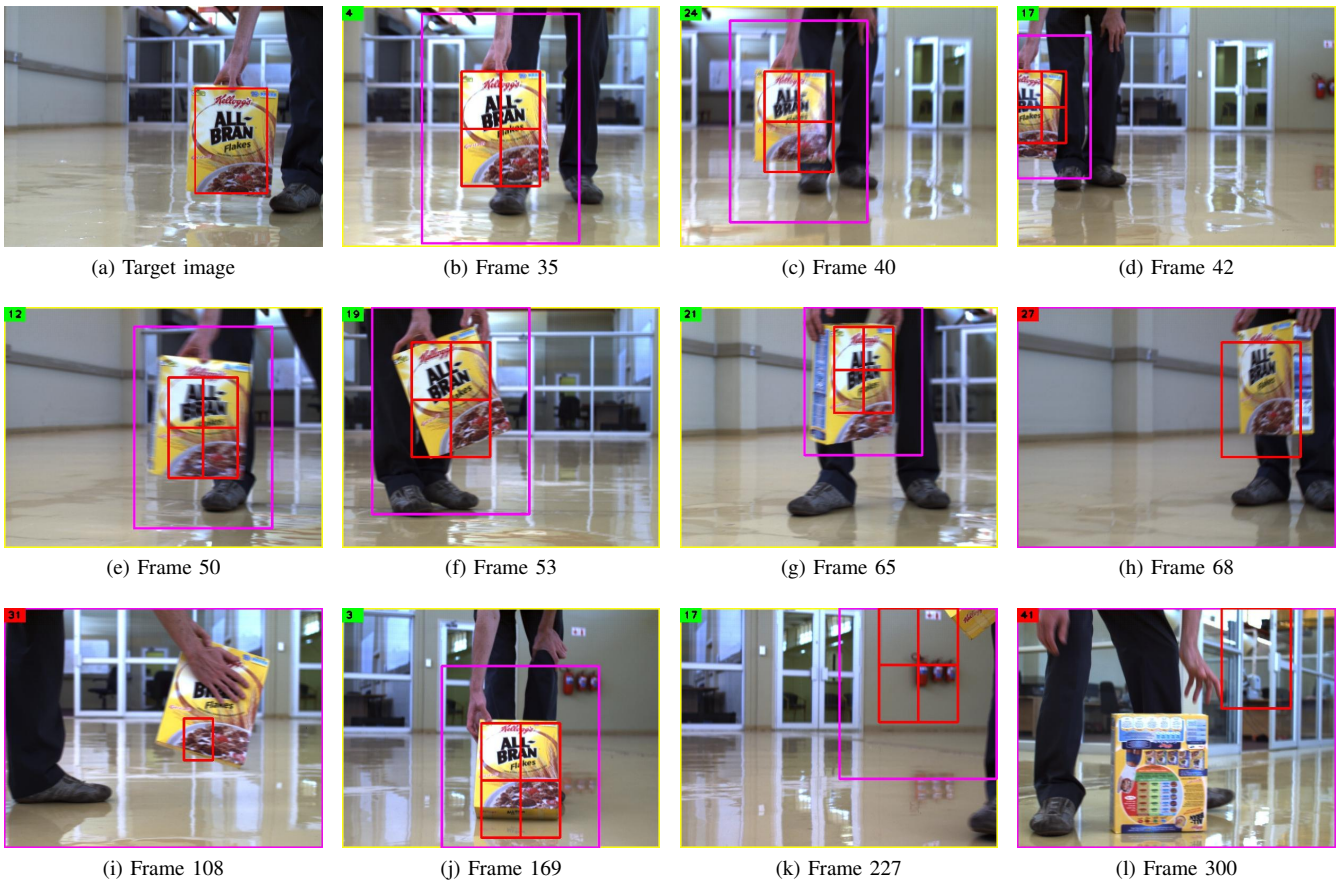
Experiments were conducted on a Duel Core Pentium D 3.00GHz (using a single core) with 2GB RAM, running Ubuntu 10.04. The images in the shirt video sequence were of dimensions 1024×768 pixels and in the cereal box video sequence 900×680 pixels. Images were scaled to 320×240 pixels by the system using binlinear interpolation. For localisation, 12 different scales from 10 pixels to 120 pixels in increments of 10 pixels were searched with a step size of 5 pixels at each scale. The average detection time (including bilinear interpolation) to process an image when restricted to the ROI was 25ms (40fps). In the case that the entire image was processed, the average detection time was 259ms (3.8fps). The system can be further improved by incorporating a model of the target behaviour, in order to further restrict the scales at which the search is conducted.

V. Conclusion

A new method for region-based detection and tracking of objects was presented. The method is based on using the correlation between features over the region as a region descriptor. The method is fast enough to be implemented as part of a real-time system, yet delivers satisfactory detection results.

Acknowledgment

This work was funded by the Council for Scientific and Industrial Research, South Africa, as part of the MIAS Au-



(a) Target image (b) Frame 35 (c) Frame 40 (d) Frame 42

(e) Frame 50 (f) Frame 53 (g) Frame 65 (h) Frame 68

(i) Frame 108 (j) Frame 169 (k) Frame 227 (l) Frame 300

Fig. 3: Results of the tracking algorithm on the cereal box video sequence. Legend: Yellow - input ROI, Magenta - output ROI, Red - best-matching region, Green/Red Box - detection/non-detection with associated score. (a) Target image, (b) True positive detection, (c) True positive detection under motion blur, (d) True positive detection under partial occlusion, (e) True positive detection under severe motion blur, (f) True positive detection under slight rotation, (g) True positive detection under out-of-plane rotation, (h) False negative detection, (i) False negative detection under partial occlusion, (j) True positive detection, (k) False positive detection, (l) True negative detection.

tonomous MULE project. The author would like to thank Deon Sabatta and Michael Burke for their ideas and suggestions. The author is grateful to Michael Burke who kindly supplied the datasets used in the experiments.

REFERENCES

[1] D.G. Lowe, "Object recognition from local scale-invariant features", in *Proceedings of the International Conference on Computer Vision*, pp. 1150-1157, 1999.

[2] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded up robust features", in *Proceedings of the European Conference on Computer Vision*, pp. 404-417, 2006.

[3] B.K.P. Horn, *Robot Vision*, The MIT Press, Cambridge, 1986.

[4] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, Volume 1, pp. 142-149, 2000.

[5] F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, Volume 1, pp. 829-836, 2005.

[6] O. Tuzel, F. Porikli and P. Meer, "Region covariance: a fast descriptor for detection and classification", *European Conference on Computer Vision*, May 2006.

[7] D.A. Forsyth and J. Ponce, *Computer Vision - A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2003.

[8] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons", *International Journal of Computer Vision*, Volume 43, Issue 1, pp. 29-44, 2001.

[9] C. Schmid, "Constructing models for content-based image retrieval", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 39-45, 2001.

[10] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images", *International Journal on Computer Vision*, Volume 62, pp. 61-81, 2005.

[11] J. Winn, A. Criminisi and T. Minka, "Object categorization by learned universal visual dictionary", in *Proceedings of the IEEE International Conference on Computer Vision*, Beijing, 2005.

[12] W. Förstner and B. Moonen, "A metric for covariance matrices", Technical report, Department of Geodesy and Geoinformatics, Stuttgart University, 1999.

[13] C. Moler and G.W. Stewart, "An algorithm for generalized matrix eigenvalue problems", *SIAM Journal on Numerical Analysis*, Volume 10, pp. 241-256, 1973.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Proceedings of the European Conference on Computer Vision*, Copenhagen, Denmark, 2002.