Stationary region predictor using a stationary camera

B.Z de Villiers*, Y. Roodt†, H. Roos†, Prof. W.A Clarke†,
HyperVision Research Lab
University of Johannesburg
South Africa
*Email: wheres.brett.dev@gmail.com
†Email: {yroodt,hroos,wclarke}@uj.ac.za

Abstract—**A method to determine the stationery probability of regions or feature points in a video sequence is proposed in this paper. This is done by identifying feature points using the Harris corner detector, finding descriptors for the feature points and then tracking the feature points. The information gained from tracking the feature points is then used to determine the stationery probability of these features. This method is shown to successfully identify probable stationery and moving regions in video sequences.**

I. Introduction

Motion estimation of local regions is useful in a number of applications such as security in surveillance systems, long range video surveillance and forest fire detection. A number of methods were proposed to detect stationary foreground objects by performing background estimation [1-3], when an object forms part of the estimated background the object is determined to be a stationary foreground object. A number of methods based on a filtering approach have been proposed to model and construct an estimated background. The filters used in these methods include the median filter [3], Wiener filter [4], minimum-maximum filter [5], $\Sigma - \Delta$ filter [6], single Gaussian filter [7] and Kalman filtering [8]. These methods do however show a lack of robustness to change in illumination, cluttered environments and non-stationary backgrounds. To improve the robustness, methods using a number of different filters have been proposed [1, 2, 9, 10], these methods have shown robust performance to change in illumination, cluttered environments and non-stationary backgrounds.

We propose a method to determine the stationary probability of regions or features points in a video sequence. Regions or feature points with a high stationary probability can be considered being part of the background of a scene and regions or feature points with a low stationary probability as part of the foreground. This was accomplished by designing a tracking algorithm to identify and track feature points in the video sequence. The change in pixel position of the feature points was used to determine the stationary probability of the feature points. This method provides probabilistic information on whether a feature or region in a video sequence forms part of the foreground or the background. Since this method uses feature tracking it could also perform background estimation for a moving video camera, for this paper however only the use of a stationary camera was investigated.

In Section II we will discuss the Harris corner detector, which is used to detect the feature points. Section III provides a discussion on methodology including a discussion on the descriptor used for matching features, the matching algorithm, the tracking algorithm and the algorithm which determines the stationary probability of the feature points. Section IV provides the results from the discussed system and Section V provides a discussion on the performance of the stationery predictor algorithm.

II. Theory

A. Harris corner detector

The Harris Corner detector is a method for determining corner responses in images [11, 12]. The Harris Corner detector has shown some invariance to rotation, different methods of sampling and quantization and scale transformations. Corner detection finds sharp corner responses in images, where the positions of the sharp corner responses are easily identified [11]. The Harris Corner Detector extracts the eigenvalues of the Hessian matrix M to determine the curve response.

$$H = \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{1}$$

$$M = \sum_{x,y} w(x, y)H \tag{2}$$

- Where $w(x, y)$ a windowing function which performs a Gaussian blur

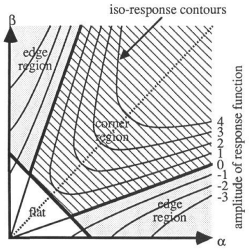


Fig. 1. Corner response where β refers to λ_1 and α refers to λ_2 [11]

- Where I_x contains the partial derivatives with respect to x of the input image I
- Where I_y contains the partial derivatives with respect to y of the input image I

The relationship between the eigenvalues λ_1 and λ_2 for the Hessian matrix M describes the curvature a surface in the input image. A corner exists at the pixel position where both λ_1 and λ_2 are large as can be seen in Figure 1. It was found that the exact computation of the eigenvalues was computationally expensive, due to this the corner metric R is calculated. The corner metric determines the relationship between the eigenvalues λ_1 and λ_2 for the Hessian matrix.

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{3}$$

$$= det(M) - k(trace(M))^2 \tag{4}$$

- Where k is a sensitivity factor ($0.04 \leq k \leq 0.06$), the smaller the value of k the more likely the Harris Corner detector can detect sharp corners

The corner metric R then determines the corner response of an input image. Non maximum suppression is used to convert the corner metric into a binary image where the positions of the corners or feature points are easily identifiable. The Harris corner detector determines the pixel positions of corners in images, these corners are used as feature points for the algorithm developed in this paper.

B. Sobel operator

The Sobel operator is a method for determining the partial derivatives as well as the angles of the partial derivatives in an image. The Sobel operator is used in the descriptor algorithm to determine the magnitudes and the angles of the gradients surrounding the features points. The Sobel operator is described by the following equations and kernels [13].

$$G_x = A \bigotimes I \tag{5}$$

$$A = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{6}$$

$$G_y = B \bigotimes I \tag{7}$$

$$B = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \tag{8}$$

- Where G_x is an array which approximates of the partial derivative in the x direction
- Where G_y is an array which approximates of the partial derivative in the y direction
- Where I is the input image

$$\Theta = \tan^{-1}(G_y/G_x) \tag{9}$$

- Where Θ is an array which contains the angles of the derivatives

III. Methodology

A. Stationery predictor algorithm overview

To determine the probability of regions in a video sequence being stationary for the next few frames of the video sequence it is necessary to determine how the objects in the video sequence have moved. To do this a tracking algorithm was designed to determine how features or objects in the video sequence have moved during the observed frames of the video sequence. The Harris corner detector was used to determine features to be tracked in the video sequence frames. A descriptor was designed to describe and help match features between the frames of the video sequence. The changes in position (from frame to frame) of the features was recorded using the tracking algorithm and this information was used to determine the stationary probability of the features. The stationary probability of the features was used to estimate the stationary probability of the regions in the video sequence. Figure 2 shows a basic overview of the stationery predictor algorithm.

The following describes how the tracking, feature detection and description algorithms interlink with each other.

1) Acquire frame(n) from video sequence
2) Detect feature points in frame(n) and add these features to the feature list
3) Determine description vectors for the features in frame(n)
4) Acquire frame($n+1$) from video sequence
5) Detect features points in frame($n+1$)
6) Determine description vectors for the features in frame($n+1$)
7) For feature i_n (where i_n = 1, 2, 3, ..., N-1, N; N is the number of features in frame(n) and i_n is a feature from frame(n)) search for features in a 20 × 20 window in frame($n+1$) (the position of the window is the predicted position of feature $i_{(n+1)}$ in frame($n+1$))

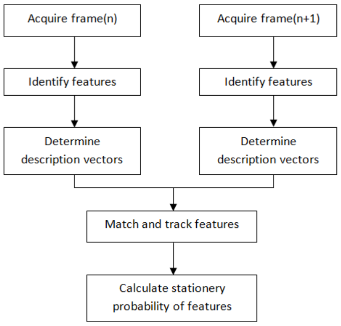

Acquire frame(n)
Acquire frame(n+1)
Identify features
Identify features
Determine description vectors
Determine description vectors
Match and track features
Calculate stationery probability of features

Fig. 2. Overview of Stationary predictor algorithm

8) Compare the description vector for feature i_n with the description vectors for the features $j_{(n+1)}$ found in the window in frame($n + 1$) and determine the best match
9) Determine whether the best match for feature i_n is an inlier or outlier using a tolerance factor
10) If the best match for feature i_n is an outlier then remove feature i_n from the feature list
11) If feature i_n is an inlier determine the x and y velocities for the feature i_n
12) Determine the component in the change matrix $X_{i,n}$ for the feature i_n
13) Update the description vector for feature i_n with the description vector of the best matched feature from frame($n + 1$)
14) Repeat steps 7 to 13 for $i_n = 1, 2, 3, ..., N$-1, N
15) Add new features to the list - Compare the positions of the matched features with the positions of the features in frame($n+1$), if some of the positions of the features from frame($n + 1$) are not equal to any of the positions of the matched features then these features are described as new features
16) Let the x and y velocities for new features equal zero and determine the description vectors for the new features
17) For frame index(n); $n = n + 1$
18) Repeat from step 4

B. Feature descriptor

The descriptor used in the tracking algorithm was designed to be partially illumination and rotation invariant. It was inspired by the descriptor used in the SIFT algorithm [14]. Partial derivatives are found in a $w \times w$ window, where w is determined by the size of the input image and the window is centered around the pixel position of a feature point. The Sobel operator was used to determine the partial derivatives in the local window surrounding the feature point.

Instead of calculating $\tan^{-1}$ the signed ratio G_y/G_x of the derivatives G_x and G_y where used to determine the angles of the derivatives of the pixels in the $w \times w$ window surrounding the feature point of interest. This was done to improve the computational performance of the algorithm. A histogram is created containing a Gaussian weighting of the angles for the derivatives of the pixels in the $w \times w$ window surrounding the feature point of interest. The Gaussian weighting is based on the distance of the pixels from the feature point of interest such that pixels further away from the feature point have a smaller contribution to the histogram. The histogram contains 16 bins and the value of sigma for the Gaussian weighting is based on the size of the $w \times w$ window such that.

$$\sigma = w/10 \tag{10}$$

The values in the 16 bins of the histogram are used for the 16 elements of the description vectors of the feature points.

The above describes the primary section of the descriptor. The primary section of the descriptor was designed to extract description information surrounding a feature where the area surrounding the feature of interest contains high frequency information. The RGB pixel information of the feature and surrounding pixels was also used as description information, this was done to better describe features which occur in low frequency areas in an image. To do this the RGB pixel information of the pixels in the $w \times w$ window surrounding the feature point of interest was averaged using the same Gaussian weighting.

C. Matching process

In the matching process the description vector of the feature of interest (feature i_n) from frame(n) in the video sequence is compared to the description vectors of a number of features (features $j_{(n+1)}$) from frame($n + 1$). The features from frame($n + 1$) which are compared to the feature of interest from frame(n) are found in a $T \times T$ window where T is determined by the size of the input image and the where the $T \times T$ window is centered around the predicted position in the frame($n+1$) of the compared feature. The following equations describe the algorithm used in the matching process.

$$C_j = \sqrt{(\frac{R_{i_n} - R_{j_{(n+1)}}}{b})^2 + (\frac{G_{i_n} - G_{j_{(n+1)}}}{b})^2 + (\frac{B_{i_n} - B_{j_{(n+1)}}}{b})^2} \tag{11}$$

$$K_j = \sqrt{(L_{i_n} - L_{j_{(n+1)}})^2 + (\frac{x_{i_n} - x_{j_{(n+1)}}}{c})^2 + (\frac{y_{i_n} - y_{j_{(n+1)}}}{c})^2} \tag{12}$$

$$D_j = \sqrt{(C_j)^2 + (K_j)^2} \tag{13}$$

- Where R_{i_n} is the average R value in the $w \times w$ window for feature i_n and $R_{j_{(n+1)}}$ is the average R value in the $w \times w$ window for feature $j_{(n+1)}$

- Where G_{i_n} is the average G value in the $w \times w$ window for feature i_n and $G_{j_{(n+1)}}$ is the average G value in the $w \times w$ window for feature $j_{(n+1)}$
- Where B_{i_n} is the average B value in the $w \times w$ window for feature i_n and $B_{j_{(n+1)}}$ is the average B value in the $w \times w$ window for feature $j_{(n+1)}$
- Where L_{i_n} is the vector for the primary section of the descriptor for the feature i_n and $L_{j_{(n+1)}}$ is the vector for the primary section of the descriptor for the feature $j_{(n+1)}$
- Where x_{i_n} is the predicted x co-ordinate of the the feature i_n in the frame(n+1) and $x_{j_{(n+1)}}$ is the x co-ordinate of the the feature $j_{(n+1)}$ in the frame$(n+1)$
- Where y_{i_n} is the predicted y co-ordinate of the the feature i_n in the frame(n+1) and $y_{j_{(n+1)}}$ is the y co-ordinate of the the feature $j_{(n+1)}$ in the frame$(n+1)$
- Where b and c are constants where both b and c are greater than zero, these constants are used to scale in the importance of each subsection of the descriptor in the matching process

The feature $j_{(n+1)}$ which yields the smallest magnitude of the vector D_j is determined to be the best possible match for the feature i_n.

D. Prediction of feature position

The predicted positions of the features is determined by calculating the velocity of the features and adding their calculated velocity to their current position.

$$x_{(n+1)} = V_x + x_n \tag{14}$$

$$y_{(n+1)} = V_y + y_n \tag{15}$$

- Where $x_{(n+1)}$ is the predicted x position of a feature in the frame$(n+1)$ and x_n is the x position of the feature in frame(n)
- Where $y_{(n+1)}$ is the predicted y position of the feature in the frame$(n+1)$ and y_n is the y position of the feature in frame(n)
- Where V_x is the calculated velocity of the feature in the x direction and V_y is the calculated velocity in the y direction

E. Rejection of outliers

If the smallest D_j is less than or equal to a chosen tolerance factor t, then the match is determined to be an inlier, if the smallest D_j is greater than the chosen tolerance factor, then the match is determined to be an outlier.

F. Output of tracking algorithm

The output from the tracking algorithm is an array $X_{i,n}$ which stores vectors (called change vectors) for each feature in the feature list. The elements in these vectors contain the change in pixel position for the specific feature for each analyzed frame in the video sequence.

G. Stationary probability algorithm

The stationary probability algorithm determines the stationary probability of the features based on two criteria. The first criterion stipulates that a feature with a high stationary probability must experience an average acceleration which decreases the magnitude of the feature's velocity. The second criterion is stipulates that a feature with a high stationary probability must have an average velocity which is near zero. The stationary probability associated with the first criterion is described by $P_{s(i)\_1}$ and the stationary probability associated with the second criterion is described by $P_{s(i)\_2}$.

The change matrix or array $X_{i,n}$ is used as an input to the probability algorithm.

$$X_i = (\frac{1}{N_i})\sum_{n=1}^{N_i} X_{i,n} \tag{16}$$

X_i is the average change in position for a tracked feature i over a number N_i of observed frames for feature i from the video sequence. If $X_i \leq \alpha$ (where α is a stationary tolerance), then the stationary probability $P_{s(i)}$ for feature i is described by

$$P_{s(i)} = 1 - (0.05)(\frac{X_i}{\alpha}) \tag{17}$$

The stationary probability is therefore linear where $X_i \leq \alpha$. $P_{s(i)\_1}$ and $P_{s(i)\_2}$ are calculated if $X_i > \alpha$

$$P_{s(i)\_2} = (\frac{1}{N_i})\sum_{n=1}^{N_i}(\frac{\alpha}{X_{i,n}}) \tag{18}$$

$$P_{s(i)\_1} = log_a((\frac{d}{N_i})\sum_{n=1}^{N_i} log_b(\frac{X_{i,n}}{X_{i,(n+1)}})) \tag{19}$$

If $(\frac{d}{N_i})\sum_{n=1}^{N_i} log_b(\frac{X_{i,n}}{X_{i,(n+1)}}) \leq 1$ then $P_{s(i)\_1} = 0$

If $log_a((\frac{d}{N_i})\sum_{n=1}^{N_i} log_b(\frac{X_{i,n}}{X_{i,(n+1)}})) \geq 1$ then $P_{s(i)\_1} = 1$

If $X_i > \alpha$ then $P_{s(i)}$ is described by

$$P_{s(i)} = (c_1)(P_{s(i)\_1}) + (c_2)(P_{s(i)\_2}) \tag{20}$$

IV. Results

A. Experimental setup

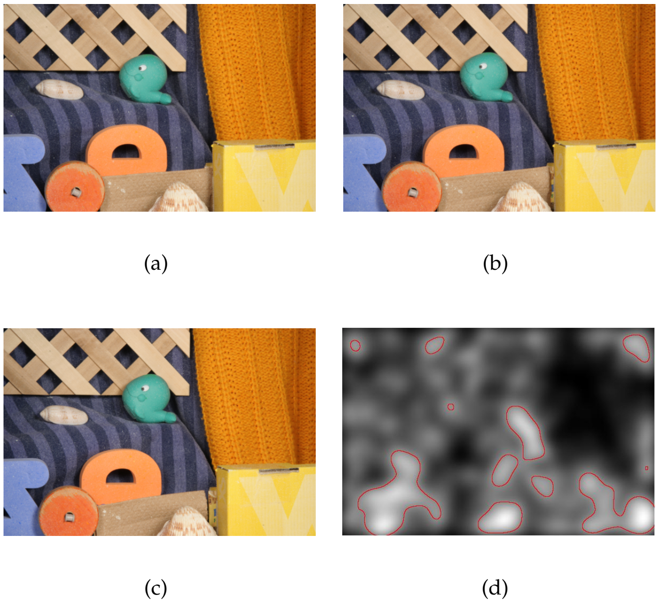
Three experiments were performed in MATLAB R2010a, the images used as inputs for the experiments consisted of the 3 sets of images from the Middlebury optical flow dataset [15]. The specifications of the PC used in the experiments are as follows;

- Processor: Core i3 540 @ 3.07GHz
- Memory: 4GB DDR3 1333
- Operating system: Windows 7 32-bit Home Basic
- MATLAB version: R2010a

B. *Discussion of results*

Figure 3 - 5 from a to c shows the images 01 to 08 of the image set from the Middlebury optical flow dataset, Figure 3 uses the RubberWhale set of images, Figure 4 uses the MiniCooper set of images and Figure 5 uses the Urban3 set of images. Figure 3 (d), Figure 4 (d) and Figure 5 (d) shows a relatively dense representation of the stationery probabilities of the features or regions in the images in the image set from the Middlebury optical flow dataset. Where pixels which are more white represent regions in the image 08 with low stationery probability and pixels which are more black represent regions in the image 08 with high stationery probability. Regions of pixels which have stationery probabilities below a certain value are regarded to be moving, these pixels are outlined in red in Figure 3 (d), Figure 4 (d) and Figure 5 (d).

In Figure 3 a large majority of the objects are moving, the image Figure 3 (d) represents this as the majority of the groups of pixels in the image are more white than than black. It can be seen that the objects in the lower half of the of images in Figure 3 from a to c as well as the tail of the rubber whale move a greater distance relative to the other objects, these objects should therefore have a lower stationery probability. It can be seen in Figure 3 (d) that these objects are close to white white and are outlined in red, the system therefore determined these objects to be moving and having low stationery probabilities. The system showed acceptable performance for the test performed in Figure 3.


(a)

(b)

(c)

(d)

Fig. 3. Test1: multiple moving objects, (a) Frame 01, (b) Frame 04, (c) Frame 08, (d) Groups of features with lowest stationary probability

In figure 4 the moving objects are restricted to the top half the man, the boot of the car and small changes in the reflections on the side of the car. The image Figure 4 (d) represents this as the pixels which make up the top half of the man and the boot of the car are close to white and are outlined in red. The system therefore determines these objects to be moving and having low stationery probabilities. The system also determines the change in the reflections on the car as a number of moving objects as these pixels are close to white and are outlined in red. The are however a number of groups of pixels not part of the moving objects in the images Figure 4 from a to c which are gray, these groups of pixels are therefore determined to have a stationery probability less than 1. It can be seen however than these groups of pixels have not moved and should therefore have a stationery probability of 1. Aside from this poor performance the system performed acceptably for the test performed in Figure 4.

In figure 5 all of the objects are moving, it can be seen however that the nearest buildings particularly the nearest buildings on the left hand side move a greater distance relative to the other objects in the images from Figure 5 from a to c. The image Figure 5 (d) represents this as all of the groups of pixels are close to white or gray and the groups of pixels making up the nearest buildings are close to white and are outlined in red. The system therefore determines these objects to be moving and having low stationery probabilities. Some groups of pixels on the nearest buildings are more white than others however indicating that they system determines these objects to have lower stationery probabilities. Whereas these groups of pixels can be observed as to move a similar distance to the surrounding groups of pixels on the nearest buildings, indicating that the groups of pixels should similar stationery probabilities. Aside from this poor performance the system performed acceptably for the test performed in Figure 5.

The system showed some inaccurate computation of the stationery probabilities of groups of the pixels in the test images. These inaccuracies can be attributed to mismatches occurring in the matching process of the tracking algorithm. The feature detection algorithm and descriptor used for this system do not take scale and orientation of features into account and does not provide enough unique information for the tacking algorithm to operate in a robust manner. The relatively poor performance of the tracking algorithm greatly effects the ability of the system to accurately determine the stationery probability of regions in a video sequence.

V. Conclusion

In this paper a method was proposed to determine the stationery probability of regions or feature points in

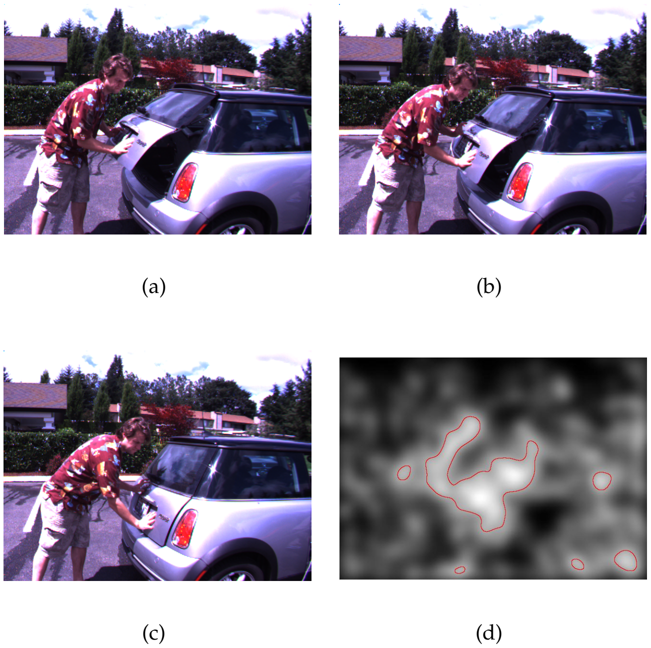

(a) (b) (c) (d)

Fig. 4. Test2: Stationary scene with moving objects, (a) Frame 01, (b) Frame 04, (c) Frame 08, (d) Groups of features with lowest stationary probability

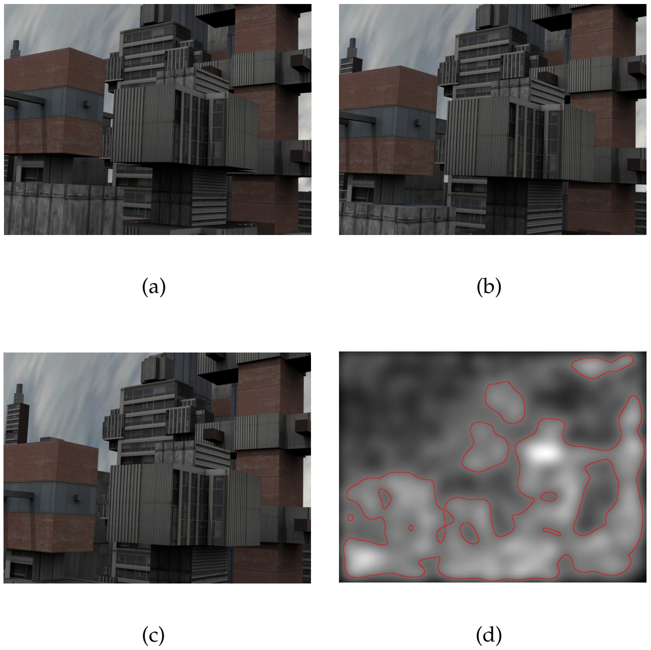

(a) (b) (c) (d)

Fig. 5. Test3: Moving scene - rotational motion, (a) Frame 01, (b) Frame 04, (c) Frame 08, (d) Groups of features with lowest stationary probability

a video sequence. The system showed acceptable results in the performed experiments. The poor performance of the tracking algorithm used for this paper did however effect the systems ability to accurately determine the stationery probability of the regions or features in a video sequence. This can be improved by implementing feature detection and description algorithms which are invariant to affine transformations such as scale and rotation changes of feature points. A more robust feature detection and description method could greatly improve the performance of the system.

References

[1] H.B. Kashani, S.A. Seyedin and H.S. Yazdi, *"A Novel Approach in Video Scene Background Estimation"*, International Journal of Computer Theory and Engineering, Vol. 2,No. 2, pp 1793 - 8201, 2 April, 2010.

[2] Y. Liang, Z. Wang, X. Xu, X. Cao, *"Background Pixel Clissification for Motion Segmentation Using Mean Shift Algorithm"*, Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, pp 1693 - 1698, 19-22 August 2007.

[3] R. Cucchiara, M. Piccardi, and A. Prati, *"Detecting moving objects, ghosts, and shadows in video streams"*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 10, pp 1 - 6, Oct. 2003.

[4] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, *"Principles and practice of background maintenance"*, Proc. IEEE Int. Conf. Computer Vision, pp 255 - 261, Sept. 1999.

[5] I. Haritaoglu, D. Harwood, and L. Davis, *"Real-time surveillance of people and their activities"*, IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp 809 - 830, Aug. 2000.

[6] F.C. Cheng and Y.K. Chen, *"Effective $\Sigma - \Delta$ background estimation for video background generation"*, IEEE Asia-Pacific Services Computing Conf., 2008.

[7] C. Wren, A. Azarbaygaui, T. Darrell, and A. Pentland, *"Real-time tracking of the human body"*, IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp 780 - 785, July 1997.

[8] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S.Russel, *"Towards robust automatic traffic scene analysis in real-time"*, Proc. ICPR, pp 126 - 131 Oct. 1994.

[9] C. Stauffer and W. Grimson, *"Learning patterns of activity using real-time tracking"*, IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp 747 - 757, Aug. 2000.

[10] D. Farin, P.H.N. de With, and W. Effelsberg, *"Robust Background Estimation for Complex Video Sequences"*, 2003 International Conference ICIP, vol.1, pp 145 - 148, 14-17 Sept. 2003.

[11] C. Harris and M.J. Stephens, *"A combined corner and edge detector"*, Alvey Vision Conference, pp 147 152, 1988.

[12] C. Schmid, R. Mohr, and C. Bauckhage, *"Evaluation of interest point detectors"*, International Journal of Computer Vision, 37(2), pp 151 172, June 2000.

[13] W. Gao, L. Yang, X. Zhang, H. Liu *"An Improved Sobel Edge Detection"*, 978-1-4244-5540-9/10/$26.00 2010 IEEE, pp 67 - 71, 2010.

[14] D.G. Lowe, *"Distinctive Image Features from Scale-Invariant Keypoints"*, International Journal of Computer Vision, 2004.

[15] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, R. Szeliski, *"A Database and Evaluation Methodology for Optical Flow"*, International Journal Computer Vision (2011), 92 pp 1 - 31, 2011.