An Automated Real-Time People Tracking System Based on KLT Features Detection

Nijad Al-Najdawi[1], Sara Tedmori[2], Eran Edirisinghe[3], and Helmut Bez[3]
[1]Information Technology Department, Al-Balqa Applied University, Jordan
[2]Department of Computer Science, Princess Sumaya University for Technology, Jordan
[3]Department of Computer Science, Loughborough University, UK

Abstract: *The advancement of technology allows video acquisition devices to have a better performance, thereby increasing the number of applications that can effectively utilize digital video. Compared to still images, video sequences provide more information about how objects and scenarios change over time. Tracking humans is of interest for a variety of applications including surveillance, activity monitoring and gate analysis. Many efficient object tracking algorithms have been proposed in literature, however part of those algorithms are semi-automatic requiring human interference. As for the fully automated algorithms, most of them are not applicable to real-time applications. This paper presents a low cost automatic object tracking algorithm suitable for use in real-time video based systems. The novelty of the proposed system is that it uses a simplified version of the Kanade-Lucas-Tomasi (KLT) technique to detect features of both continuous and discontinuous nature. As discontinuous feature selection is subject to noise, and would result in non-optimal feature based object tracking, the authors propose the use of a Kalman filter for the purpose of seeking optimal estimates in tracking. The integrated tracking system is capable of handling shadows and is based on a dynamic background subtraction strategy that minimises errors and quickly adapts to scene changes. Experimental results are provided to demonstrate the system's capability of accurately tracking objects in real-time applications where scenes are subject to noise particularly resulting from occlusions and sudden illumination variations.*

Keywords: *Object tracking, kalman-filter, features selection, KLT.*

Received March 13, 2010; accepted May 20, 2010

1. Introduction

Automatic object detection and recognition is of paramount importance for security systems and video surveillance applications. Automated video surveillance addresses real time observation of people and vehicles within a busy environment. Surveillance systems must be able to detect and track objects moving in its field of view, classify these objects and detect some of their activities. Existing surveillance systems can be classified into categories according to the environment they are primarily designed to observe (i.e., indoor, outdoor or airborne; the number of sensors i.e., single camera vs. multiple cameras; etcetera). Even though tracking is considered an important part of the above process, it is the most error prone component of a surveillance system. This can be attributed to changes in illumination, shadows, occlusions, and reflections in natural environments.

Processing a video stream for characterizing events of interest relies on a high level description of the video stream, which is based on an accurate detection and tracking of the moving objects and on the relationship of their trajectories to the scene. Large numbers of surveillance systems have been proposed in recent years. Most of the work related to visual surveillance tracking is based on change detection or frame differencing and assumes the use of a stationary camera. Wren *et al*. [20] have developed the PFinder tracking system, which uses a background model to locate objects. Their system tracks the full body of a person, but it assumes that only a single person is present in the scene. Stauffer and Grimson [17] propose an adaptive multi-modal background subtraction method, which can deal with slow changes in illumination, repeated motion from background clutter and long-term scene changes. After background subtraction, the detected objects are tracked using a multiple hypothesis tracker. Haritaouglu *et al*. [12] employ a dynamic appearance model to track people. Single person and groups are distinguished using projection histograms. Ricquebourg and Bouthemy [15] proposed tracking people by exploiting spatio-temporal slices. Their detection scheme involves the combined use of intensity, and temporal differences information. This process is dynamically updated. Their approach tracks the apparent contours of moving articulated structures by using spatio-temporal slices from the image sequence. Shah and Javed [18] have formulated object tracking as a region correspondence problem, given the background subtraction results from the approach proposed in [17]. The choice of the right filter and data association methodologies has a direct impact on the tracking system, in terms of enhancing their functionalities. However, the right choice is highly related to the tracking scenarios. Particle filtering was introduced as the "Condensation algorithm" by Blake and Isard [6]. Boykov and Huttenlocher [7] employed the Kalman filter to track vehicles in an adaptive framework. Rosales and Sclaroff [16] used the Extended Kalman Filter to estimate a 3D object trajectory from 2D image motion. MacCormick and Blake [14] introduced the probabilistic exclusion for tracking multiple objects. Chen, *et al.* [10] used the Hidden Markov Model formulation combined with JPDAF data association for tracking people.

This paper represents a continuation of the authors' efforts to enhance the process of tracking people in real-time environments [4]. In this paper, the proposed work is based on an enhanced version of the Kanade-Lucas-Tomasi algorithm [19] of object tracking. However, instead of detecting and tracking a large amount of feature points of an object in a continuous manner (i.e., if continuity of detection is broken, the features are predicted to maintain continuity), this paper proposes a simplified algorithm that detects and tracks a limited amount of both, continuous and discontinuous features. To overcome the problem of noise created by the adaptation of such an approach, a Kalman Filter is employed as an extension to the KLT method. Dynamic background subtraction and shadow detection algorithms are included in this approach. The experimental results obtained from the proposed algorithm demonstrate the robustness of tracking in real time applications.

For clarity of presentation, the paper is divided into seven sections. Apart from this section which provided an introduction to the application domain, section 2 provides an insight to the shadow detection algorithm used in this research. Section 3 discusses the dynamic background subtraction method used for object segmentation. Section 4 discusses the principles of feature detection and the simplified KLT algorithm introduced in this paper. Section 5 explains the usage and adaptation of Kalman filtering in object tracking. Section 6 introduces the proposed algorithm and provides experimental details along with a comprehensive analysis of the results. Finally, section 7 concludes with an insight into possible future developments and improvements of the proposed scheme.

2. Shadow Detection

Many different approaches have been proposed in literature for moving object segmentation from image sequences. Unfortunately, none of these segmentation methods are able to accurately distinguish between moving objects and their corresponding shadows. Misclassifying shadow areas as foreground leads to inaccurate segmentation and extraction of moving objects. The fact that shadows have similar motion to the objects casting them, and are detected as an element of the object, has lead researchers to investigate techniques for effective shadow detection and removal. Generally, shadows are classified as either self or cast shadows. Object tracking algorithms are mostly concerned with cast shadows, i.e. shadows cast by an illuminated object onto other objects. Although, many shadow detection and removal algorithms have been proposed in literature, most of the proposed methods that claim to be object and environment independent include some minor assumptions about the scene geometry or spectral distributions of the light sources. In this paper, the shadow detection method proposed by the authors in [2] is adopted. This shadow detection method is based on a physically-derived hypothesis for shadow identification. The algorithm is proven to be fast, reliable, and can be applied in real-time applications. It is shown that the algorithms effectively remove shadows umbra and penumbra under various lighting and environmental conditions. The technique can be described as follows:

1. Shadow Condition: Let $q \in R^3$ be a point on the surface of an object in an illuminated three-dimensional scene, and let n_q be a neighbourhood of q in the surface. Using the simple geometric representation of light rays and a simple reflection model, it is possible to show that the light energy received at points $r \in n_q$ in the absence of an object casting a shadow over n_q is affinely related, to a high degree of approximation, to the energy received when a shadow is cast over n_q by an object. The same affine parameters being applicable to the whole neighbourhood n_q. It is of course clear that when a shadow is cast over a neighbourhood, less light is received there - as compared to the fully illuminated state - and that this condition should also be included in a shadow model. It follows that reflected energies behave similarly and hence: The luminance function $L : n_q \to R$ when no shadow is cast over n_q is affinely related to the luminance function $L^* : n_q \to R$ when a shadow is cast; i.e. for n_q to be in shadow we have $L^*(r) \approx \lambda L(r) + \mu$ and $L^*(r) < L(r)$, for some constants λ and μ, for all $r \in n_q$. These neighbourhood relationships are fundamental to the remainder of the shadow detection technique, and constitute the basis of the shadow detection algorithms.
2. Determination of the Affine Parameter: If J_p denotes the matrix $(J_p)_{i,j} = 1, \forall 0 \le i, j \le k-1$

then the neighbourhood luminance relation $L^* = \lambda L + \mu$ translates directly to the relation $P^* = \lambda P + \mu J_p$ for pixel blocks P^* and P at identical positions in the object and background frames F^* and F respectively. Appropriate affine parameters (λ, μ) may be computed, for the block-pair (P^*, P) in a number of ways. For example we could compute them from the relations $\overline{P}^* = \lambda \overline{P} + \mu$, $\sigma(P^*) = \lambda\sigma(p)$ from which we obtain: $\lambda = \sigma(P^*)/\sigma(P)$ and $\mu = \overline{P}* - \lambda\overline{P}$. If λ, μ are the affine parameters for a block pair (P, P^*), then the conditions for P^* to be a shadow block are equation 1:

$$\overline{P}^* < \overline{P} \text{ and } \frac{\| P^* - (\lambda P + \mu J_p \|^2}{\| p \|_2} \approx 0 \quad (1)$$

i.e., the mean luminance of P^* is lower than that of P and the affine condition hold for the pair (P, P^*). Figures 1-3 show examples of the shadow detection and removal algorithm when applied to different indoor and outdoor environments. Processed frames show the accuracy of the results. A quantitative evaluation and some more applications of the proposed algorithm can be found on [2].

Figure 1. Sara," represents an indoor video sequence, with multiple combination of light sources, spectrally equal and of equal intensities. This figure illustrates the use the shadow detection and removal algorithm. Frames on the top represent the original frames from the sequence; frames on the bottom are the corresponding processed frames.

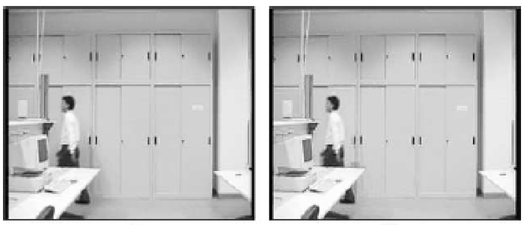

Figure 2. The benchmark "Laboratory" video sequence, image on the left corresponds to frame number 198 of the original video sequence, image on the right corresponds to the processed frame, cast shadow has been detected and removed.

3. Objects Segmentation

Object segmentation intends to separate foreground regions (object/s to be tracked) from background areas of the image. Static background subtraction techniques are not capable of handling illumination and scene changes; therefore they are not considered in this research. Many dynamic background subtraction algorithms have been proposed in literature, such as the approaches proposed in [1, 5, 9, 11, 13, 17, 21]. Such approaches are mainly designed to track vehicles, and are capable of handling huge variations in the environment.

The approach proposed in [11] is adopted as a part of this research primarily for its simplicity and efficiency. In this work, the varying region in the monitoring image is derived from the background and time differences, and is classified into (a) moving objects, (b) stationary objects, and (c) the change due to illumination variation.

Figure 3. The benchmark "Campus" video sequence, image on the left corresponds to frame number 109 of the original video sequence, image on the right corresponds to the processed frame, cast shadow has been detected and removed.

Based on this classification, region (c) and the region without any change are used for background updating. Also, by forming a cumulative image in which the binary images composed of regions (a) and (b) are stored for a specified length of time, the detection of the stationary objects is performed and the environmental conditions are determined. Given the above method, for any given frame of video, where it represents the frame number, the dynamic generated background model is subtracted, to form a background subtracted frame. The magnitudes of the pixel values in the background subtracted frame thus allow pixel segmentation into foreground and background regions.

4. The Simplified KLT Technique

In order to avoid having to track all pixels in the resulting image and within a given foreground object, a variety of techniques for tracking objects based only on a limited set of feature points have been proposed in literature. Out of these techniques Kanade-Lucas-Tomasi (KLT) method has been chosen as a basis/benchmark for many algorithms proposed in previous literature. This is due to its simplicity and limited assumptions made about the underlying image. The KLT tracks an object in two steps, it locates the trackable features in the initial frame, then tracks each one of the detected features in the rest of the frames by means of its displacement. In this work, the authors use the first part of the KLT (selecting trackable features) and subsequently track the whole set of features together instead of tracking each feature separately.

4.1. Features Detection and Tracking

Image gradients provide information about the linear intensity of image texture; thus the selection of “trackable features“ can rely on texture. Texture pattern exists when multiple pixels in a certain area have different distinguishable values. The features to be tracked can be precisely defined by the texture within a finite size window. The window size would determine the number of features detected. In this application, a window of size 4x4 pixels is used (8×8 and 16×16 block-sizes can be used but the number of detected features will decrease as the block-sizes increase). For each pixel in the window, the pixel gradient values ϕ_x, ϕ_y are calculated to form a 2×1 gradient vector, $\phi = (\phi_x, \phi_y)^T$. Thus,

$$\phi\phi^T = \begin{bmatrix} \phi_x{}^2 & \phi_x\phi_y \\ \phi_x\phi_y & \phi_y{}^2 \end{bmatrix} \quad (2)$$

by considering all pixels within the window, matrix w is defined as follows:

$$w = \sum_{i=1}^{n} \sum_{j=1}^{n} \phi_{ij}\phi_{ij}{}^T \quad (3)$$

Equation 3 forms the first part of the Kanade-Lucas-Tomasi tracking equation. Note that matrix w contains pure texture information. Thus by analyzing the eigen values of w, λ_1 and λ_2, it is possible to classify the texture in the window as follows: Two small eigen values characterizes approximately an invariable intensity pattern; one small and one large eigen value indicates a detected linear pattern; two large eigen values indicate a trackable feature. This feature can either be a multidirectional texture such as a corner, or any texture pattern, which has a major change in intensity in more than one direction. To detect the presence of a feature point, it is important to determine when the eigen values are big enough. Thus it is critical to set up a threshold for the eigen values, which depends on the environment where tracking is performed. For experimental purposes, the threshold for λ_1 and λ_2 is set to 2500. In addition, it is also critical to keep the number of features as minimum as possible. This is to ensure a reduction of the amount of calculations preformed by the system for real-time purposes.

The detected features are represented by white dots as can be seen in the experimental figures provided in this paper. Once the features are extracted, the tracking step begins. In tracking, the location of the object is determined in the scene by means of its detected features. Therefore, the center of gravity C_g , for the whole set of the detected features is calculated as follows:

$$C_{g_x} = \frac{1}{N} \sum_{i=1}^{n} x_i \ , \quad C_{g_y} = \frac{1}{N} \sum_{j=1}^{n} y_j \quad (4)$$

Where x , y , n and N are the location of the feature in the x direction, the location of the features in the y direction, the feature number, and the total number of the features, respectively. A black rectangle is drawn in the scene whose center is the value of C_g , and the dimensions are based on the average size of the desired extracted object, (see Figure 4). Due to the randomness associated with feature detection, in some cases, the value of C_g is not exactly correct. This is true particularly when some features are not detected on the boundary of the object, which may cause the center of gravity to be shifted towards the location of the higher density regions.

Figure 4. Represents frame (5) of the benchmark “Hall“ video sequence.

Therefore, the Kalman filter (introduced in the next section) is employed as a corrector for the detected object position.

5. Kalman Filtering

Kalman Filter (KF) is a set of mathematical equations that implement a predictor-corrector type estimator that optimally minimizes the estimated error covariance. The Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by the linear stochastic difference equation [8] equation 5:

$$\vec{X}(k) = A\vec{X}(K-1) + Bu_{(k)} + w_{(K-1)} \quad (5)$$

with a measurement equation given by equation 6:

$$Z_{(K)} = Hx_{(K)} + v_{(k)} \quad (6)$$

Where $X_{(K)}$ and $Z_{(K)}$ are the system state and the measurements respectively, $w_{(K)}$ and $v_{(K)}$ are the process and measurement noise, respectively. These two equations are often referred to as the process and the measurement models; they serve as the basis for all linear estimation methods, such as the Kalman filter which consists of the following five equations 7-11:

(Predictor)

$$\vec{\hat{X}}_{(k|k-1)} = A\vec{\hat{X}}_{(k|k-1)} \tag{7}$$

$$P_K(k \mid k-1) = AP_k(k-1 \mid k-1)A^T + Q \tag{8}$$

(Corrector)

$$K = P_{K(k|k-1)}C^T\left(CP_{K(k|k-1)}C^T + R\right)^{-1} \tag{9}$$

$$\vec{\hat{X}}_{(k|k)} = \vec{\hat{X}}_{(k|k)} + K\left(\vec{Y}_{(k)}\right) - C\vec{\hat{X}}_{(k|k-1)} \tag{10}$$

$$P_{K(k|k)} = (I - CK)P_{K(k|k-1)} \tag{11}$$

Where X, P, A, C, I, K, are: the system state, error covariance, propagation, transformation, identity, and Kalman Gain matrices, respectively. The proposed tracking system requires the use of the Kalman filter in each step; therefore a simplified version of the filter is implemented to keep the required computations minimal for real-time applications.

6. Architecture of Proposed System

In past literature, researchers have attempted to combine Kalman Filtering with object tracking. These contributions exploit both predicted and measured motion information, while trying to yield a best estimate of motion vectors. Nevertheless, the contribution differs in how objects are located, and/or what part of that object should be tracked. In this paper, objects are tracked using a surrounding arbitrary size rectangle. The size of the rectangle is determined according to the detected features coordinate at the object boundary. The measurements part of the filter is the location of the rectangle upper-right hand side corner. The initial Kalman filter parameter values have been calculated experimentally with Q=0.7, R=0.8, and A=0.9. In the proposed system as shown as in Figure 5, the Motion Vector (MV) is calculated between each two consecutive frames. The motion vector value is fed into the KF as the input Predicted motion vector (Pmv). For the next frame, the Measured motion vector (Mmv) is fed into the KF as the measured input. Consequently, the Optimal estimate of the motion vector (Omv) is taken from the KF and used as a predictor for the next iteration.

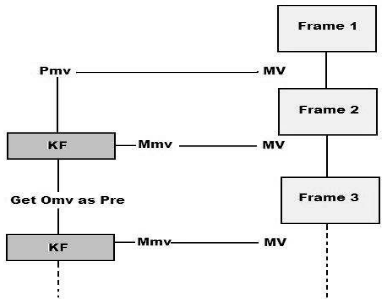

Figure 5. The proposed system architecture.

The process continues as the camera feeds frames into the system. Based on the optimal estimate, a white rectangle is drawn around the object (Black rectangles are kept in the scene to highlight the accuracy of the algorithm after the use of Kalman filter). This demonstrates the KF capability of handling sudden changes in the object's position. Figure 6 shows a case where the measured location (black-box) has suddenly shifted upwards because of the high density of the detected features at the top of the object. The Kalman predicted location (white-box) has accurately detected the sudden change and located the object at the right place using information from previous frames.

Figure 6. Represents frame (12) of the benchmark "Hall" video sequence.

Figure 7 shows compatible results between the Kalman and the detected location, since the detected location is located correctly. Figures 8, 9 and 10 show how the Kalman corrected trajectories give more accurate results. It is clear that the Kalman results have detected and corrected the sudden changes in the scene.

Figure 7. Frame 8- hall sequence.

The test has been run into more than 600 frames of the benchmark "Hall" video sequence, and the created Holywell-la, Farnham, and Holywell-s video sequences.

The results show that the Kalman filter predicted trajectories through the sequences are smooth and closer to the original trajectory compared with the detected location (relying on features detection only). The Mean Square Error (MSE) is calculated as the basic tool of comparisons, (using pixel unit distance) results for these particular tests (in average) are shown in Table 1. It is clear that using Kalman filter enhances the MSE results.

Figure 8. Frame 24- holywell-la sequence.

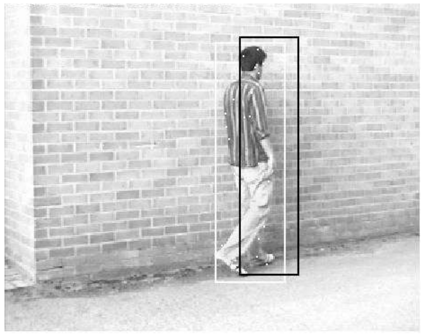

Figure 9. Frame 17- farnham sequence.

Figure 10. Frame 39- holywell-s sequence.

Table 1. MSE results of the proposed system, results show how kalman filer enhances the tracking process by means of reducing the MSE.

Frame Number	Hall Video Sequence		Holywell-ls Video Seuqnce		Holywell-s Video Sequence		Farnham Video Sequence	
	MSE Simplified KLT	MSE (KLT+Kalman)	MSE Simplified KLT	MSE (KLT+Kalman)	MSE Simplified KLT	MSE (KLT+Kalman)	MSE Simplified KLT	MSE (KLT+Kalman)
1	20.1	19.1	23.4	20.2	21.2	19.2	15.9	12.5
2	21.4	17.7	28.2	24.6	21.6	18.3	16.8	12.4
3	22.3	19.2	20.1	18.4	21.1	18.4	16.4	12.1
4	23.2	20.3	20.4	17.7	19.3	18.9	15.3	11.7
5	21.9	19.6	23.4	18.2	22.4	19.3	14.0	11.7
6	21.2	19.2	21.6	19.4	23.6	18.0	15.8	12.3
7	22.1	15.2	23.2	16.6	23.2	18.5	15.6	12.4
8	22.3	19.9	22.9	19.3	22.1	18.2	15.3	12.5
9	21.5	18.3	27.1	18.4	22.9	18.7	15.9	12.8
10	22.1	20.3	23.2	19.3	22.6	19.2	15.4	12.6
11	22.4	18.2	16.3	19.3	21.9	19.6	14.2	11.7
12	23.1	19.4	20.4	19.9	23.1	18.7	14.8	12.6
13	22.3	20.6	23.0	18.8	20.4	18.3	15.2	12.5
14	22.6	19.4	23.2	18.8	21.2	19.4	14.9	11.7
15	22.3	17.3	23.1	19.2	20.1	19.8	15.1	12.2
16	21.6	19.6	19.8	17.3	20.7	18.9	15.3	11.9
17	23.4	18.4	20.7	19.7	20.2	18.1	15.1	12.1
18	21.3	19.3	20.3	19.2	22.1	19.2	14.8	11.6
19	22.1	18.8	21.3	20.2	20.4	18.4	14.1	11.2
20	20.8	19.4	22.7	18.4	21.8	18.7	15.2	12.4
Average	21.95	18.96	22.215	19.145	21.595	18.79	15.255	12.145

7. Conclusions

This paper introduced an automated novel algorithm, for object tracking in video scenes, and tracking in real-time security systems. The system comprises an enhanced version of the Kanade-Locus-Thomasi features extraction algorithm of object tracking. However, instead of detecting and tracking a large amount of feature points of an object in a continuous manner, this paper proposes a simplified algorithm that detects and tracks a limited amount of both, continuous and discontinuous features. To overcome the problem of noise created by the adaptation of such an approach, a Kalman filter is employed as an extension to the KLT method. The Kalman filter is also used to maintain the continuity of features detection when it is broken. The proposed system provides excellent real-time tracking capabilities. The system is capable of tracking multiple objects in regular indoor/outdoor environments. A dynamic background subtraction algorithm in addition to a shadow detection and removal techniques are integrated into the system to facilitate and enhance the tracking process.

References

[1] AlGarni S. and Abdennour A., "Moving Vehicles Detection Using Automatic Background Extraction," *in Proceedings of World Academy of Science, Engineering and Technology*, Pennsylvania, pp. 1307-6884, 2006.

[2] AlNajdawi N., Bez H., and Edirisinghe E., "A Novel Approach for Cast Shadow Modelling and Detection," *in Proceedings of the IET International Conference on VIE*, India, pp. 553-558, 2006.

[3] Al-Najdawi N., Bez H., and Edirisinghe E., "A Novel Hypothesis for Cast Shadow Modelling and Detection for Objects Tracking and Recognition," *in Proceedings of the 6th IASTED on Visualization, Imaging and Image Processing*, Palma, Spain, pp. 377-382, 2006.

[4] AlNajdawi N., Edirisinghe E., and Bez H., "Real-Time Object Tracking Based on a Limited, Discontinuous Feature Set," *in Proceedings of the 5th IASTED International Conference Visualization, Imaging and Image Processing*, Benidorm, Spain, pp. 645-649, 2005.

[5] Amamoto N. and Fujii A., "Detecting Obstructions and Tracking Moving Objects by Image Processing Technique," *Computer Journal of Electronics and Communication*, Japan, vol. 82, no. 11, pp. 28-37, 1999.

[6] Blake A. and Isard M., "Condensation Conditional Density Propagation for Visual Tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[7] Boykov Y. and Huttenlocher D., "Adaptive Bayesian Recognition in Tracking Rigid Objects," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition-CVPR*, Canada, pp. 697-704, 2000.

[8] Brookner E., *Tracking and Kalman Filtering Made Easy*, John Wiley and Sons, 1999.

[9] Chen T., Lin Y., and Chen T., "Intelligent Vehicle Counting Method Based on Blob Analysis in Traffic Surveillance," *in Proceedings of the IEEE International Conference on Innovative Computing, Information and Control*, USA, pp. 238-238, 2007.

[10] Chen Y., Rui Y., and Huang T., "JPDAF Based HMM for Real-Time Contour Tracking," *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, pp. 543-550, 2001.

[11] Han C. and Zhang Q., "Real-Time Detection of Vehicles for Advanced Traffic Signal Control," *in Proceedings of the International Conference on Intelligent Transportation Systems*, Phuket, pp. 245-249, 2008.

[12] Haritaoglu I., Harwood D., and Davis L., "W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People," *in Proceedings of International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 222-227, 1998.

[13] Jinglei Z. and Zhengguang L., "A Vision-Based Road Surveillance System Using Improved Background Subtraction and Region Growing Approach," *in Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, Qingdao, pp. 819-822, 2007.

[14] MacCormick J. and Blake A., "Probabilistic Exclusion and Partitioned Sampling for Multiple Object Tracking," *International Journal on Computer Vision*, vol. 39, no. 1, pp. 57-71, 2000.

[15] Ricquebourg Y. and Bouthemy P., "Tracking of Articulated Structures Exploiting Spatio-Temporal Image Slices," *in Proceedings of the 4th IEEE International Conference on Image Processing*, California, pp. 480-483, 1997.

[16] Rosales R. and Sclaroff S., "3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, USA, pp. 1-7, 1999.

[17] Stauffer C. and Grimson E., "Learning Patterns of Activity Using Real Time Tracking," *Computer Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-767, 2000.

[18] Shah M. and Javed O., "Tracking and Object Classification for Automated Surveillance," *in Proceedings of the 7th European Conference on Computer Vision*, Copenhagen, pp. 343-357, 2002.

[19] Tomasi C. and Kanade T., "Factoring Image Sequences into Shape and Motion," *in Proceedings of the IEEE Workshop on Visual Motion*, Japan, pp. 21-28, 1991.

[20] Wren C., Azarbayejani A., Darrell T., and Pentland A., "Pfinder: Real-Time Tracking of the Human Body," *Computer Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.

[21] Zhifang L. and Zhisheng Y., "A Real-Time Vision-Based Vehicle Tracking and Traffic Surveillance," *in Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/ Distributed Computing*, Qingdao, pp. 174-179, 2007.

Nijad Al-Najdawi received his BSc degree in computer science from Mu'tah University, Jordan in 1999. He obtained his MSc degree in multimedia and Internet computing, and a PhD degree in machine vision and autonomous systems from Loughborough University, UK in 2003 and 2006, respectively. After which he joined Loughborough University as a Research Associate (Post-Doc position) in the Electronic and Electrical Engineering Department. Currently, he is appointed as an assistant professor at Al-Balqa Applied University, Jordan. His research interests include: Image processing, video coding, objects tracking and recognition.

Sara Tedmori received her BSc degree in computer science from the American University of Beirut, Lebanon. In 2003, she obtained her MSc degree in multimedia and Internet computing from Loughborough University, UK. In 2008, she received her PhD degree in computer science from Loughborough University, UK. Currently, she is appointed as an assistant professor in the Computer Science Department at Princess Sumaya University of Technology, Jordan. Her research interests include: Object Tracking, image processing, expertise locator, knowledge extraction, knowledge sharing, and privacy.

Eran Edirisinghe received his PhD degree from the Department of Computer Science, Loughborough University, in September 1999. He was promoted to a Senior Lecturer in February 2004 and received the title of Reader in digital imaging in 2008. He is a member of the College of Peers of the Engineering & Physical Sciences Research Council, UK. His research interests include image and signal processing, video coding, texture synthesis and novel mobile applications.

Helmut Bez received a first class honours degree in Mathematics from the University of Wales in 1972, and MSc and PhD degrees from Oxford University in 1973 and 1976, respectively. He joined Rolls Royce Aero Engines in 1976, and in 1980 was appointed to the academic staff of the Department of Computer Science at Loughborough University, where he now holds the title of Reader in geometric computation. Current research interests include: The determination and application of the invariant properties of path functions, rational parametrisation, image processing and parallel computation.