Techset Composition Ltd, Salisbury
Doc: {IEE}CVI/Articles/Pagination/CVI20110054.3d

www.ietdl.org

Published in IET Computer Vision
Received on 4th November 2010
Revised on 28th June 2011
doi: 10.1049/iet-cvi.2011.0054

ISSN 1751-9632

Adaptive mean-shift for automated multi object tracking

C. Beyan A. Temizel

Graduate School of Informatics, Middle East Technical University, 06531, Ankara, Turkey
E-mail: atemizel@gmail.com

Abstract: Mean-shift tracking plays an important role in computer vision applications because of its robustness, ease of implementation and computational efficiency. In this study, a fully automatic multiple-object tracker based on mean-shift algorithm is presented. Foreground is extracted using a mixture of Gaussian followed by shadow and noise removal to initialise the object trackers and also used as a kernel mask to make the system more efficient by decreasing the search area and the number of iterations to converge for the new location of the object. By using foreground detection, new objects entering to the field of view and objects that are leaving the scene could be detected. Trackers are automatically refreshed to solve the potential problems that may occur because of the changes in objects' size, shape, to handle occlusion-split between the tracked objects and to detect newly emerging objects as well as objects that leave the scene. Using a shadow removal method increases the tracking accuracy. As a result, a method that remedies problems of mean-shift tracking and presents an easy to implement, robust and efficient tracking method that can be used for automated static camera video surveillance applications is proposed. Additionally, it is shown that the proposed method is superior to the standard mean-shift.

1 Introduction

Object tracking is an important and challenging task in many computer vision applications such as surveillance, vehicle navigation and autonomous robot navigation. In the past few decades, various object tracking algorithms have been presented.

One of the most popular techniques is mean-shift algorithm and it is firstly adapted for tracking of non-rigid objects in [1] by focusing on histogram-based target representation and localisation using Bhattacharyya coefficients as similarity measures. Mean-shift tracking is a very commonly used tracking algorithm because of its robustness, ease of implementation and computational efficiency. However, the standard mean-shift algorithm suffers from a number of problems which adversely affect tracking performance and could cause inaccurate or even false tracking. It is not adaptable to changes in objects' size or shape since it only optimises the position of tracker by maximising a similarity function between the candidate and target objects' bounding box and its performance is dependent on correct kernel size selection in the object initialisation phase. Therefore some studies prefer to track a part of the object or an area inside the object instead of tracking the whole bounding box. However, this may cause false positives (FPs), for instance if the object colour is not homogenously distributed, the selected part might not represent the object and tracking might shift. Another shortcoming is the inclusion of background information into the object model as the kernel shape does not always fit the object. Additionally, the tracking might shift and even fail when the object is occluded or background colours are similar to the foreground objects' colours.

Many studies have attempted to improve on mean-shift to solve these problems. For instance, additional spatial information to mean-shift tracking was used in [2] to obtain a better description of the target object in order to increase the robustness of tracking. In this study, usage of spatiogram, which is a kind of histogram, where each bin is weighted by the spatial mean and covariance of pixels, was presented. The inclusion of spatiogram to standard mean-shift was reported to provide a better matching performance. In [3], a method is presented to solve the problems that may occur when the background colour is similar to the colour of the object that will be tracked using the background position on the previous frame and the current frame to compute the target model. Although this method has low computational load, it does not handle occlusions and merging of objects and tracking of multiple objects is also not possible with this method. A study to solve the problems that may arise because of incorrect mean-shift kernel scale selection was addressed in [4]. This study used difference of Gaussians kernel and provides a good tracking performance by handling changes in target scale. However, it requires high computational cost and is not suitable for real-time applications. To adapt the kernel scale and the orientation of kernel, various approaches have been proposed. For instance, Qifeng *et al.* [5] combined the mean-shift method with adaptive filtering. Even though the kernel scale and orientation estimations are successful because of the use of symmetric kernels, actual object shape might not be matched. An alternative human tracking method that uses

multiple radially symmetric kernels is proposed in [6]. In this study, a flexible tracking method was presented that allows optimisation of kernel parameters for a specific class of objects. This study is useful especially when larger kernel sizes such as arms, legs and/or smaller kernel sizes such as torso are needed to be tracked. On the other hand, Quast and Kaup [7] introduced an adaptive asymmetric kernel that is able to deal with out-of-plane rotations by using some heuristics. Another adaptive mean-shift tracking using multi-scale images is presented in [8]. In this study, the Gaussian kernel is preferred and the kernel bandwidth is determined by using a log-likelihood function. Although this method exactly estimates the position of the tracked object, it would not be efficient if it is used in real-time applications and for multiple objects tracking because of the fact that it needs nearly three iterations per object to converge to the correct object position. A method for multiple objects tracking is proposed in [9] and tries to solve the problem of inclusion of background information into the object model which may result when the relocation of an object is large. Multiple kernels were utilised to track moving areas, background and template similarities were used to improve the convergence of tracker.

A hybrid method using motion detection, template tracking and colour-based tracking is proposed in [10]. In this study, tracking using the motion detection alone, colour tracking alone and combination of colour and template tracking are compared and the most successful combination is found as colour and template tracking.

In the literature, mean-shift tracking based methods generally focus on a single shortcoming of mean shift. However, to achieve a robust automated tracking, all the problems need to be handled. Most of these methods require manual object identification and need human input to define the object that will be tracked. Besides, overwhelming majority of studies aim at tracking only one object at a time and does not provide a solution for tracking of multiple objects. In this paper, we propose an easy to implement and fully automatic multiple-object tracking algorithm for static cameras based on the standard mean-shift method. An update mechanism utilising foreground detection is used to initialise and refresh trackers to improve the tracking performance by changing the kernel size or shape and present solutions to handle the other shortcomings of mean-shift. By removing shadows and noise, FPs are also decreased.

The paper is organised as follows: Section 2 includes the detailed information about the proposed method. Section 3 contains experimental results, comparison between standard mean-shift tracking and Section 4 summarises and concludes the paper.

2 Proposed method

The block diagram of the proposed method is given in Fig. 1. As shown in this figure, firstly background subtraction is applied to the visible band image. Then, shadows are eliminated. A connected component analysis is used to classify objects as the ones that will be tracked or as noise that will be ignored. Finally, improved mean-shift tracking that contains update condition, re-initialisation of mean-shift trackers, correspondence-based object matching and standard mean-shift tracking with masking the search area is applied and all the objects in the video sequence are tracked. These steps are described in more detail below.

2.1 Background subtraction

As a background subtraction method, an improved adaptive Gaussian model that is a successful, reliable and computationally not very complex method [11] is used. According to this pixel-based background subtraction method, each pixel is defined as a mixture of Gaussians with M components as follows

$$\hat{p}(\boldsymbol{x}|X_T, \text{BG} + \text{FG}) = \sum_{m=1}^{M} \hat{\pi}_m N(\boldsymbol{x}; \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\sigma}}_m^2 I) \qquad (1)$$

where $x^{(t)}$ is the value of pixel at time t, $X_T = \{x^{(t)}, \ldots, x^{(t-T)}\}$ is the training set at time t, whereas T is the time period, BG is

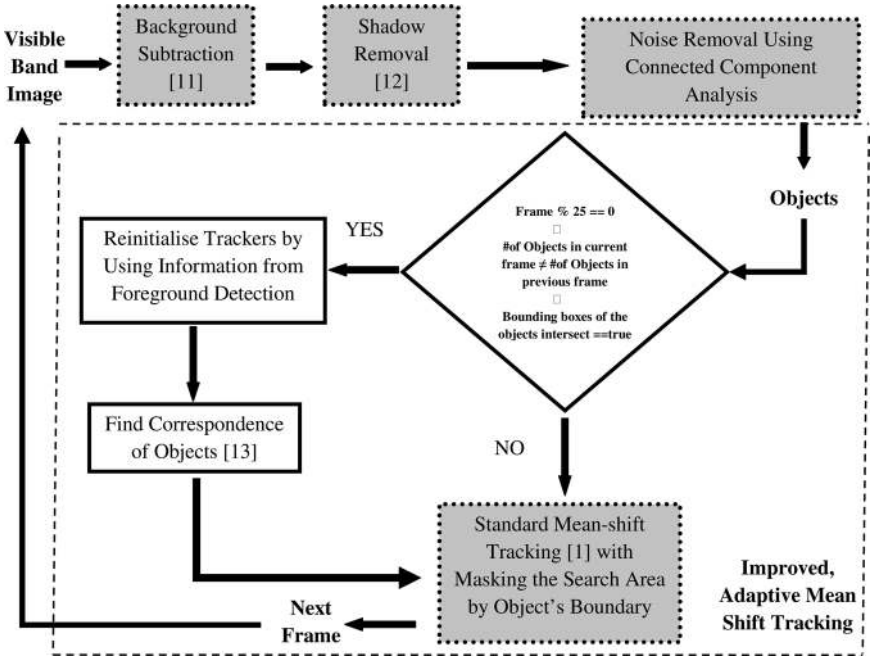

Fig. 1 *Block diagram of the proposed system*

doi: 10.1049/iet-cvi.2011.0054

the background, FG is the foreground, $\mu_1, \mu_2, \ldots, \mu_M$ and $\sigma_1, \sigma_2, \ldots, \sigma_M$ are the estimates of mean and variance for the Gaussian components, respectively. $\pi_1, \pi_2, \ldots, \pi_M$ are the weight values that are non-negative and summation is equal to 1. To provide the adaptation, the parameters of the model are updated with new samples and they are adapted to changes in background. Equations (2)–(4) show how Gaussian model parameters are being updated [11].

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \tag{2}$$

$$\hat{\boldsymbol{\mu}}_m \leftarrow \hat{\boldsymbol{\mu}}_m + o_m^{(t)}(\alpha/\hat{\boldsymbol{\pi}}_m)\boldsymbol{\Delta}_m \tag{3}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\boldsymbol{\pi}_m)(\boldsymbol{\Delta}_m^T\boldsymbol{\Delta}_m - \hat{\sigma}_m^2) \tag{4}$$

where $\boldsymbol{\Delta}_m = x^{(t)} - \mu_m$, $o^{(t)}$ is the ownership and α is the learning parameter, approximately $\alpha = 1/T$, T is the time period. While determining the background and foreground pixels, for each Gaussian component, m is set to 1 if its 'close' component to largest π_m and the others are set to 0. The new sample is 'close' to the component if the Mahalanobis distance between them is less than 4 standard deviations. The square distance from mth component can be calculated by using (5) [11]

$$D_m^2(\boldsymbol{x}^{(t)}) = \boldsymbol{\Delta}_m^T\boldsymbol{\Delta}_m/\hat{\sigma}_m^2 \tag{5}$$

If the new sample is close to the component, then the new sample belongs to 99% confidence level and is determined as a part of foreground.

Although this method is based on mixture of Gaussians because it does not use a fixed number of components, it is proposed as more adaptive and robust when compared with mixture of Gaussian methods and it could automatically select the proper number of components per pixel [11].

2.2 Shadow and noise removal

As a result of background subtraction shadows, which should belong to the background, could be classified as a part of a foreground object. However, shadows might cause problems such as giving rise to merging of objects, distortions of colour histogram of objects and inclusion of background information because the bounding box of the object may become larger and make the following steps performed inaccurately. Therefore a shadow removal step is required to make the tracking more accurate.

In this study, we use a shadow detection scheme that is based on the HSV colour space and presented in [12]. We make use of the fact that shadow cast on a background does not significantly change its hue and saturation information considering shadows decrease the saturation of the pixels. For each pixel that is found as foreground from the background subtraction step, saturation, hue and value components are checked according to (6) and the pixel is classified as the foreground pixel or shadow. (see (6))

where $I^{(t)}(x, y)$ is the pixel that is classified as foreground from the background subtraction step at time t and $B^t(x, y)$ is the background model pixel at time t. H denotes the hue component, S denotes the saturation component and V denotes the value component of a vector in the HSV space. θ is a maximum value for the darkening effect of shadows on the background and β is the upper bound to handle the pixels that the background was darkened too little when compared with the effect of shadows. T_s is the threshold value that defines the upper bound of absolute difference between saturation of pixel and background model. T_H is defined as the upper bound of hue value [12].

After removing shadows, the connected component analysis is applied. By using the connected component analysis in addition to removing noise and detecting the foreground objects that are to be tracked, the segmentation errors that could occur after background subtraction and shadow removal are also eliminated. To remove the noise and errors, the bounding box of the object, the number of pixels that each connected component has and the area of the object's bounding box are found. Then, the density of each object is calculated by using (7)

$$D = N/A_{\text{rect}} \tag{7}$$

where D is the density of object, N is the number of pixels that an object has, A_{rect} is the area of the bounding rectangle.

After finding the density of object, the connected component is classified as an object that will be tracked if its density is greater than the density threshold and number of pixels that belongs to this object is greater than the maximum number of pixel threshold, otherwise the connected component is classified as noise or error and it is ignored.

In Fig. 2, examples of shadow and noise removal steps are shown. In these images, shadows are shown in green and foreground pixels are shown in red.

2.3 Standard mean-shift tracking algorithm

The mean-shift tracking method is an iterative method based on object representation. Additionally, it is an optimisation problem, and uses a non-parametric kernel. It basically tries to find an object in the next image frame that is most similar to the initialised object (object model) in the current frame. Similarity is found by comparing the histogram of

Fig. 2 *Result of shadow removal*

$$p(x, y) = \begin{cases} \text{foreground} & \text{if } \theta \le \dfrac{I^t(x, y) \cdot V}{B^t(x, y) \cdot V} \le \beta \wedge |I^t(x, y) \cdot S - B^t(x, y) \cdot S| \le T_s \wedge D_H \le T_H;\ \theta \quad \text{and} \quad \beta \\ \text{shadow} & \text{otherwise} \end{cases} \tag{6}$$

the object model and the histogram of the candidate object in the next frame.

At the initialisation step, an object model that is to be tracked is selected; bin size, kernel function, size of the kernel and maximum iteration number are determined. Colour histogram of the object model is found and the probability density function (pdf) of the object model is calculated as follows

$$q_u = C\sum_{i=1}^{n} k(||x_i||^2)\delta[b(x_i) - u] \tag{8}$$

In this equation, k is the kernel function that gives more weight to the pixels at the centre of the model, C is a normalising constant that provides the sum of histogram elements is 1, u represents histogram bin and n is the pixel in the object model. δ is the Kronecker delta function and b represents the histogram binning function for pixels at location x_i [1].

After defining the target model in the initialisation step, the candidate model is constructed. Similar to the target model's pdf, candidate model's pdf at location y is given by

$$p_u(y) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta[b(x_i) - u] \tag{9}$$

where h is the kernel size that provides the size of candidate objects and n is the number of pixels [1].

After defining the pdf of candidate model, it is compared with target model's pdf. To compare colour-based pdfs, generally the Bhattacharyya coefficients are used. In (10) $p(u)$ and $q(u)$ represent the Bhattacharyya coefficients where m represents the number of bins

$$\rho[q_u, p_u(y)] = \sum_{i=1}^{m}\sqrt{p_u(y)q_u} \tag{10}$$

The larger ρ means the more similar the pdfs are.

If the candidate model is not similar to the target model then the current search area is shifted. This iteration continues until the result of similarity is less than a threshold or when the iteration number is converged to the predefined number. By applying this method to each video frame, object model can be tracked over time.

However, searching for the new position of trackers in the following frame results in high computational complexity especially when the number of trackers increases. Moreover, a good approximation to the optimal position of the target object may not be found owing to the inclusion of the background pixels into the kernel. Therefore as mentioned in the Section 2.4, decreasing the search area of the mean-shift tracker by using the foreground information increases the tracking accuracy as the search space is restricted. Meanwhile the computational complexity is decreased as smaller number of iterations are required to converge.

2.4 Improved, adaptive mean-shift tracking algorithm

Object detection that could be either manual or automatic is the main step of object tracking. In the literature, many studies use manual object detection and initialisation of the trackers initiated by an operator. However, if manual initialisation is used, since new objects cannot be tracked when they enter the scene after the initialisation frame, it is expected that all objects exist in the starting frame of the video sequences or the operator regularly detects all the new objects. On the other hand, when automatic initialisation is applied, any new object entering the scene could be tracked without any need for a human operator.

In this study, foreground detection is used for automatic initialisation. Firstly, improved adaptive Gaussian background subtraction is applied, then noise and shadow removal methods are executed and objects that will be tracked are determined. In addition to the benefits of using the foreground detection for automatic initialisation of objects, foreground objects are also used as a kernel mask to decrease the search area of the mean-shift tracker. In other words, the bounding boxes of the objects extracted from the foreground detection are used as a mask to make the system more efficient by decreasing the search area to find the new positions of the objects in the next frame. This increased our system's tracking accuracy by reducing incorrect matches since the search space is restricted and there is no need to search the entire frame. Additionally, the required number of iterations to find the new position of object model is decreased.

Although tracking objects by only using the result of foreground detection seems possible, it is not a robust method. When multiple objects are required to be tracked in crowded places and in the presence of occlusions, matching of objects and finding the correspondences become difficult. To solve this problem, in addition to information coming from foreground detection, correspondence-based tracking can be used similar to the one proposed in [13]. However, applying this method in every frame is not efficient as it has high memory requirement to hold the objects and also objects' correspondence objects.

Although using foreground detection in the tracker initialisation step has advantages, it is not sufficient to make the system fully automatic since it still does not detect the new objects entering the scene or the objects leaving the scene. Moreover, the mean-shift tracker only optimises the new position of tracker maximising the Bhattacharyya coefficients (9) between candidate and target objects and therefore cannot adapt the tracker according to the change in object's size or shape. To solve these problems and to handle inclusion of background information, we reinitialise the trackers by using foreground information at regular time intervals. This update mechanism that includes re-initialisation of trackers, an update condition, standard mean-shift tracking with masking the search area by object's boundary and correspondence-based object matching is shown in Fig. 1. To handle the change in size or shape we update mean-shift trackers every 25 frames. To detect new objects as well as objects that leave the scene, numbers of objects in subsequent frames are compared and if those numbers are not equal then mean-shift trackers are updated. To handle occlusion-split between the tracked objects and to detect newly emerging objects; the location of bounding box of each objects are compared. If an intersection exists then mean-shift trackers are refreshed to handle inclusion of front objects colour. However, as a result of re-initialisation, trajectories of objects (object correspondence) are lost. To overcome this, we also find the correspondence of objects after each re-initialisation step.

To establish the matching between objects and provide the correspondence in frames, we adapted the correspondence-based tracking method [13] to our method and used object's size, centre of mass, bounding box and colour histogram features. In this method, the aim is to find the object in the previous frame that is closest and most similar to an object

doi: 10.1049/iet-cvi.2011.0054

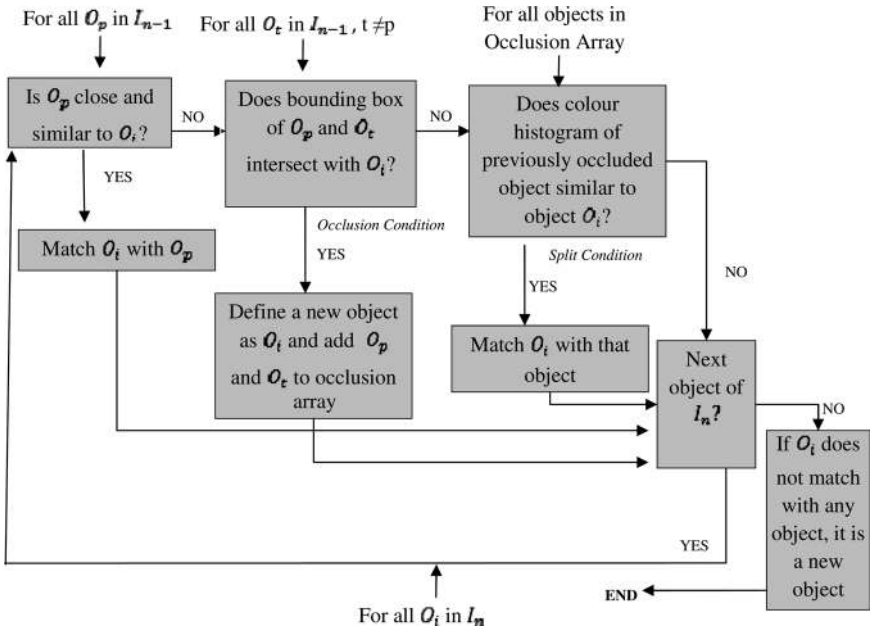

Fig. 3 *Correspondence-based object matching after re-initialisation of mean-shift trackers*

in the current frame. Closeness is defined as the distance between the centre of mass of two objects and Euclidean formula (11) is used to calculate this distance where O_i is an object in the current frame, O_p is an object in the previous frame, $d(O_p, O_i)$ is the Euclidean distance, x represents x and y components of centre of mass of objects O_i and O_p and t_{dist} is distance threshold.

$$d(O_p, O_i) = \sqrt{\sum_{i=1}^{2} (x_{ip} - x_{io})^2} \leq t_{\text{dist}} \tag{11}$$

$$\text{if} \quad S_p > S_i, \quad \frac{S_p}{S_i} \leq t_{\text{size}} \quad \text{otherwise} \quad \frac{S_i}{S_p} \leq t_{\text{size}} \tag{12}$$

Similarity is calculated by using the size ratio of the objects (12) where S_i and S_p are the sizes of O_i and O_p, respectively, and t_{size} is the size threshold.

If the distance between object O_i and object O_p is smaller than the distance threshold t_{dist} and the size ratio of the objects O_i and O_p is smaller than a size threshold t_{size}, we define object O_i as the corresponding object for O_p. Using closeness is a useful criterion as the displacement of an object between adjacent frames is expected to be small. However, it is not a sufficient criterion as objects that are close to each other in the previous frame may be matched incorrectly. Using similarity criterion is also useful as object's scale is not expected to change significantly between consecutive frames [13].

If object O_i cannot be matched to any object in the previous frame, then there are two possibilities: O_i could be a new object or it could have been occluded by another object. To check whether an occlusion between the tracked objects exists or not, we check if there is intersection between the bounding boxes of objects. If bounding box of O_i is overlapping with bounding boxes of two objects in the previous frame, O_p and O_t, then it is highly possible that O_p and O_t were merged to generate object O_i. In such a case, a mean-shift tracker is created to follow O_i and colour histograms of O_p and O_t are stored in the occlusion array to use when a split occurs.

For each detected object, we check whether it is a new object entering the scene or formed after a split. Therefore we check whether its bounding box is overlapping with an object in the occlusion array. If its bounding box is not overlapping with any object in the occlusion array, it is assumed to be a new object and a new mean-shift tracker is defined to track it. Otherwise, we compare its histogram pdf with occluded object's histogram pdfs in order to handle a possible split. To compare histogram pdfs, we use Bhattacharyya coefficients (10), similar to the mean-shift tracking. If there is a similarity, that is, if the distance between pdfs is smaller than a threshold, then the object is matched with the occluded object and that occluded object's histogram is removed from the occlusion array.

The mechanism to establish the matching of objects when a re-initialisation occurs is given in Fig. 3.

3 Experimental results

To evaluate the performance and to verify the robustness of our algorithm we have used both Performance Evaluation of Tracking and Surveillance (PETS) 2006 dataset (video sets 3 and 4) [14] and a dataset that we captured. PETS 2006 dataset contains sequences taken in a real-world public environment and includes busy scenarios such as people walking with their luggage as single or as a part of a larger group, whereas partial occlusions occur. These videos were captured with DV cameras, in PAL standard with a 720×576 resolution and 25 frames/s and compressed as JPEG image format [15]. We have also captured our own dataset to test the proposed method with more challenging scenarios that contain multiple full occlusions between tracked objects. For capturing the videos, Sony HDR-HC1 camera was used and videos were captured at 25 frames/s at 320×240 resolution. We have made these videos publicly available at http://ii.metu.edu.tr/content/visible-thermal-tracking. Using datasets having different types of scenarios helps in evaluating the proposed method with various distances between the camera and the objects of interest, different cameras positions, angles and video resolutions.

While testing the proposed method on both datasets, all the parameters have been set to the values below and kept the same throughout all the tests. To perform background subtraction (Section 2.1) the maximum number of Gaussians (1) was chosen as 4, background model learning rate α (2–4) was taken as 0.0002, threshold on the squared Mahalanobis distance (5) was taken 16 which means 4 standard deviations in order to provide 99% confidence and initial standard deviation was taken as 11. To remove shadows (Section 2.2, (6)) θ, the maximum value for the darkening effect of shadows on the background was chosen as 0.6, β the upper bound of the darkening effect was chosen as 0.9, T_S was defined as 0.6 and T_H was used as 0.9. To remove noise (Section 2.2); the minimum object density not classified as a noise was chosen as 0.4 and number of pixel threshold was used as 1000 (7). YCrCb colour space is preferred as in this colour space luminance and chrominance layers are represented separately. For mean-shift tracking (Section 2.4), a three dimensional (*Y*, *Cr*, *Cb*) histogram is used, histogram bin is taken as $32 \times 32 \times 32$. Additionally, using the number of mean-shift iterations as one is enough to find the best converge of the new location of mean-shift trackers in the following frame because the bounding box of object coming from foreground detection is used as a mask and the search area of the mean-shift tracker is decreased. Distance threshold while finding the correspondence of object is taken as 25 pixels (11), and size threshold is defined as 1.3 (12) for both types of dataset. Experiments show that all the parameters except the distance and size threshold could be used without changing for a variety of scenarios captured with different parameters. On the other hand, although the size and distance thresholds were successful for the various sets that we have tested, in a surveillance system, these parameters could be allowed to be set by the user to make the system more flexible.

In Fig. 4, cases of tracking a group of objects and multiple occlusions between the tracked objects are demonstrated. In this scenario, a group of people marked as object 3 enter the scene and after a while they split, since there is no predetermined occlusion step; as a result of this splitting new objects are compared with the objects previously in the scene, as no match is found and these two new objects are marked as objects 3 and 4. Then, two occlusions occur and the merged objects are marked as 'occlusion'. After splitting they take their old index numbers correctly and the newly entered object that was occluded while entering the field of view is marked with a new index number. Additionally, although objects 2 and 6 are nearly stationary for more than 427 frames, they could still be tracked correctly.

In Fig. 5, an extreme shadow case is presented. In this scenario, by eliminating the shadows, possible FPs are

Fig. 4 *Example result for the proposed method (handling of multiple occlusions between tracked objects, group of people exist) using PETS 2006 S3-T7-A Video 3*

doi: 10.1049/iet-cvi.2011.0054

Fig. 5 *Example result for the proposed method (occlusion between the tracked objects, extreme shadow case) using PETS 2006 S4-T5-A Video 3*

averted. Additionally, the bounding box of the object is extracted more accurately which could affect the tracking adversely by increasing the inclusion of background into the tracker. The bag is also identified as an object when it splits from the owner and the nearly stationary owner and the fully stationary bag are detected and tracked until the end of the scenario. Occlusion between the tracked objects that may cause false tracking is also handled in this case.

In Fig. 6, a crowded scene with multiple partial occlusions between the tracked objects is presented. As it is seen, the proposed method could achieve to track all the objects in the scene without losing their trajectories and handling the partial occlusions successfully.

In Figs. 7 and 8, full occlusion cases are presented. As it is seen, the proposed system could handle full occlusions between the tracked objects and could match the objects correctly once the objects split.

In Fig. 8, the objects 1 and 4 firstly merge when they are handshaking. Then object 4 fully occludes object 1, these objects form a new object marked as object 5, then these two objects split and the correspondences are correctly matched and continue to be tracked as objects 1 and 4.

Similar results have been observed for all tested scenarios of PETS 2006 and the videos that we have captured. The proposed method has also been compared with standard mean-shift tracking. While performing standard mean-shift tracking, a minimum bounding box including all parts of the objects (external box) is selected. This results in some background information also included in the kernel.

In Table 1, the number of objects that are correctly tracked from the beginning to the end is given using the PETS 2006 video types 3 and 4 (which have different camera poses, lighting conditions and camera angles) and some scenarios of our dataset. While executing standard mean-shift tracking, trackers are manually initialised for both approaches and the starting and ending frames are selected as given in Table 1. Starting frames are chosen as the frames consisting of all the objects to be tracked as the standard mean-shift method cannot automatically detect new objects.

As it is seen in Table 1, standard mean-shift tracking when an external box is chosen correctly tracked only two objects in three scenarios containing a total of 49 objects, although the objects were manually initialised. These correctly tracked objects are almost stationary from the beginning to the end. However, objects that are running, walking, carrying a luggage, partially or fully occluded by other objects were not tracked correctly. On the contrary, the proposed method could correctly track all the objects whether running, walking, carrying a luggage or occluded by other objects.

In addition to this comparison, the proposed method has also been compared with standard mean-shift (external box) and naive background subtraction tracking in terms of recall and precision metrics to evaluate the tracking accuracy. For calculating the recall and precision values; the ground truth information that was found in terms of the bounding boxes of objects for each frame and the tracked object's bounding boxes that the proposed tracking system found were utilised. Recall and precision have been calculated using (13) and (14). While determining these metrics true positive (TP), FP and false negative (FN) have been calculated and used as shown in Fig. 9.

All TP, FN and FP values are calculated based on pixel count where TP is the total number of pixels where the ground truth and tracking system agree on these pixels belonging to an object. FN is the total number of pixels that ground truth denotes the pixel as a part of the object while

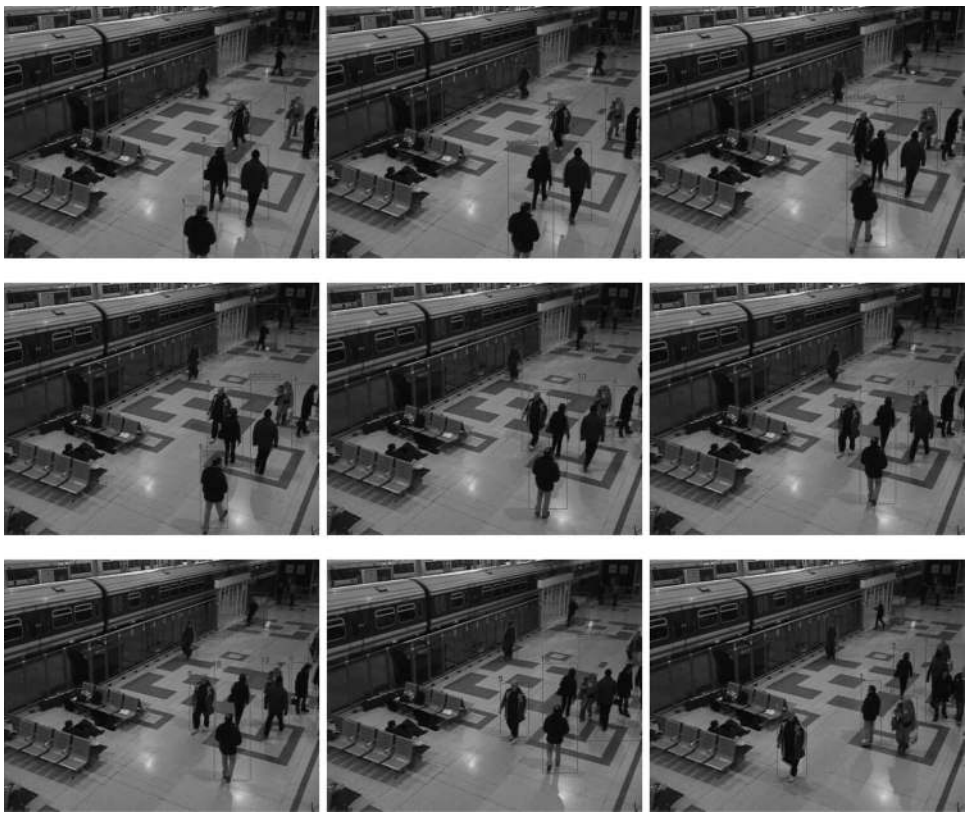

Fig. 6 *An example result for the proposed method (partial occlusions between the tracked objects) using PETS 2006 S7-T6-B Video 4*

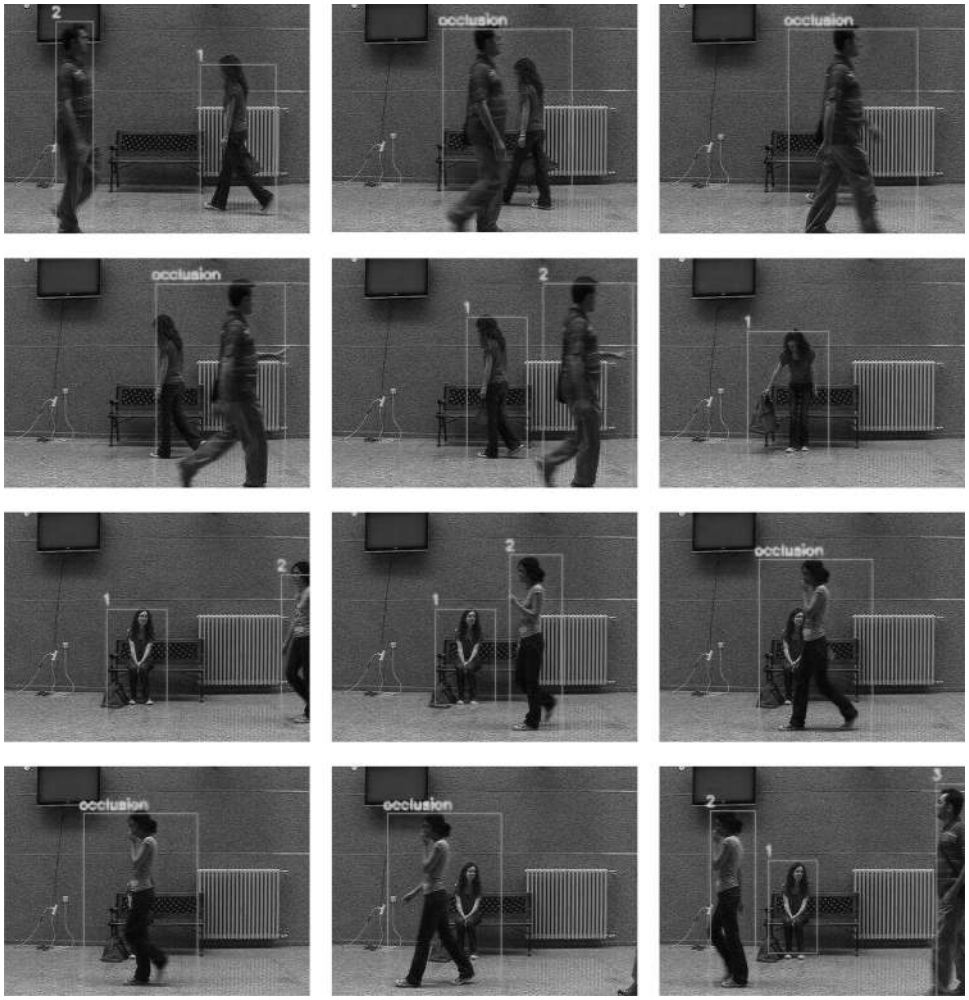

Fig. 7 *Example result (handling of multiple occlusions, tracked object is fully occluded) using Set 6*

doi: 10.1049/iet-cvi.2011.0054

Fig. 8 *Example result for the proposed method (merging of two objects and fully occluded object) using Set 10*

Table 1 Comparison of proposed method and standard mean-shift considering correctly tracked objects from the beginning to the end

Scenario	Frames	No. of moving objects in the scenario	Standard mean shift tracking external box No. of objects that are correctly tracked	Proposed method No. of objects that are correctly tracked
S1-T1-C-Video3	(284–430)	4	0	4
S2-T3-C-Video3	(554–686)	5	0	5
S4-T5-A-Video3	(1237–1786)	3	2	3
S7-T6-B-Video3	(505–1382)	4	0	4
S1-T1-C-Video4	(161–284)	4	0	4
S2-T3-C-Video4	(76–283)	3	0	3
S3-T7-A-Video4	(37–113)	7	0	7
S6-T3-H-Video4	(726–850)	4	0	4
S7-T6-B-Video4	(598–730)	8	1	8
Set 7	(761–984)	2	0	2
Set 10	(95–420)	2	0	2
Set 12	(399–446)	3	0	3

the tracking system cannot detect these pixels as part of an object. FP is defined as the number of pixels that the tracking system finds as an object while ground truth does not agree. In Fig. 9, the dotted area belongs to TP, the cross hatching area belongs to FN and the vertical hatching area belongs to FP while the bounding box drawn in solid lines represents the ground truth bounding box and the one drawn in dashed lines is the bounding box found by the tracking system.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

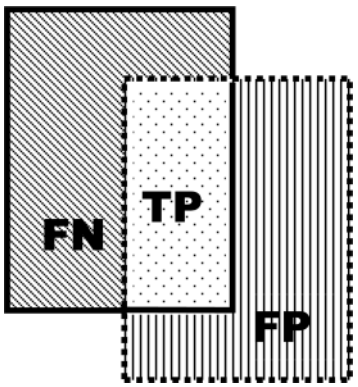

Fig. 9 *Calculation of TP, FP and FNs*

Ground truth bounding box for the object is shown in solid lines while the bounding box found by the tracking algorithm is shown in dashed lines

While calculating recall and precision values in order to compare standard mean-shift (external box) and naive background subtraction tracking with the proposed method, we used scenario *S1-T1-C-Video3* starting from frame number 73 to frame number 191 in order to track a walking object and scenario *S4-T5-A-Video3* starting from 854 to 1004 in order to track a stationary object. Standard mean-shift tracking was initialised manually while the proposed method has been run without any initialisation. For naive background subtraction tracking, the result of the background subtraction step (Section 2.1) is used and to find the trajectory of the objects the correspondence-based object-matching method has been utilised. Plots for tracking recall and precision for these two sequences are given in Figs. 10 and 11.

As it is seen from these figures, the proposed method's performance in overall is better than standard mean-shift and naive background subtraction tracking both for stationary and moving object cases. For the video sequence with moving object, standard mean-shift fails a few frames later, whereas the proposed method can track moving object with 0.89 recall and 0.90 precision on average. On the other hand, naive background subtraction combined with the correspondence-based object matching was more successful than standard mean-shift with 0.88 recall and 0.78 precision on average. For the video sequence with the stationary object, although the standard mean-shift could track the object, the performance of the proposed method is better with a recall of 0.97 against 0.95 and precision of 0.95 against 0.88 on average while the naive background subtraction tracking has 0.92 recall and 0.80 precision on average. On the other hand, while the proposed method can handle occlusions between the tracked objects, performance of the standard mean-shift method degrades significantly in the presence of occlusions.

The proposed method has been evaluated by varying the frequency of the re-initialisation (which is used to handle the change in size or shape). To show the tracker performance, Set13 which includes a person walking with her backpack and *S6-T3-H-Video3* which is a more crowded scene covering occlusions and merge-splits have been used.

The results are given in Fig. 12 in terms of recall (13), precision (14) and *F*-score (15)

$$F\ \text{score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{15}$$

As it is seen from Fig. 12, the average precision, recall and *F*-score values decrease with the increasing re-initialisation frequency (shown as *t* in Fig. 12), although the changes are not significant. It has been observed that the tracker performance is highly dependent on the scene. For instance,

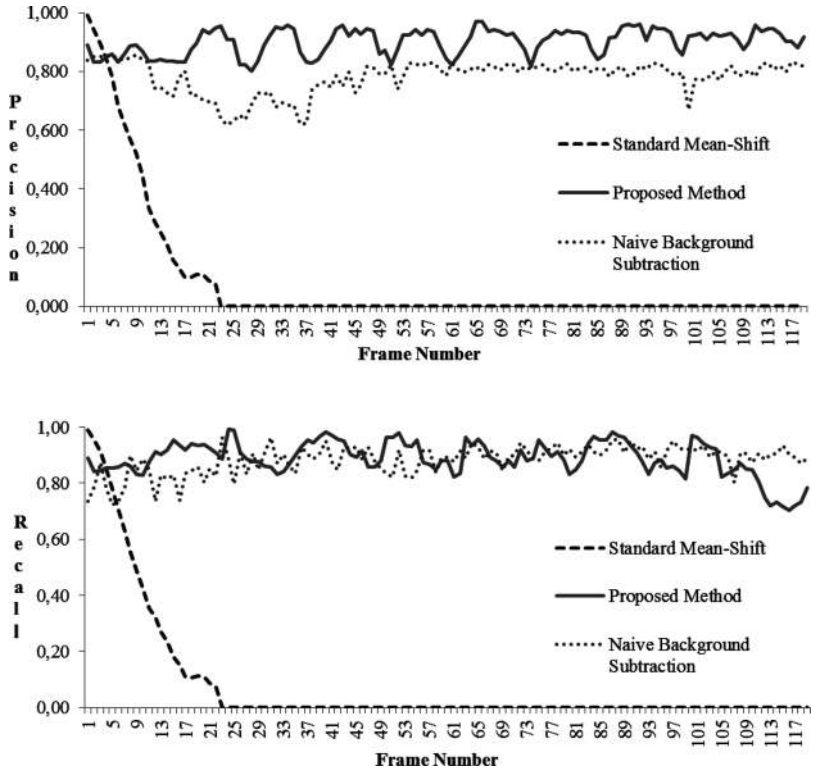

Fig. 10 *Illustration of tracking performance in sequence S1-T1-C-Video3, tracking a walking man*

Recall and precision are plotted against the frame number at top and bottom, respectively

doi: 10.1049/iet-cvi.2011.0054

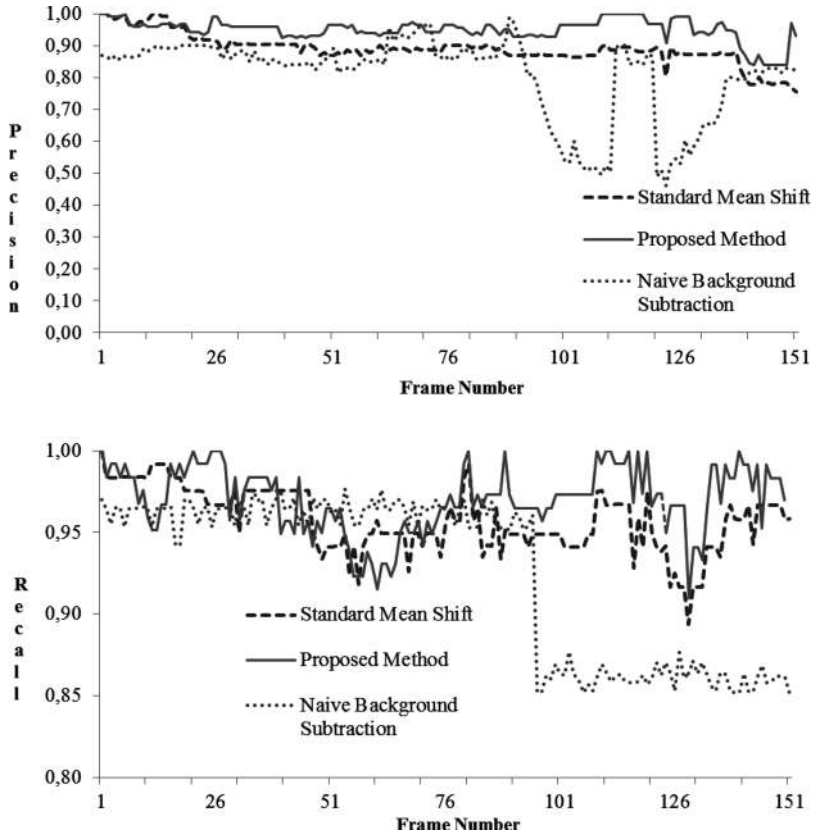

Fig. 11 *Illustration of tracking performance in sequence S4-T5-A-Video3, tracking a stationary man*
Recall and precision are plotted against the frame number at top and bottom, respectively

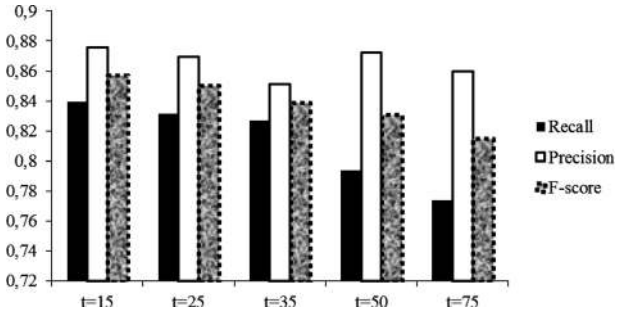

Fig. 12 *Illustration of tracking performance for varying re-initialisation frequencies in terms of average recall, precision and F-score*

the tracker performance is not significantly affected by re-initialisation frequency if the scene is crowded and many occlusions occur. This is because of the trackers being updated automatically before the re-initialisation period when merge-splits happen. On the other hand, if there is a single object or object paths are not crossing, then re-initialisation affects the accuracy as it forces the trackers' size and shape to adapt. However, it is not possible to detect the best re-initialisation frequency as the overall performance depends on the number of objects and their interactions in the scene. Therefore as mentioned before, in our tests, re-initialisation period has been fixed at 25 frames in which the average precision is 0.87, recall is 0.83 and the *F*-score is 0.85. It has to be noted that the re-initialisation period could be increased for crowded scenes without much performance penalty, while it could be reduced if the scene contains only a single object or a few objects with no occlusions between them for better accuracy.

In conclusion, as it is seen from the results, the proposed method successfully detects occlusions and splits which could happen between the tracked objects and finds the correspondence of tracked objects after merging and splitting. Multiple occlusions between tracked objects are also handled. For the occlusions between the tracked objects and anything in the background, a new mean-shift tracker is defined after split occurs and object appears in the field of view as it is done when a new object is detected. By refreshing the trackers, mean-shift tracking becomes adaptive to changes in objects' size and shape. By comparing the numbers of objects in consecutive frames, new objects or objects that are leaving the scene are immediately detected. Using foreground detection in initialisation step and the update mechanism tracking system becomes fully automatic. Additionally, by removing shadows, accuracy is increased and the FPs are reduced.

Since the proposed method is based on the foreground detection, accuracy of this step is important not to cause any FPs. Although we applied connected component analysis to remove noise and it has found to be performing well, in the high noise videos, it is possible that a tracker is initialised for noise falsely detected as a new object. To prevent false tracking, alternative methods such as delaying the tracking of newly detected objects for a predefined number of frames could be applied. Furthermore, while applying background subtraction, using a reference frame that does not contain any moving object makes the segmentation more accurate. To reduce the segmentation errors, the system can be started when the scene is static. In

the proposed method, a new tracker is immediately initialised when a new object enters the scene. However, the object might not be completely in the scene at the time of initialisation and hence the tracker might be initialised using a part of the object resulting in a smaller tracker kernel being used until the kernel is re-initialised which could adversely affect the recall performance. On the other hand, objects having similar colour, size and shape may cause problems especially in the case of occlusions. Therefore as a future work, the proposed method could be improved by using more object features such as using SIFT features in addition to the currently used features.

4 Conclusions

Mean-shift tracking plays an important role in video surveillance systems because of its robustness, ease of implementation and computational efficiency. However, the standard mean-shift algorithm suffers from a number of problems which adversely affect tracking performance and could cause inaccurate or even false tracking. In this study, we proposed a fully automatic multiple objects tracking algorithm based on standard mean shift method that could be used as a part of a static camera video-surveillance system. In the proposed method, foreground detection is used to make the system fully automatic and the bounding boxes coming from foreground detection are used as a kernel mask to decrease the search area of the mean-shift tracker. As a result of this masking, the tracking accuracy is increased and fewer iterations are required to find the new location of object.

By removing shadows, the robustness of tracking mechanism is increased and the FPs are reduced. Objects entering and leaving the scene could all be detected in real-time. By regularly updating the trackers using the foreground information, mean-shift becomes more adaptive to the changes in object size and shape. Occlusion, split and merging scenarios between the tracked objects which are not handled by standard mean-shift are also handled without any human intervention. On the other hand, as the proposed method requires background subtraction, it is not suitable for moving camera scenarios and application area is limited to static camera videos, whereas the standard mean-shift could also be applied to moving camera cases. The proposed method presents an easy to implement, robust and efficient mechanism for automated object tracking in the presence of multiple objects for static cameras, whereas superior to the standard mean-shift by handling its drawbacks.

5 References

1 Comaniciu, D., Ramesh, V., Meer, P.: 'Kernel-based object tracking', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (5), pp. 564–577

2 Birchfield, S.T., Rangarajan, S.: 'Spatiograms versus histograms for region-based tracking'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, CA USA, June 2005, pp. 1158–1163

3 Boonsin, M., Wettayaprasit, W., Preechaveerakul, L.: 'Improving of mean shift tracking algorithm using adaptive candidate model'. Proc. ECTI-CON, Chiang Mai, Thailand, June 2010, pp. 894–898

4 Collins, R.T.: 'Mean-shift blob tracking through scale space'. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), Madison, WI, USA, June 2003, pp. 234–240

5 Qifeng, Q., Zhang, D., Peng, Y.: 'An adaptive selection of the scale and orientation in kernel based tracking'. Proc. IEEE Conf. on Signal-Image Technologies and Internet-Based Systems (SITIS), Shanghai, China, 2007, pp. 659–664

6 Parameswaran, V., Ramesh, V., Zoghlami, I.: 'Tunable kernels for tracking'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), New York, USA, 2006, pp. 2179–2186

7 Quast, K., Kaup, A.: 'Scale and shape adaptive mean shift object tracking in video sequences'. Proc. 17th European Signal Processing Conf. (EUSIPCO), Glasgow, Scotland, 2009, pp. 1513–1517

8 Jiang, Z., Li, S., Gao, D.: 'An Adaptive Mean Shift Tracking Method Using Multiscale Images'. Proc. 2007 Int. Conf. on Wavelet Analysis and Pattern Recognition, Beijing, China, November 2007

9 Porikli, F., Tuzel, O.: 'Multi-kernel object tracking'. Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), Amsterdam, Netherlands, 2005, pp. 1234–1237

10 Pers, J., Kovacic, S.: 'A system for tracking players in sport games by computer vision', *Electrotech. Rev. – J. Electr. Eng. Comput. Sci.*, 2000, **67**, (5), pp. 281–288

11 Zivkovic, Z.: 'Improved adaptive Gaussian mixture model for background subtraction'. Proc. 17th Int. Conf. on Pattern Recognition (ICPR), Cambridge, UK, August 2004, pp. 28–33

12 Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: 'Detecting objects, shadows and ghosts in video streams by exploiting color and motion information'. Proc. 11th Int. Conf. on Image Analysis and Processing (ICIAP), Palermo, Italy, 2001, pp. 360–365

13 Dedeoglu, Y.: 'Moving object detection, tracking and classification for smart video surveillance'. Master's thesis, Bilkent University, Department of Computer Engineering, Turkey, August 2004

14 Performance Evaluation of Tracking and Surveillance (PETS): 2006 Benchmark Data, http://www.cvg.reading.ac.uk/PETS2006/data.html, accessed June 2010

15 Thirde, D., Li, L., Ferryman, J.: 'Overview of the PETS2006 Challenge'. Proc. Ninth IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS), New York, USA, June 2006, pp. 47–50

doi: 10.1049/iet-cvi.2011.0054