Background Subtraction Based on Time-Series Clustering and Statistical Modeling

Ahmed Mahmoud Hamad[1,2] * and Norimichi Tsumura[1]

[1]*Graduate School of Advanced Integration Science, Chiba University, Chiba 263-8522, Japan*

[2]*Department of Information Technology, Faculty of Computers and Information, Menofia University, Egypt*

This paper proposes a robust method to detect and extract silhouettes of foreground objects from a video sequence of a static camera based on the improved background subtraction technique. The proposed method analyses statistically the pixel history as time series observations. The proposed method presents a robust technique to detect motions based on kernel density estimation. Two consecutive stages of the k-means clustering algorithm are utilized to identify the most reliable background regions and decrease the detection of false positives. Pixel and object based updating mechanism for the background model is presented to cope with challenges like gradual and sudden illumination changes, ghost appearance, non-stationary background objects, and moving objects that remain stable for more than the half of the training period. Experimental results show the efficiency and the robustness of the proposed method to detect and extract the silhouettes of moving objects in outdoor and indoor environments compared with conventional methods.

Keywords: background subtraction, motion detection, kernel density estimation, statistical modeling, k-means clustering.

1. Introduction

Motion detection aims to segment moving objects from stationary or quasi-stationary background. Motion detection is the core and the basic step in any system that is related to gesture and activity

*E-mail address: hamad@graduate.chiba-u.jp

recognition, analysis, or monitoring. Motion detection step provides the subsequent stages with the suitable information that decides the way of representing the moving object. Many applications today depend on video cameras and consider the motion detection task as the first step like surveillance, motor traffic analysis, sportive performance evaluation, monitoring patients in hospitals, and so on.

The difficulty of the moving object detection step stems from any change occurs in the weather, illumination, shadow and repetitive motion from clutter.[1)] Also, inherent changes in the background, such as fluctuations in monitors and fluorescent lights, waving flags and trees, and water surfaces may result in incompletely stationary background.[2)] In addition, other difficulties may make the detection task a difficult process such as the homogeneity of the human clothing color and the background textures[3)] and the appearance of the moving object in the initial training period of the video sequence.[4)]

The background subtraction is a simple, obvious, and popular approach to segment the moving object especially under the static background condition. It detects moving objects by selecting a reference image, computing the difference between the current image and the reference image, and thresholding the result. The modeling of the reference image can be categorized to: basic background modeling,[5)] adaptive background modeling,[6,7)] and statistical background modeling.[8–13)]

Statistical background modeling is the most common background modeling technique because of its robustness to overcome most background problems. Several techniques for modeling background statistically have been proposed during past years. A method based on mixture of Gaussians (MOG) was proposed by Stauffer and Grimson[9,10)] for statistic background modeling. MOG method depends on recovering and updating background images based on modeling each pixel as a mixture of gaussians. Although this modeling deals reliably with illumination changes and repetitive motion from clutter, it faced a great difficulty to model a background with fast variations.[14)] Another disadvantage of MOG method is that it tries to solve a two-class classification problem (i.e., foreground and background) with a model of only one of the classes (the model of the background).[15)] In addition, MOG method models the moving object that stops for awhile in the scene as a background object until the

object moves again.

Haritaoglu et al.[16] proposed a real time system (called W4) for detecting and tracking multiple people and monitoring their activities in an outdoor environment. Despite the robustness of W4 in detecting foreground objects even when the background is not completely stationary, W4 could not overcome completely problems like shadow, global illumination changes, and sudden changes in the scene. Jacques Jr et al.[17] improved W4 by applying the normalized cross-correlation to foreground pixels and the candidate shadow pixels are obtained. A refinement process is then applied to further improvement for the shadow segmentation.

Kim et al.[4] introduced a scheme for background modeling based on layered codebook (CB) in a color model or in a gray level model with minor modifications. However, they claimed that layered CB is robust against illumination variations, but it cannot model very slow scene illumination variations in long time.[18] Furthermore, CB has another disadvantage which limits the appearance of the moving object in the initial frames during the training period. In the case that the moving object appears from the first frame in the video sequence, CB codes the moving object as a background object.

In this paper, we propose an efficient method to extract silhouettes from complex backgrounds based on the improved background subtraction technique under the condition of a static camera. Time-series statistical modeling and two stages of k-means clustering are employed to separate and represent most reliable background regions. Each pixel is formulated as time series observations over t frames; where t represents the number of frames used in the training period. Due to varying effects of the noise and the illumination on each pixel, a single global threshold is not an efficient way to classify pixels. The proposed method uses a threshold for each pixel. A set of morphological processes are utilized to handle foreground pixels. The proposed method presents an effective updating mechanism that exploits color and shape features of each detected object to cope with difficulties like sudden illumination changes, the appearance of ghosts, and non-stationary background objects. A ghost is a detected object that doesn’t correspond to a real moving object, it may appear as result of a physical

change in the background scene or a stable moving object that erroneously classified as background object then moves in the scene.[19)]

This paper is organized as follows: Section 2 describes how the background can be modeled and updated. Section 3 presents how to obtain silhouettes of foreground objects. Section 4 proposes the object based updating mechanism. Section 5 presents the experimental results. Finally, Section 6 concludes the proposed method with some future work guidelines.

2. Proposed Background Modeling

The heart of any background subtraction algorithm is the construction of a model that describes the background state of each pixel. In many surveillance and tracking applications, several assumptions are necessary during the training phase to build the background image for example the assumptions imposed on the generation of initial parameters of the background model. Another example of assumptions imposed by some surveillance and tracking applications is the absence of moving objects from the training phase.[20)] This requirement is difficult or sometimes impossible because of the difficulty to control the monitored area.

The proposed method builds the background model in the presence of moving objects during the training period. In order to obtain a clear background image, the proposed method assumes that the stable moving object appears less than a half of the training period (3–5 s). In the case that the moving object is stable for most of the training time or any other disturbances for the modeling of the background occur such as small background object motion (e.g., swaying tree branches); sudden illumination changes (e.g., indoor light on and off); physical changes in the background, an efficient mechanism that based on spatial and color features of the detected moving object is called to update the background model instantaneously. Furthermore, a running average is used to update each background model pixel to overcome the gradual illumination changes. The proposed method operates on both RGB image sequence and grayscale image sequence as well. Figure 1 illustrates that the proposed

method consists of three main phases. The proposed background modeling phase aims to construct the initial background model and the first threshold image. The first background model and the first threshold image are the input for the silhouette acquisition phase. The proposed updating mechanism that attempts to construct the second background model and the second threshold image is the third phase of the proposed method.

The proposed method formulates the history of each pixel (x, y) as time series observations over t frames, where t represents the number of frames used in the training period (usually 90–150 frames) as shown in Fig. 2. These time series observations construct an array $v_t^c(x, y)$ such that

$$v_t^c(x, y) = \{I_1^c(x, y), I_2^c(x, y), \cdots, I_t^c(x, y)\}, \tag{1}$$

where $I_t^c(x, y)$ is the intensity value of the pixel (x, y) at frame t and for RGB channel c.

The array $v_t^c(x, y)$ consists of the recent history of the pixel (x, y) over the time. The array $v_t(x, y)$ replaces $v_t^c(x, y)$ if the history is for grayscale pixels or working with only one RGB channel. This history may be for a background pixel which is the case most of the time or a background pixel that occluded by moving objects for some time. Figure 3(b) shows the unimodal intensities representation for a background pixel history over one RGB channel shown in Fig. 3(a). The variance of this pixel history is small. Figure 4(b) shows the bimodal intensities representation of a combination of a background pixel that is occluded by a human movement over one RGB channel as shown in Fig. 4(a).

Figures 3 and 4 illustrate that the variation of the pixel history in the case of a moving object occlusion is significant. This obvious variation allows the clustering technique to easily classify the background and the foreground regions. The behavior of RGB pixel intensities can be replaced by the behavior of the grayscale pixel intensities, but operating with color sequence leads to avoid loss of image sequence information and gives better results.

2.1 Initial background image construction

The construction of the background model is the key factor in the background subtraction process. The ideal background scene is realized if intensity values of each pixel in an image sequence remain constant over time. This condition couldn't be met such that each pixel is exposed to different effects such as moving objects, noise, illumination changes, etc. The proposed method aims to determine the most reliable background observations over each pixel history.

The proposed method models each pixel history by four statistical values; the mean $\mu^c(v_t^c(x,y))$ (for simplification μ^c), the minimum, the maximum, and the standard deviation $\sigma^c(v_t^c(x,y))$ (for simplification σ^c) of the pixel history for each RGB channel c. The mean value is used to represent the deviation of the pixel history (due to the camera noise) in the background image. The mean and the standard deviation of a pixel history for each RGB channel are estimated as

$$\mu^c = \frac{1}{t}\Sigma_{i=1}^{t} I_i^c(x,y),\ \sigma^c = \sqrt{\frac{1}{t}\Sigma_{i=1}^{t}(I_i^c(x,y)-\mu^c)^2}. \tag{2}$$

2.1.1 Motion detection

To detect if there is a motion occurred, the proposed method employs kernel density estimation (KDE) to determine the stability of each pixel history. KDE is used to estimate the probability of cumulative density function (CDF) at each pixel history based on the number of samples in this history. CDF estimates the area under KDE function up to that value of the pixel history mean μ^c. This probability is the key to detect if $v_t^c(x,y)$ contains motion or not. KDE is a non-parametric way of estimating the probability density function of a random data or population. Kernel estimator function is chosen to be a normal function $N(0;\sigma^{c^2})$, where σ^{c^2} represents the kernel function bandwidth, then the density functions can be estimated as follows[14)]

$$P(I_n^c) = \frac{1}{t}\Sigma_{i=1}^{t}\frac{1}{2\pi|\sigma^c|^{1/2}} exp[^{-\frac{1}{2}(I_n^c-I_i^c)^T\sigma^{c^{-1}}(I_n^c-I_i^c)}], \tag{3}$$

where $I_n^c(x,y)$ is the RGB channel c intensity value of pixel (x,y) at frame n.

When monitoring the intensity value of a pixel over time in a static scene (i.e., with no motion),

the pixel intensity can be reasonably modeled with a normal distribution $N(\mu^c;\sigma^{c^2})$.[14] Therefore, we can conclude that CDF(μ^c)≃ 0.5 as shown in Fig. 5(a). In the case that the pixel history occluded by a motion, the normal distribution is skewed and CDF of the mean is deviated to be less than 0.5 [i.e., CDF(μ^c)≪ 0.5] as shown in Fig. 5(b). As the background pixel exposed to a camera noise and gradual illumination changes, CDF of the background pixel history mean may be deviated a little. This deviation is denoted in the following context by δ.

2.1.2 Background model representation

No motion detected situation is realized if the CDF of μ^c is deviated from 0.5 by less than or equal to δ [i.e.$0.5\text{-}\delta \leq CDF(\mu^c) \leq 0.5+\delta$]. The range ϕ^c of $v_t^c(x,y)$ is employed to threshold coming new pixels as described later. This range is estimated as

$$\phi^c(x,y) = \max(v_t^c(x,y)) - \min(v_t^c(x,y)). \tag{4}$$

The mean of $v_t^c(x,y)$ is the most reliable value to represent the deviation of the pixel history (due to the camera noise) in the background image. Therefore, the background image for each RGB channel c at position (x,y) is formulated as

$$Bg_1^c(x,y) = \mu^c(v_t^c(x,y)). \tag{5}$$

The threshold image for each RGB channel c at position (x,y) is formulated as

$$Th_1^c(x,y) = \phi^c(x,y) + \phi^c(x,y) \cdot \frac{1}{\sigma^c(x,y)}, \tag{6}$$

where $1/\sigma^c(x,y) \cdot \phi^c(x,y)$ is added to the range to allow the detection step to be adapted to camera noise and limited changes in the illumination (like turning on a desk lamp or a car light). The value $1/\sigma^c(x,y)$ is a weight to handle the problem of small range of pixels history which is the case for modern cameras image sequences.

The second possibility comes true if CDF(μ^c) is much less 0.5. It means that the background pixel is subjected to a movement by an object and may be to consequent changes like shadows, highlights,

and so on. In such cases, the proposed method employs K-means clustering algorithm to partition each pixel history $v_t^c(x,y)$ into two clusters ($k=2$) which represent the background cluster and the foreground cluster. The seeds of the two clusters are the minimum and the maximum values of $v_t^c(x,y)$. Under the assumption that most of the time pixels belong to the background, we concluded that the cluster whose duration is longer should be classified as a background cluster. This is the first classification stage which handles the situation when there are one background cluster and one foreground cluster as shown in Fig. 4.

In another situation, more foreground clusters that are darker and brighter than the background observations are found as shown in Fig. 6. To handle this situation, the motion detection module is recalled again to detect if the background cluster contains motions. This cluster is denoted as $BC^c(x,y)$ in the following context. If the result of the motion detection module is that $BC^c(x,y)$ is a background cluster (contains no motions), this cluster is denoted as $B^c(x,y)$. On the other hand, if the result of the motion detection mechanism is that $BC^c(x,y)$ contains motions, K-means clustering algorithm is recalled again to partition $BC^c(x,y)$ cluster into two clusters. The cluster with longer duration is classified as background and is denoted as $B^c(x,y)$. After the second classification stage, the reliable background region $B^c(x,y)$ is extracted and the background image for RGB channel c is set at position (x,y) to the mean of the cluster $B^c(x,y)$ such that

$$Bg_1^c(x,y) = \mu^c(B^c(x,y)). \tag{7}$$

The range ϕ_b^c of the background cluster and the threshold image for each RGB channel c at position (x,y) are formulated as

$$\phi_b^c(x,y) = \max(B^c(x,y)) - \min(B^c(x,y)), \tag{8}$$

$$Th_1^c(x,y) = \phi_b^c(x,y) + \phi_b^c(x,y) \cdot \frac{1}{\sigma_b^c(x,y)}. \tag{9}$$

At the end of the training phase, two images are generated; the background image and the threshold image. The proposed method generates accurate and clear background image in the presence of challenges like non-stationary background object, ghosts appearance, and sudden illumination changes but under the assumption that most of the time pixels belong to the background. On the contrary, if this assumption is not met, the proposed method relies on the proposed object based updating mechanism to cope with these challenges. Figures 7–9 show the clear background images restored from different scenarios. The first scenario includes a complex background that occluded by a human that enters the scene and remains stable for awhile as shown in Fig. 7(a). Figure 7(b) clarifies that the proposed background modeling method can restore the background image accurately in the condition that the human stops for awhile in a complex scene. In the second scenario, a walking person appears on the training period from the first frame till the last frame as shown in Fig. 8(a). Figure 8(b) shows that the background image could be accurately generated on the appearance of the moving object from the first frame to the last frame in the training period. Therefore, the proposed background modeling method does not require any previous conditions on the appearance of the moving objects during the training period. Finally, the third scenario displays an example of non-stationary background objects as shown in Fig. 9(a). Figure 9(b) shows the background image restored from an image sequence of a man walking with the movement of a tree in the background (incomplete stationary background).

2.2 Pixel-based updating

Any change occurs in the background leads to a change in the background statistics and consequently a change in the background model and the threshold image. To cope with gradual illumination changes, the proposed method updates the statistics of each pixel in the background model if the new coming pixel I^c_{new} is classified as background such that

$$\mu^c_{i+1}(x,y) = \alpha \cdot I^c_{new}(x,y) + (1-\alpha) \cdot \mu^c_i(x,y), \tag{10}$$

$$(\sigma^c_{i+1}(x,y))^2 = \alpha \cdot (\mid I^c_{new}(x,y) - \mu^c_i(x,y) \mid)^2 + (1-\alpha) \cdot (\sigma^c_i(x,y))^2, \tag{11}$$

$$\phi_{i+1}^{c}(x,y) = \alpha \cdot (\mid \mu_{i+1}^{c}(x,y) - \mu_{i}^{c}(x,y) \mid) + (1-\alpha) \cdot \phi_{i}{}^{c}(x,y), \quad (12)$$

where α =0.1.

3. Silhouette Acquisition

Like any background subtraction technique, the step that follows the construction of the background image is the foreground acquisition. The common strategy of foreground acquisition is to subtract the upcoming frame from the background image. The proposed method performs the subtraction process in pixel-wise according to

$$d_1^c(x,y) = |I_n^c(x,y) - Bg_1^c(x,y)|, \quad (13)$$

where $I_n^c(x,y)$ is RGB channel c intensity value of the pixel (x,y) of the new frame n. The value $d_1^c(x,y)$ represents the distance between the coming frame and the background image of each RGB channel c at position (x,y). The classification of the pixel (x,y) to a background pixel which is represented by 0 or a foreground pixel which is represented by 1 is formulated as

$$Silh(x,y) = \begin{cases} 0 & if \quad \frac{1}{3}\Sigma_{c=R,G,B} d_1^c(x,y) < \frac{1}{3}\Sigma_{c=R,G,B} Th_1^c(x,y) \\ \\ 1 & if \quad \frac{1}{3}\Sigma_{c=R,G,B} d_1^c(x,y) \geq \frac{1}{3}\Sigma_{c=R,G,B} Th_1^c(x,y) \end{cases} . \quad (14)$$

Subsequent to background-foreground separation, each video frame is represented by two values (i.e., 0 and 1). The proposed method applies a set of morphological operations to extract the silhouette. An opening operation followed by a closing operation is performed to remove the noise presented in each frame. To preserve the edges of the silhouette, the closing operation is performed again. Figure 10 illustrates the steps followed to extract silhouettes using morphological operations

4. Proposed Object-based Updating Mechanism

Object-based updating mechanism (OUM) follows the silhouette extraction process. The pixel history deviations show different shapes from pixel to pixel in the presence of moving objects. A

single statistical distribution cannot model such deviations nor updating background image in the presence of difficulties like illumination changes, ghost appearance, and non-stationary background. The proposed method relies on shape and color features of each detected object to robustly update the background and threshold images then modifies the silhouette extraction module. OUM aims to cope with challenges like sudden illumination changes, ghost appearance, and non-stationary background objects. Figure 11 shows examples of such challenges.

4.1 Updating background and threshold images

The cue of OUM is to locate the starting position of each detected object. The starting position for regular moving objects begins outside the scene then passes in the front of the camera to outside the scene. If the detected object starts its motion from inside the scene, this is considered an irregular situation. Consequently, this object and its motion are analysed to realize if that object is a real moving object, a ghost, or a moving background object. Spatial and color features are extracted for each detected object. Due to the repetitive occurrence of the moving background objects (e.g., tree branches) or repetitive switching light on and off, OUM applies the updates to a copy of the background and the threshold images which are denoted $Bg_2^c(x,y)$ and $Th_2^c(x,y)$, respectively. The input for OUM is the binary frame resulted from the silhouette extraction module and the corresponding color frame.

OUM uses spatial features: the object skeleton centroid, area, and bounding box to characterize each detected object shape. The centroid of an object is influenced by large motions of extremities[16] while the object skeleton centroid isn't influenced by such extremities. The parameters of the bounding box are the link between the binary frame and corresponding color frame such that these parameters are used to locate the detected object in the corresponding color frame.

Color moments are the features extracted from the corresponding region of the detected object in the color sequence. The first order (mean), the second order (variance) and the third order (skewness) color moments have shown its efficiency and effectiveness in representing color distributions of images. Three moments for each RGB channel leads to a feature vector that contains nine values. For

simplification, the detected object in binary frame will be denoted BO_i^j, the corresponding object in the color frame CO_i^j, and the centroid of BO_i^j skeleton as x_i^j (where *i* represents the frame number and *j* identifies object number).

The key factor of OUM is to specify if BO_i^j initiates its motion inside the scene or it exists in the previous frames. Euclidean distance $dist_i^j$ between x_i^j and the centroids of detected objects in previous frame *i-1* is estimated. If $dist_i^j$ is less than $r\sqrt{2}$ (BO_i^j is searched in block of size *2r* x *2r* centred at x_i^j in the previous frame *i-1*), bonding box parameters are used to locate CO_i^j and the detected object within $r\sqrt{2}$ block in the frame *i-1* which is denoted DCO_{i-1}^j. Color moments features are computed to construct the feature vector for both objects. Euclidean distance is computed again between CO_i^j feature vector and DCO_{i-1}^j feature vector. The previous step is employed to realize if DCO_{i-1}^j is the same as CO_i^j but in the previous frame and it is not a different object. If the distance is less than a threshold T_1, this means that BO_i^j exists in frame *i-1* and no update is required to the background image. On the contrary, if BO_i^j is not exist in frame *i-1*, we conclude that BO_i^j initiates its motion from frame *i*. Consequently, the motion status of BO_i^j is to be analysed as the following scenarios:

(a) Non-stationary Background Objects: The bounding box parameters are employed to locate CO_i^j in the first background model $Bg_1^c(x,y)$. Eight neighbour blocks to CO_i^j in $Bg_1^c(x,y)$ (with the same size of CO_i^j) are generated. Color moment features are used to characterize CO_i^j and the eight neighbour blocks. Euclidean distance is computed between moment features of CO_i^j and moment features of eight neighbour blocks. If the distance is less than a threshold T_1 for one or more of neighbour blocks, therefore BO_i^j is a part of non-stationary background (e.g, swaying tree branches). In this case, the background and the threshold values of the closest block matching CO_i^j updates $Bg_2^c(x,y)$ and $Th_2^c(x,y)$. Figure 12 shows an example of updating the background image in the case of non-stationary background tree branches. In the case that there is no match between the moment features of CO_i^j and the moment features of the eight neighbour blocks the decision is that BO_i^j is a real moving object.

(b) Ghost Appearance: The ghost may appear as a result of moving object that remains constant most of the training time and erroneously classified as background or as a result of a sudden physical change at the background as shown in Fig. 11. If the centroid x_i^j appears constant for N s. The area and color moment features for objects in the frame that precedes stationary status are computed. Euclidean distance is used to measure the distance between the feature vector of BO_i^j and CO_i^j and the feature vectors of every object in the frame that precede N s. If the distance is less than a threshold T_2, we can conclude that BO_i^j is a moving object that stops for sometime. Whereas, if the distance is greater than a threshold T_2 this means that a sudden physical change occurs in the background. As a result of a ghost appearance, OUM recalls the background modeling technique discussed earlier in section *2* to update Bg_2^c and Th_2^c.

(c) Sudden Illumination Changes: OUM adapts to sudden changes in background illumination (like switching light on and off) instantaneously. If the area of BO_i^j is greater than 80% of the scene, OUM responds quickly and calls the background modeling technique to update Bg_2^c and Th_2^c.

4.2 Modification of silhouette acquisition process

Silhouette acquisition process described in section 3 depends on subtracting the upcoming frame from only the background image $Bg_1^c(x,y)$ obtained after the training period. OUM constructs an updated version of $Bg_1^c(x,y)$ and $Th_1^c(x,y)$. So, the silhouette acquisition process should be modified to handle the new constructed images $Bg_2^c(x,y)$ and $Th_2^c(x,y)$.

OUM subtracts the upcoming new frame from both $Bg_1^c(x,y)$ and $Bg_2^c(x,y)$ and the result is stored in $d_1^c(x,y)$ and $d_2^c(x,y)$ respectively. The values $d_1^c(x,y)$ and $d_2^c(x,y)$ are thresholded by $Th_1^c(x,y)$ and $Th_2^c(x,y)$ respectively on pixel-wise basis. The pixel is classified as foreground if both $d_1^c(x,y)$ and $d_2^c(x,y)$ is greater than or equal $Th_1^c(x,y)$ and $Th_2^c(x,y)$. This modification is applied to eq. (14) as follows

$$Silh(x,y) = \begin{cases} 1 & if((\frac{1}{3}\Sigma_{c=R,G,B} d_1^c(x,y) \geq \frac{1}{3}\Sigma_{c=R,G,B} Th_1^c(x,y)) and \\ & (\frac{1}{3}\Sigma_{c=R,G,B} d_2^c(x,y) \geq \frac{1}{3}\Sigma_{c=R,G,B} Th_2^c(x,y))) \\ 0 & otherwise \end{cases} . \quad (15)$$

5. Experimental Results

The robustness and the effectiveness of the proposed method are presented in this section. To consider a variety of moving objects and different critical scenarios, a variety of videos from different datasets are utilized; i2r dataset,[21] VSSN 2006 dataset,[22] shadow detection dataset,[23] and keck gesture dataset.[24] The proposed method is implemented using Matlab 7.6.0 on a PC with 2.1 GHz speed and 2GB memory. The training period is set to 3–5 s (90–150 frames). We ran a series of experiments on different datasets to determine the best values of the following parameters; δ is estimated to be 0.02, N is set to 0.5 s (15 frames), and for normalized feature vector, T_1 and T_2 are estimated as 0.3 and 0.25, respectively. Results of the proposed method are analysed qualitatively and quantitatively against MOG and CB methods.

5.1 Qualitative analysis

We conduct some experiments to test the proposed method in different situations. Firstly, the proposed method is applied to videos including challenges like switching the room light on and off, ghost appearance, and non-stationary background. OUM adapts to the changes in the background by updating the background and the threshold images. This updating relies on the shape and the color features of each detected object. Figures 13(a) and 13(b) show the efficiency of the proposed method in updating background images and the extraction of silhouettes against difficulties such as sudden illumination changes and ghost appearance. Figure 13(c) is a complex scene that includes non-stationary background and ghost appearance. The black rectangle occluded on the first background image specify

the position of a parked car that remain more than the half of the training period. After the movement of this parked car, OUM detects a ghost that appears in the position of the moving car. Therefore, OUM updates the second background image. *The silhouette extracted* column in Fig. 13 represents the silhouettes of the moving objects in the *reference image 1* or *reference image 2* columns.

Figure 14 shows the comparison among the results of the proposed method and two commonly used baseline methods: MOG and CB. We implemented MOG method with three Gaussian components ($K=3$). The threshold used to identify the matched component is defined as the Gaussian components standard deviation scaled by 2.5. The threshold that is used to identify the components used to model the background is estimated to be 0.25 and α: learning rate is set to 0.007. Kim et al. presented CB-BGS program[25] to extract moving foreground objects from the background scene.[4] The background image trained using CB method is constructed by training the same number of frames as the proposed method. Other CB parameters are set to the default values defined by the program.

Figure 14 shows the robustness of the proposed method to detect and extract silhouettes in addition to restore clear background images in different applications like gesture detection and analysis [as shown in Fig. 14(d)] and traffic monitoring [see Fig. 14(a)], and so on. Figure 14(a) shows that the proposed method extracts the silhouettes of different moving objects (for example human, cars, etc). Besides that, the proposed method doesn't provide any limitations on the appearance of the moving objects in the initial training period.

The proposed method copes with the problem of non-stationary background. OUM detects the moving background objects and updates $Bg_2^c(x, y)$ for all pixels of those objects. Figures 14(b) and 14(c) show examples of incompletely stationary background and the robustness of the proposed method to extract silhouettes of real moving objects

The proposed method copes with the moving objects stability problem. If the moving object remains stable most of the training period then it moves, OUM detects the ghost appeared and updates the background image as shown in Fig. 13(c). In the case that the moving object moves then stops

after the training period, OUM detects that this object is a real moving object. On the contrary, MOG models this stationary object as a background object so that it gives the worst results in Figs. 14(c) and 14(d). Figure 14(e) shows the efficiency of the proposed method to restore a clear background image even if the scene contains a complex background and extract the silhouettes of the moving objects from this complex background.

5.2 Quantitative analysis

Varcheie et al. [26] used two metrics to evaluate the performance of his method (called RECTGAUSS-Tex) for background subtraction. These two metrics are true positive rate (TPR) and false positive rate (FPR). The method with the highest TPR and the lowest FPR will be the best background subtraction technique.[26] We compare the proposed method to MOG and CB based on TPR and FPR results. Wallflower dataset which contains ground-truths is used in the evaluation process with six different situations .[27] Figure 15 shows the TPR and FPR obtained for MOG, CB, and the proposed method which evaluated by using wallflower dataset. Figure 16 shows silhouettes extracted by the proposed method for the wallflower dataset.

As noted in Fig. 15(a) that the proposed method has the highest TPR average. This indicates that the number of real foreground pixels detected in the extracted foreground is much larger than the number of real foreground pixels that are detected in the background.[26] Also, Fig. 15(b) the proposed method has the lowest FPR average among MOG and CB methods. This shows that our method combines a high TPR and a small FPR.

As a summary of the previous experiments, the proposed method proves that it is an efficient and a robust method to extract silhouettes from a video sequence under the condition of static or quasi-stationary camera. It succeeds to overcome the problems of the sudden and gradual illumination changes, the slow motion of background objects for outdoor and indoor environments, and the stability of the moving objects. In addition, the proposed method efficiently restores the complex background image for both outdoor and indoor scenes.

6. Conclusions

In this paper, we proposed a simple and robust method to detect and extract moving object silhouettes from a sequence of video frames of a static camera. The proposed method is based on the background subtraction technique. Each pixel history is modeled using four statistical values; the mean, the standard deviation, the minimum, and the maximum. Two stages of K-means clustering technique are employed to identify the most reliable background regions. Pixel and object based updating mechanism cope with challenges like gradual and sudden illumination changes, ghost appearance, and non-stationary background objects. Experimental results showed the efficiency and the effectiveness of the proposed method under indoor and outdoor scenarios.

The limitation of this method is the homogeneity between moving object and background colors. This is a problem of most of background subtraction technique. Also, a local illumination change like shadows is still a difficulty for the proposed method. For the future works, we aim to adapt the proposed method to cope with this problem.

References

1) L. Wang, W. Hu, and T. Tan: Pattern Recogn. **36** (2003) 585.
2) A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu: Mach. Vision Appl. **20** (2009) 395.
3) W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis: Proc. IEEE Int. Conf. on Computer Vision (ICCV'09), 2009.
4) K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis: Real-Time Imaging **11** (2005) 172.
5) N. Friedman and S. Russell: Proc. 13th Conf. Uncertainty in Artificial Intelligence (UAI'97),1997, p. 175.
6) N. McFarlane and C. Schofield: Mach. Vision Appl. **8** (1995) 187.
7) C. Ridder, O. Munkelt, and H. Kirchner: Proc. Int. Conf. Recent Advances in Mechatronics, 1995, p. 193.
8) C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland: IEEE Trans. Pattern Anal. Mach. Intell. **19** (1997) 780.
9) C. Stauffer and W. Grimson: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99), 1999, p. 246.
10) W. Grimson, C. Stauffer: R. Romano, and L. Lee: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'98), 1998, p. 22.
11) H. Lin, T. Liu and J. Chuang: Proc. IEEE Int. Conf. on Image Proc. (ICIP'02), 2002, p. 893.
12) J. Wang, G. Bebis, and R. Mille: IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum in conjunction with CVPR, 2006.
13) A. Tavakkoli, M. Nicolescu, and G. Bebis: Proc. 2nd Int. Symposium on Visual Computing (ISVC'06), 2006, p. 40.
14) A. Elgammal, D. Harwood, and L.S. Davis: Proc. of 6th European Conf. on Computer Vision (ECCV'00), **2**(2000), p. 751.
15) P.J. Withagen, K. Schutte, and F. C. Groen: Proc. IEEE Int. Conf. Patt. Recogn. (ICPR'04), 2004, p. 31.
16) I. Haritaoglu, D. Harwood, and L. Davis: IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 809.
17) J.C.S. Jacques Jr, C.R. Jung, and S.R. Musse: Proc. of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05), 2005, p.189.
18) M.H. Sigari and M. Fathy: Proc. Int. Multi-Conf. of Engineers and Computer Scientists, 2008.
19) R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti: Proc. Intelligent Transportation Systems Conf., 2001, p. 334.
20) B. Gloyer, H. Aghajan, K. Siu, and T. Kailath: Proc. Of SPIE Symposium on Electronic Imaging: Image and Video Processing , 1995.

21) L. Li, W. Huang, I. Gu, and Q. Tian: Proc. of IEEE Trans. Image PROCESS **13** (2004) 1459.

22) *http* : //*mmc*36.*informatik.uni* − *augsburg.de*/*VSSN*06_*OSAC*/

23) Z. Lin, Z. Jiang, and L. S. Davis: Proc. of IEEE 12th Int. Conf. on Computer Vision (ICCV'09), 2009, p. 444.

24) M. Trivedi, I. Mikic, and G. Kogut: Proc. of IEEE Conf. on Intelligent Transportation Systems, 2000, p. 155.

25) *http* : //*www.umiacs.umd.edu*/*knkim*/*UMD* − *BGS*/*index.html*

26) P.D.Z. Varcheie, M. Sills-Lavoie, and G. A. Bilodeau: Sensors **10** (2010) 1041.

27) K. Toyama, J. Krumm, B. Brumitt, and B. Meyers: Proc. of Int. Conf. on Computer Vision (ICCV'99), 1999, p. 255.

Figure Captions

Fig.1 Steps of the proposed method.

Fig.2 Pixel (x, y) history over six frames.

Fig.3 Pixel intensity values over t frames for A background pixel history such that: (a) Frame samples with black circles represent samples of pixel history at position (20,110) and (b) Plotting of pixel history.

Fig.4 (Color online) Pixel intensity values over t frames for background and foreground pixel history such that: (a) Frame samples with white circles represent samples of pixel history at position (80,100) and (b) Plotting of pixel history.

Fig.5 Cumulative density function for (a) A background pixel history and (b) A background pixel history that occluded by a motion.

Fig.6 (Color online) Pixel intensity values over t frames for foreground observations that are darker and brighter than background observations.

Fig.7 (Color online) The background image restoration using the proposed method for a human stops for awhile in the scene such that: (a) Input video frames at times 1, 90, 120, and 150, respectively and (b) A background image restored after training the first 150 frames.

Fig.8 (Color online) The background image restoration using the proposed method for a moving object that appears from the first frame and remains till the last frame such that: (a) (a) A sample of a video sequence at times 1, 15, 30, and 40, respectively and (b) A background image restored after training the first 40 frames.

Fig.9 (Color online) The background image restoration using the proposed method for non-stationary background such that:(a) (a) Input video frames at times 1, 25, 50, and 70, respectively and (b) A background image restored after training the first 70 frames.

Fig.10 Illustration of silhouette extraction (a) Original image, (b) Separated foreground, (c) Removing noise by opening followed by closing operation.

Fig.11 (Color online) Examples of background modeling difficulties such that: (a) Ghost appearance as a result of a moving object misclassification as a background such that it remains more than 50% of the training time, (b) Sudden switching light off experiment, and (c) Ghost appearance as a result of physical background motion.

Fig.12 (Color online) Example of updating the background image in non-stationary background scenario: (a) $Bg_1^c(x, y)$ after training period and (b) $Bg_2^c(x, y)$ after the updating by OUM.

Fig.13 (Color online) The result of applying the proposed method for (a) sudden illumination changes experiment, (b) Ghost appearance experiment, and (c) Non-stationary background.

Fig.14 (Color online) A comparison among the results of the proposed method and two commonly used baseline methods MOG and CB methods using: (a) Traffic video, (b) Fountain video (i2r dataset), (c) Sea surface video (i2r dataset), (d) Gesture video (keck gesture dataset), and (e) Intelligent room video (shadow detection dataset).

Fig.15 A comparison between four background subtraction methods using (a) TPR and (b) FPR for the Wallflower dataset.

Fig.16 (Color online) Silhouette detection of the proposed method the proposed method for the wallflower dataset.

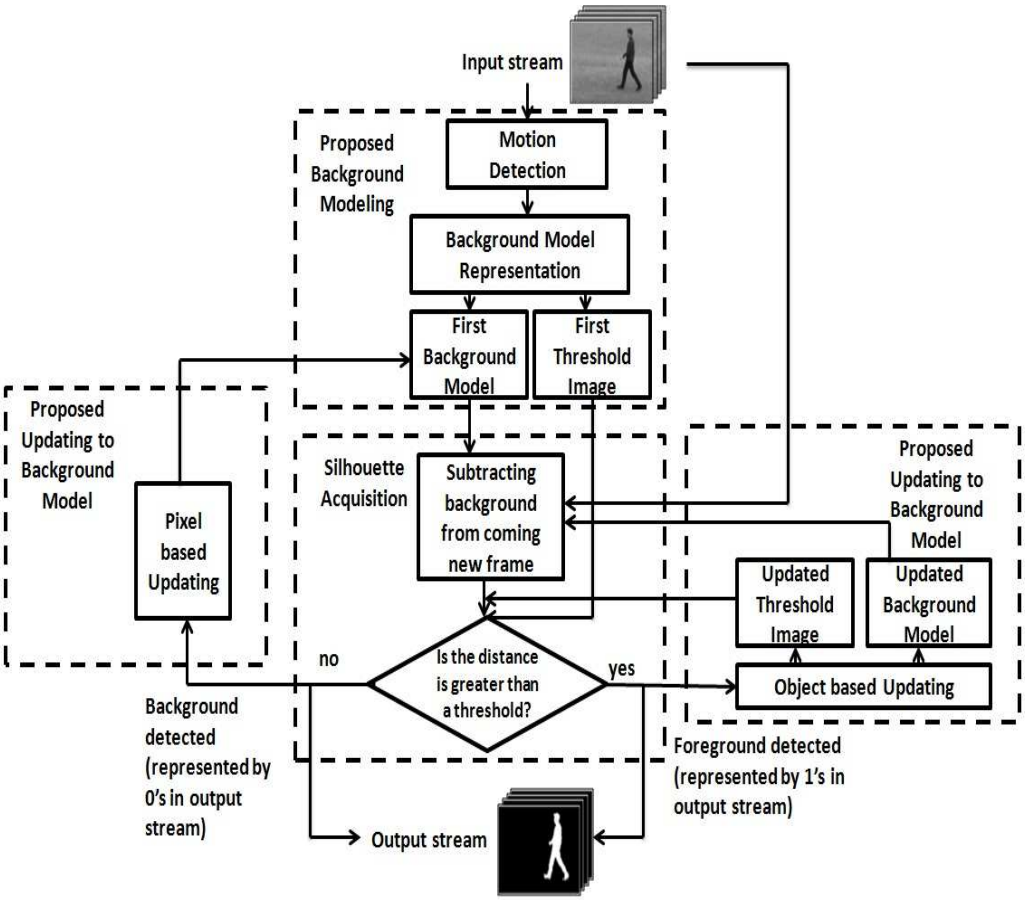

Input stream
Proposed Background Modeling
Motion Detection
Background Model Representation
First Background Model
First Threshold Image
Proposed Updating to Background Model
Pixel based Updating
Silhouette Acquisition
Subtracting background from coming new frame
Proposed Updating to Background Model
Updated Threshold Image
Updated Background Model
Object based Updating
no
Is the distance is greater than a threshold?
yes
Background detected (represented by 0's in output stream)
Foreground detected (represented by 1's in output stream)
Output stream

Fig. 1

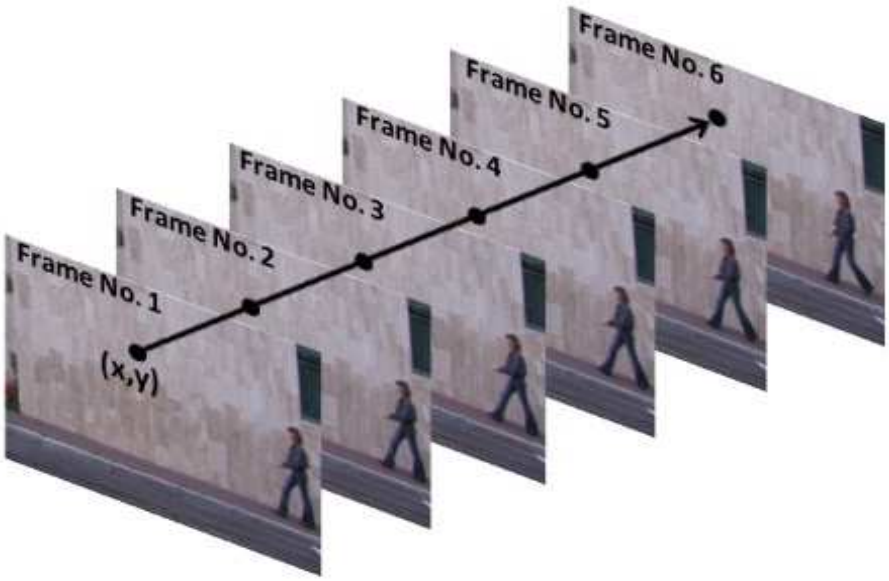
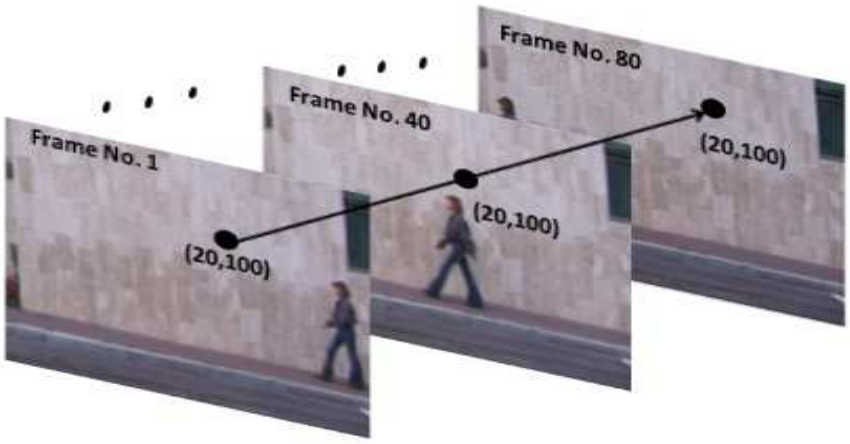
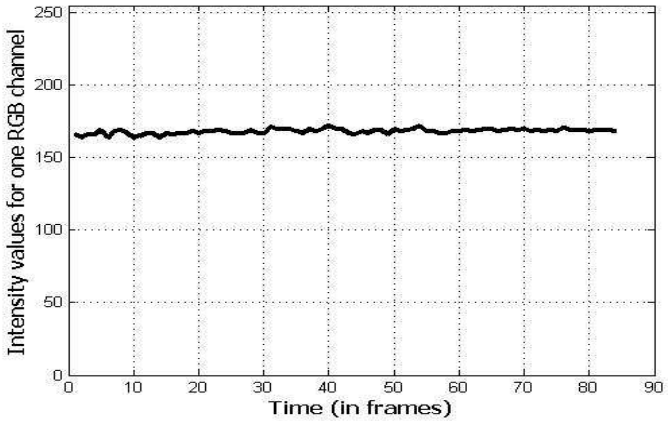

Frame No. 1
Frame No. 2
Frame No. 3
Frame No. 4
Frame No. 5
Frame No. 6
(x,y)

Fig. 2

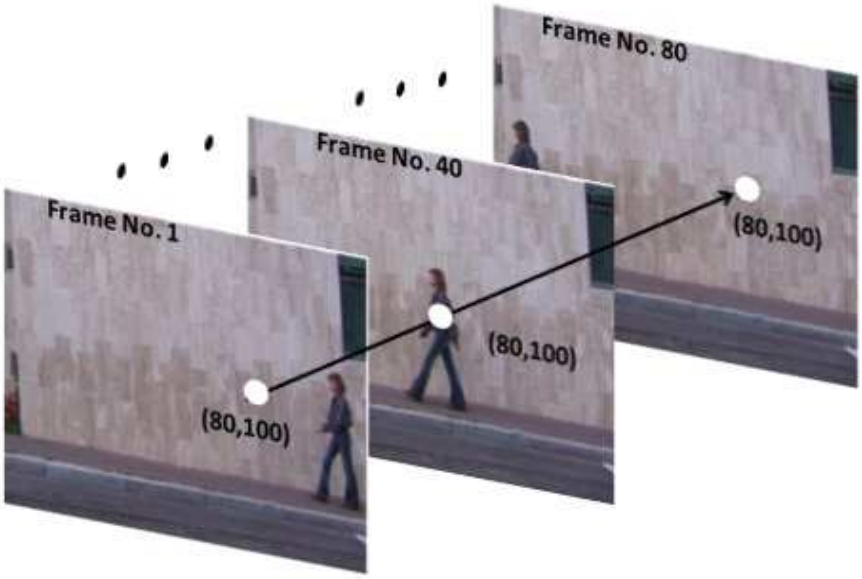

Frame No. 1
Frame No. 40
Frame No. 80
(20,100)
(20,100)
(20,100)

(a)

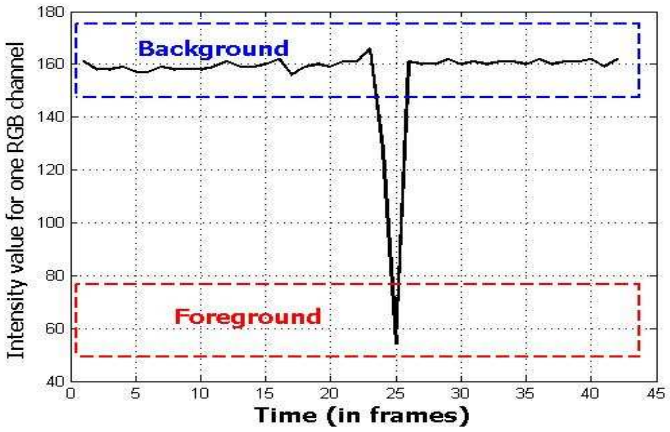

Intensity values for one RGB channel
Time (in frames)
0
50
100
150
200
250
0
10
20
30
40
50
60
70
80
90

(b)

Fig. 3

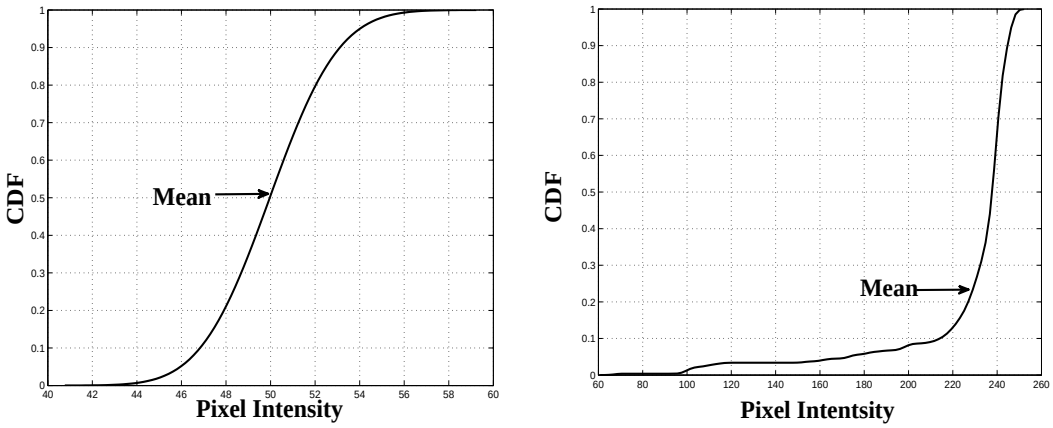

Frame No. 1
Frame No. 40
Frame No. 80
(80,100)
(80,100)
(80,100)

(a)


Background
Foreground
Intensity value for one RGB channel
Time (in frames)
180
160
140
120
100
80
60
40
0
5
10
15
20
25
30
35
40
45

(b)

Fig. 4



(a) (b)

Fig. 5

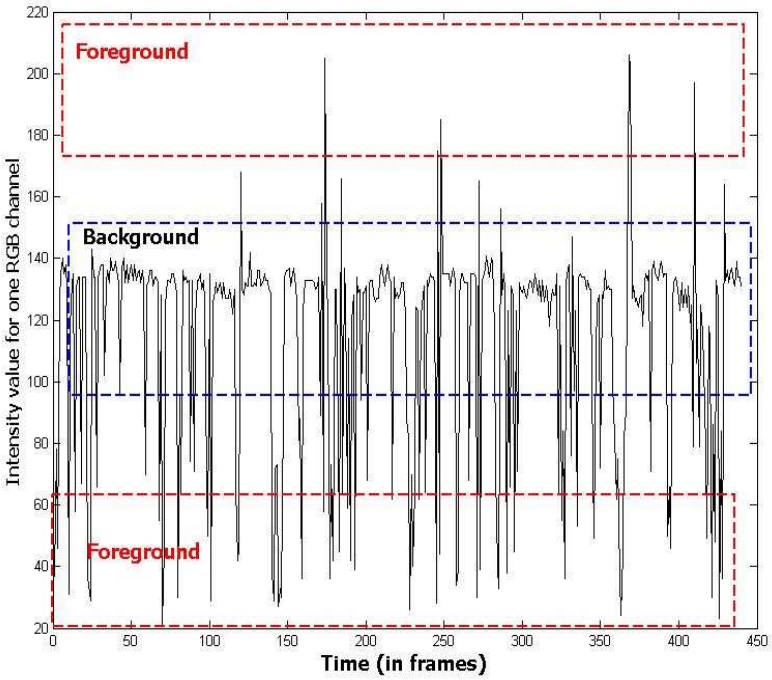

Foreground
Background
Foreground
Intensity value for one RGB channel
Time (in frames)
220
200
180
160
140
120
100
80
60
40
20
0
50
100
150
200
250
300
350
400
450

Fig. 6

(a)

(b)

Fig. 7

(a)

(b)

Fig. 8

(a)

((b)

Fig. 9

(a) (b) (c)

Fig. 10

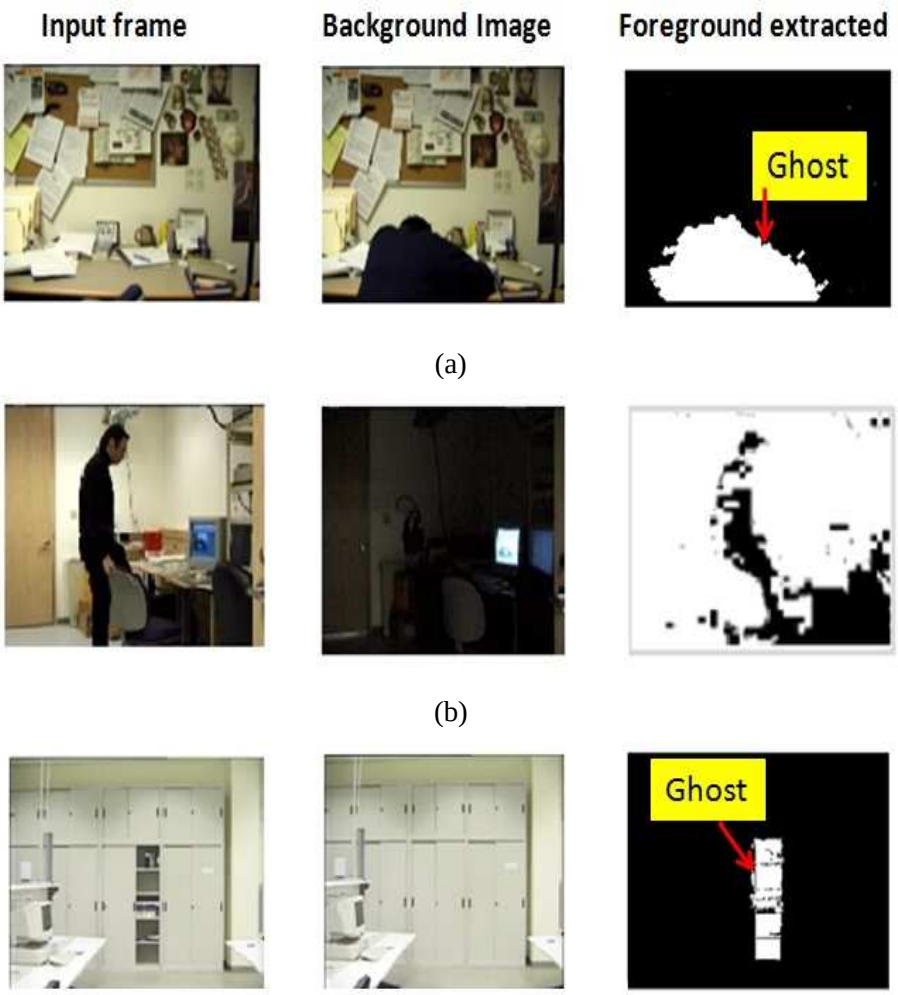
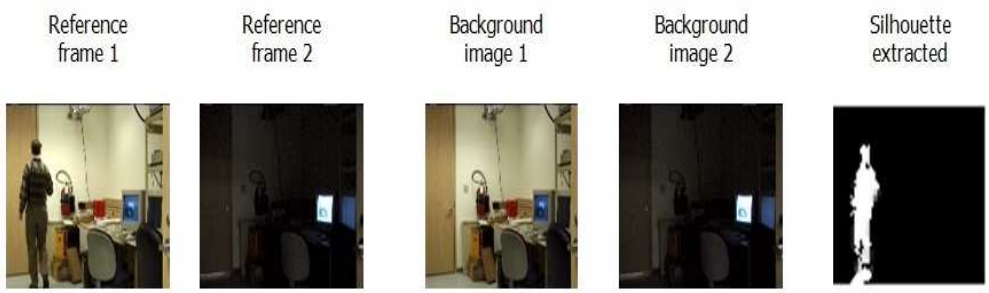

Input frame
Background Image
Foreground extracted
Ghost
(a)
(b)
Ghost
(c)

Fig. 11

(a) (b)

Fig. 12


Reference frame 1
Reference frame 2
Background image 1
Background image 2
Silhouette extracted

(a)

(b)

(c)

Fig. 13

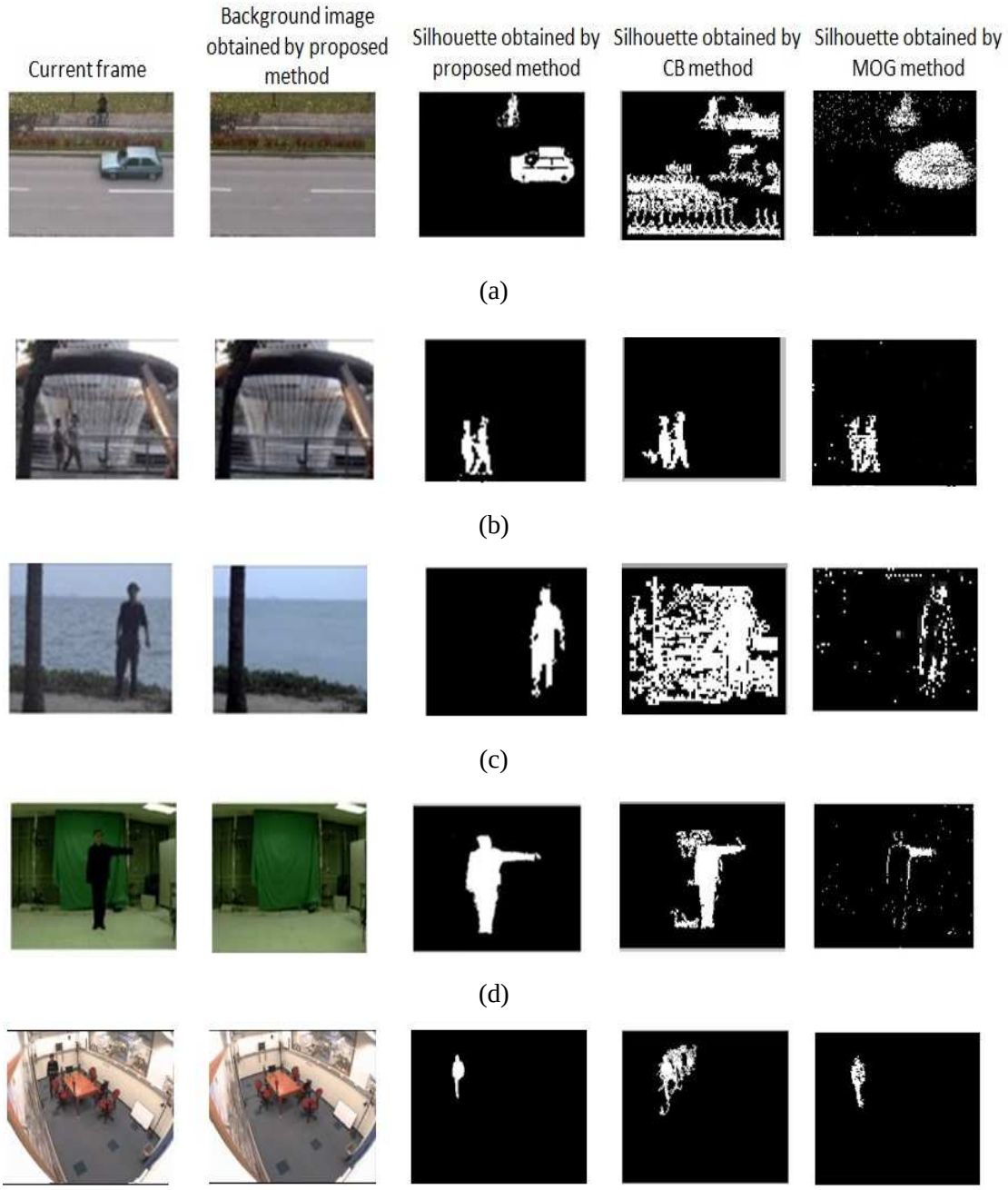
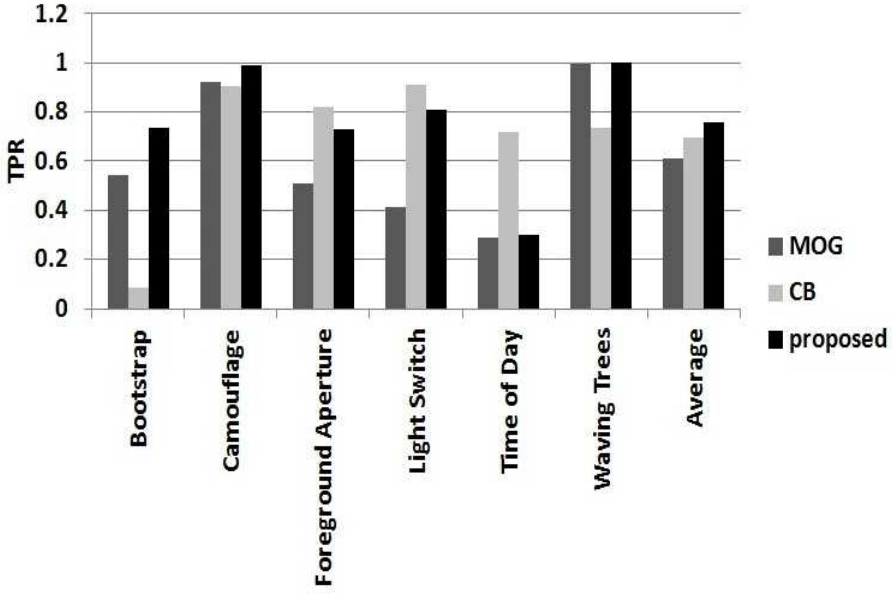
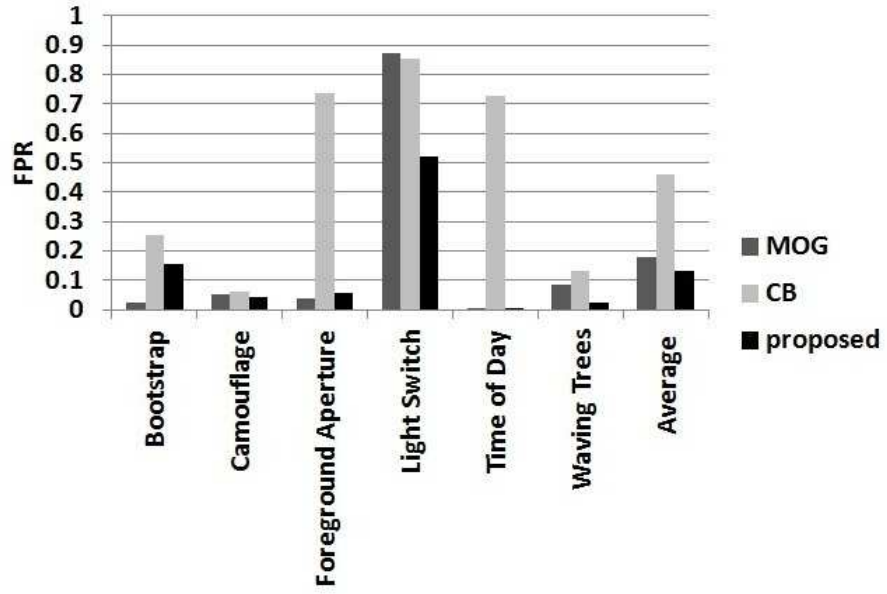

Current frame
Background image obtained by proposed method
Silhouette obtained by proposed method
Silhouette obtained by CB method
Silhouette obtained by MOG method
(a)
(b)
(c)
(d)
(e)

Fig. 14


1.2
1
0.8
0.6
0.4
0.2
0
TPR
Bootstrap
Camouflage
Foreground Aperture
Light Switch
Time of Day
Waving Trees
Average
MOG
CB
proposed

(a)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
FPR
Bootstrap
Camouflage
Foreground Aperture
Light Switch
Time of Day
Waving Trees
Average
MOG
CB
proposed

(b)

Fig. 15

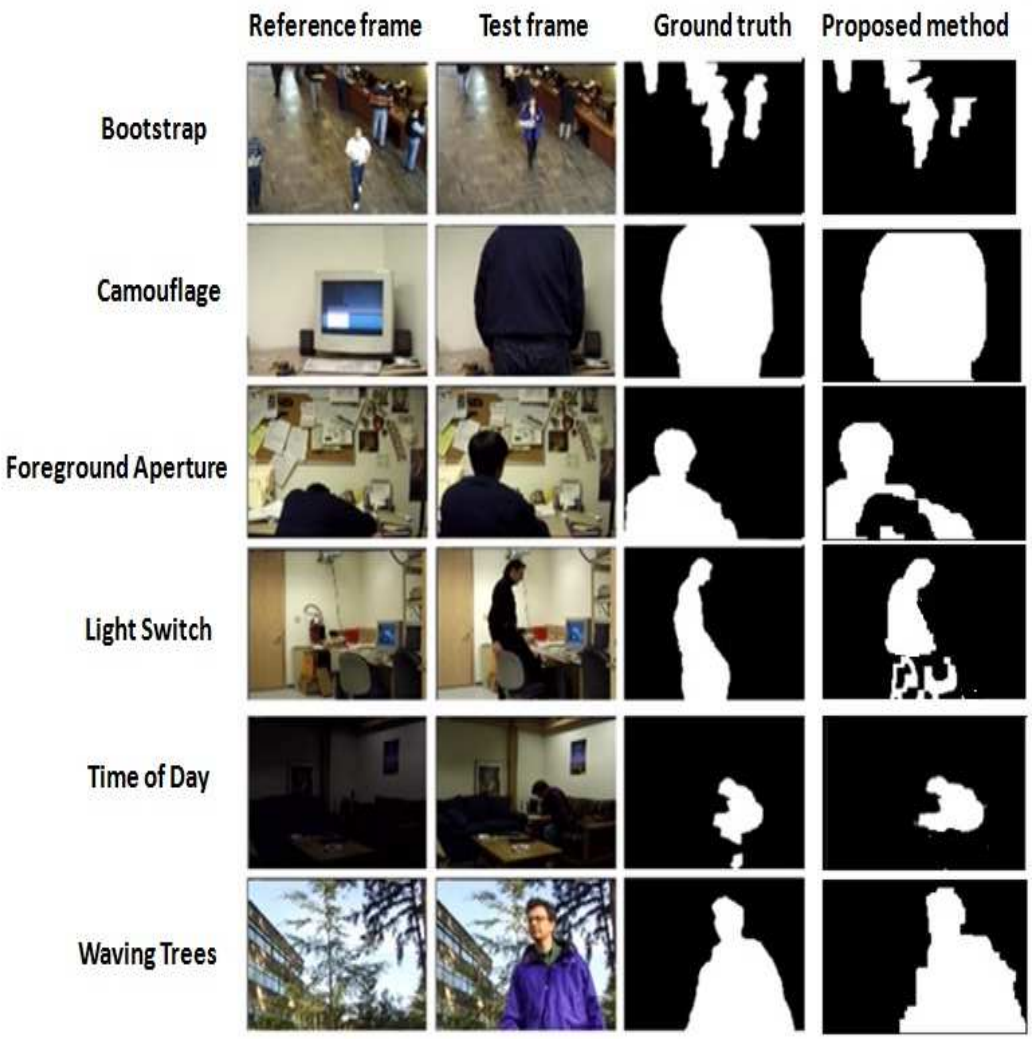

Reference frame
Test frame
Ground truth
Proposed method
Bootstrap
Camouflage
Foreground Aperture
Light Switch
Time of Day
Waving Trees

Fig. 16