Background Updating and Shadow Detection Based on Spatial, Color, and Texture Information of Detected Objects

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*

Background model updating is a vital process for any background subtraction technique. This paper presents an updating mechanism that can be applied efficiently to any background subtraction technique. This updating mechanism exploits the color and spatial features to characterize each detected object. Spatial and color features are used to classify each detected object as a moving background object, a ghost, or a real moving object. The starting position of each detected object is the cue for updating background images. In addition, this paper presents a hybrid scheme to detect and remove cast shadows based on texture and color features. The robustness of the proposed method and its effectiveness in overcoming challenging problems such as gradual and sudden illumination changes, ghost appearance, non-stationary background objects, the stability of moving objects most of the time, and cast shadows are verified quantitatively and qualitatively.

Keywords: background subtraction, statistical modeling, texture analysis

1. Introduction

Moving object detection is a low-level and basic task in many applications such as traffic monitoring, surveillance, patient monitoring, and so forth. The goal of this task is to separate pixels that represent moving objects from those corresponding to stationary background objects. Various approaches to modeling this problem can be found in the literature such as techniques based on optical flow,[1)] whose main drawbacks are its sensitivity to noise and the high computational time compared

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

with other approaches;[2] frame differencing, which is efficient but may not be able to segment the interior pixels of large uniformly colored moving objects;[3] or background subtraction, which models the background and then subtracts sequence frames from the background model. The background subtraction method is the easiest and most effective method among the above methods for detecting and segmenting moving objects. The critical step of background subtraction methods is how to build and maintain the background model.

Background subtraction methods should solve two problems. The first problem is how to model the real background of a video sequence as accurately as possible, so that the shape of the moving object can be detected accurately. The second problem is how to update the background model in order to have sufficient sensitivity to changes in the background scene such as moving background objects, sudden illumination changes, and physical changes in the background. In the case that a sudden change occurs in a scene, background pixels in the location of the change may be misclassified as foreground pixels, which causes the appearance of *ghosts*. A ghost is defined as a set of foreground pixels that does not correspond to a real moving object.

To model the background, many methods have been developed. These methods can be classified into two classes: parametric and non-parametric. For parametric methods, a set of parameters is modeled to represent the background (reference image). Stauffer and Grimson [4,5] modeled the probability density function of each pixel as a mixture of Gaussians (MOG) and used an online approximation to update the model. Toyama *et al.*[6] applied a linear predictive filter for each pixel history to estimate the current background. The filter coefficients are computed from the sample covariance values of each pixel history. Haritaoglu *et al.*[7] proposed a real-time system (called W4) for detecting and tracking multiple people in addition to monitoring their activities in an outdoor environment. Despite the robustness of W4 method in detecting foreground objects, W4 method could not address completely difficulties such as cast shadows, moving background objects, and sudden changes. Jacques Jr *et al.* [8] improved W4 by applying the normalized cross-correlation to foreground pixels to obtain candidate shadow pixels. A refinement process was then applied to further improvement of the shadow segmentation. Monnet *et al.* employed the Kalman filter to model a background with dynamic textures.[9)]

Non-parametric modeling methods model each pixel as a random variable in a feature space with an associated probability density function. Elgammal *et al.*[10)] built a non-parametric background model by kernel density estimation. For each pixel, observed intensity values are retained for estimating the underlying probability density function, and then the probability of the new intensity values can be calculated using the kernel function. One major issue that needs to be addressed when using a kernel density estimation technique is the choice of a suitable kernel bandwidth (scale). The constructed background model is robust and can handle situations when the background of the scene is cluttered and not completely static. Tavakkoli *et al.*[11)] proposed adaptive kernel density estimation (AKDE) as a baseline system that addresses the issue of scene dependence. Then, a statistical technique called recursive modeling (RM) is used to overcome the weaknesses of AKDE in modeling slow changes in the background.

However, most background modeling techniques do not explicitly handle changes suddenly occurring to a background (for example, parked cars) after the background modeling process. In addition, these background modeling techniques cannot handle the problem of moving objects that remain stationary for most of the modeling time (for example, a sleeping person). Therefore, online updating of the background model is a vital process. Background updating or maintenance aims to adapt the background model to each change in the scene that is not modeled explicitly during the training period. Changes that occur in the background can be classified as follows:

- A moving object appears in the scene and then stops to become short-term background.
- A stationary moving object is modeled as background and then leaves the scene, causing the appearance of a ghost.
- The movement or displacement of one or more of the objects classified as background such as chairs or tree leaves.

The background updating or maintenance methods can be grouped into two categories. In the first category, which is referred to as pixel-based updating methods, the pixel-updating strategies can be summarized as follows:

(1) All the pixels are updated according to the new pixel values. For example, MOG method replaces the least probable distribution with the distribution of the new pixel values. However, this updating mechanism has difficulty in dealing with a background that has rapid variations or repetitive motion such as tree leaves.[10)]

(2) Updating pixels that are classified as background pixels after the background subtraction process.[10)] The problem with this mechanism is that pixels erroneously classified as background are used to update an incorrect pixel. In addition, this updating mechanism cannot overcome the problem of ghost appearance as all the pixels of the ghost are always classified as foreground.

The second category, which is referred to as pixel-and-objects-based updating methods, includes the updating of background pixels and background objects that are erroneously classified as foreground. Haritaoglu *et al.* periodically updated the pixels classified as background that remain unchanged for an amount of time. In addition, an object-based updating method is used to update the background model to adapt to physical changes in the background scene. If a pixel is continuously classified as foreground for an amount of time, this pixel value is used to update the background model. Despite the robustness of this method, it destroys object integration, for example, if a pixel that is classified as foreground is then classified as background erroneously, this pixel will not update the background model. In addition, W4 method does not provide a solution to the problem of a non-stationary background such as tree leaves. Hamad and Tsumura[12, 13)] proposed an effective updating mechanism that combines the updating of pixels and objects. This mechanism exploits color and shape features of each detected object to cope with difficulties such as sudden illumination changes, ghost appearance, and non-stationary background objects. The high consumption of CPU time and the poor detection of shadows are considered the main drawbacks of their mechanism.

In this paper, a fast and efficient mechanism for updating or maintaining a background model is presented. This mechanism relies on extracting spatial and color features of the detected objects and specifies their starting position of motion. The starting position of motion is used as a cue to classify each detected object as a moving background object, a ghost, or a real moving object. In addition, this paper presents a hybrid scheme for detecting and removing shadow pixels. This scheme relies on the texture and the color features of pixels. The proposed method is applied to a reference image learned and extracted by W4 method to demonstrate the ability of the proposed method to efficiently update a reference image constructed by any background subtraction technique. Figure 1 shows an overview of the proposed method.

This paper is organized as follows: a brief description of W4 method is given in §2. The learning and pixel-based updating of the reference image are described in §3. The extraction of foreground pixels is discussed in §4. The updating of the reference image and shadow suppression using spatial, color, and texture features are outlined in §5. A quantitative and qualitative analysis is presented in §6. A conclusion is provided in §7.

2. Related Work

W4 method was first proposed by Haritaoglu *et al.* as a real-time system for detecting and tracking multiple people and monitoring their activities in indoor and outdoor environments.[7] Our concerns in this work are the background modeling and foreground detection modules. W4 detects moving objects after learning the initial background model during the training period (20–40 s) and then updating the model parameters. During the training phase, W4 method attempts to separate the pixel history (the intensities of a pixel in all frames of the training period) into stationary pixels and moving pixels. The median and standard deviation of each pixel history are used to construct the initial reference model. The initial reference model is exploited to separate the foreground moving pixels and stationary pixels for each pixel history. Stationary pixels for each pixel history located at x are modeled by three statistical values: the minimum *m(x)*, maximum *n(x)*, and maximum interframe intensity difference between two consecutive frames *d(x)*.

To adapt the initial background model to changes, W4 method uses two different approaches that are based on pixel and object updating. The pixel-based update adapts to changes in the illumination. The object-based update adapts to physical changes in the background, for example, a parked car that remains stationary for a long time and then moves. W4 method constructs a change map to apply the pixel-and-object-based updates. The change map consists of three main components:

- *A detection support map (gS)*, which stores each pixel that was detected as background in last N frames. Each time the pixel is classified as background, the pixel value in *gS* is incremented.

- *A motion support map (mS)*, which stores the number of times the pixel is classified as foreground. The classification occurs by subtracting three consecutive frames. The pixel is classified as foreground during the training period if $| I_i(x) - I_{i-1}(x) | > 2\sigma(x)$, where $I_i(x)$ is the intensity of pixel x at frame i and $\sigma(x)$ is the standard deviation of the intensities at pixel location x in all frames in the training period. If the pixel is classified as foreground, the corresponding pixel value in *mS* is incremented.

- *A change history map (hS)*, which represents the elapsed time in frames since the last time the pixel was classified as a moving pixel.

In the tracking module, W4 updates the parameters of the background model at pixel x based on the change map separately for all pixels that are classified as foreground $(m^f(x), n^f(x), d^f(x))$ and for all pixels that are classified as background $(m^b(x), n^b(x), d^b(x))$. The values $(m^u(x), n^u(x), d^u(x))$ are the current background model parameters being used. The background model is updated as follows:

$$[m(x), n(x), d(x)] = \begin{cases} [m^b(x), n^b(x), d^b(x)] & if(gS(x) > k * N) \\ & (pixel - based) \\ [m^f(x), n^f(x), d^f(x)] & if(gS(x) < k * N \wedge mS(x) < r * N) \\ & (object - based) \\ [m^u(x), n^u(x), d^u(x)] & otherwise \end{cases}, \quad (1)$$

where k and r are typically 0.8 and 0.1, respectively. k and r are very important parameters. From eq. (1), $k*N$ is the elapsed time in frames in which the pixel remains classified as background and $r*N$ is the elapsed time in frames in which the pixel remains classified as foreground. Therefore, the condition $mS(x) < r * N$ leads to the updating of all pixels in all frames. Thus, this condition should be replaced by $mS(x) > r * N$.

Foreground pixels are then segmented from the background in each frame by calculating the difference between the current pixel intensity and the corresponding pixel in the background model. The difference is then thresholded by the parameter d_λ. The parameter d_λ is the median of the largest interframe absolute difference $d(x)$ over the entire background model. The pixel location x is classified as background or foreground as follows:

$$f(x) = \begin{cases} 0 & if(|\ I_t(x) - m(x)\ | < T * d_\lambda) \vee (|\ I_t(x) - n(x)\ | < T * d_\lambda) \\ 1 & otherwise \end{cases}, \quad (2)$$

where T was chosen to be 2 after running a series of experiments.[7] After the segmentation of foreground regions, three stages are employed: noise removal, morphological operations, and object detection.

3. Learning and Updating Reference Images

Background modeling is the key to any background subtraction technique. The proposed method mainly relies on two background models. The first model is constructed and learned after a training period based on the idea of W4 method. The second model is an updated version of the first model after adapting the model to sudden changes in a scene. The proposed method exploits the spatial and color information of the detected objects to update the second model. Similarly to W4 method, the proposed method obtains the first reference image even if moving objects exist in the field of view during the training period.

Unlike W4 method, the proposed method can handle RGB color image sequences. Working with color information gives rich details that can be employed to update the reference image and perform shadow suppression, as will be illustrated in §5. The first reference image is denoted in the following context as $R_1^c(x)$, where c represents one of the R, G, and B color channels. The training period is estimated to be 90–150 frames (equivalent to 3–5 s) and is represented by N frames. The median $\lambda^c(x)$ and standard deviation $\sigma^c(x)$ of the intensities at pixel x in all frames in the training period are computed. The proposed method exploits $\lambda^c(x)$ and $\sigma^c(x)$ to separate stationary pixels and foreground pixels for each pixel history at location x. The stationary pixels for a pixel history located at x are given by

$$| I_i^c(x) - \lambda^c(x) | < 2 * \sigma^c(x), \tag{3}$$

where $I_i^c(x)$ is the intensity of pixel x over RGB channel c at frame i and $i \in \{1, ..., N\}$.

The array $S^c(x)$ represents the stationary pixels for RGB channel c for the pixel history located at x. The array $S^c(x)$ is modeled by four statistical values: the median $\lambda_s^c(x)$, standard deviation $\sigma_s^c(x)$, minimum $min_s^c(x)$, and maximum $max_s^c(x)$. The first reference model, $R_1^c(x)$ at pixel x over RGB channel c, is set to $\lambda_s^c(x)$. The modeling parameters $\sigma_s^c(x)$, $min_s^c(x)$, and $max_s^c(x)$ are used to determine a threshold for each pixel x in order to classify pixel x as foreground or background. Unlike W4 method, the proposed method does not use the maximum interframe intensity difference parameter, which consumes a considerable amount of memory and processing time. Figure 2 shows a clear background image restored from a video that contains different moving objects in the field of view during the training period.

$$R_1^c(x) = \lambda_s^c(x). \tag{4}$$

Any change occurring in the background leads to a change in the background statistics and consequently in the background model. The proposed method copes with gradual changes in the background by updating the modeling parameters of $S^c(x)$. If the new intensity of the pixel located at x (denoted as $I_{new}^c(x)$) is classified as background, the parameters $\lambda_s^c(x)$, $\sigma_s^c(x)$, $min_s^c(x)$, and $max_s^c(x)$ are updated as follows:

$$\lambda_{i+1}^c(x) = \alpha * I_{new}^c(x) + (1-\alpha) * \lambda_i^c(x), \tag{5}$$

$$(\sigma_{i+1}^c(x))^2 = \alpha * (I_{new}^c(x) - \lambda_i^c(x))^2 + (1-\alpha) * (\sigma_i^c(x))^2, \tag{6}$$

$$min_{i+1}^c(x) = min(I_{new}^c(x), min_i^c(x)), \tag{7}$$

$$max_{i+1}^c(x) = max(I_{new}^c(x), max_i^c(x)), \tag{8}$$

where $\lambda_i^c(x)$, $\sigma_i^c(x)$, $min_i^c(x)$, and $max_i^c(x)$ are the ith update of the parameters $\lambda_s^c(x)$, $\sigma_s^c(x)$, $min_s^c(x)$, and $max_s^c(x)$, respectively. For $i = 1$, $\lambda_1^c(x)$, $\sigma_1^c(x)$, $min_1^c(x)$, and $max_1^c(x)$ are set to $\lambda_s^c(x)$, $\sigma_s^c(x)$, $min_s^c(x)$, and $max_s^c(x)$, respectively. The parameter α is empirically set to 0.1 in our experiments. We ran a series of experiments to determine the optimal value of α. The results show that setting α to be greater than 0.1 leads to the appearance of distortion in the reference image.

4. Foreground Extraction

On the basis of the constructed reference image, the moving object is detected and its silhouette is extracted from the video sequence. Jacques Jr *et al.* showed that the foreground extraction module of W4 method may lead to a misclassification of background pixels as foreground.[8)] Although the new pixel intensity may lie between the minimum and maximum values, this pixel may be classified as foreground. In addition, W4 method uses a global threshold d_λ to classify pixels into background and foreground. Owing to the varying effects of noise and illumination on each pixel, a single global threshold is not an efficient way to classify pixels.[13)] Therefore, the proposed method uses a pixel-based threshold that depends on pixel history parameters modeled previously. The threshold $Th_1^c(x)$ at pixel x relies on the minimum, maximum, and standard deviation of $S^c(x)$ and is given by

$$Th_1^c(x) = (max_s^c(x) - min_s^c(x)) + \frac{1}{\sigma_s^c(x)} * (max_s^c(x) - min_s^c(x)), \tag{9}$$

where $\frac{1}{\sigma_s^c(x)} * (max_s^c(x) - min_s^c(x))$ is added to the range of $S^c(x)$ to allow the detection step to be adapted to camera noise and limited changes in the illumination (such as turning on a desk lamp or a car light). The weight $\frac{1}{\sigma_s^c(x)}$ is provided to handle the problem of small ranges of $S^c(x)$, which is the case in modern cameras.

A common strategy that follows the construction of the reference model is to subtract the new frame from the reference image. The proposed method performs the subtraction process pixel by pixel as follows:

$$dist_1^c(x) = |I_{new}^c(x) - R_1^c(x)|, \tag{10}$$

where $I_{new}^c(x)$ is the intensity of pixel location x over RGB channel c in the new frame. The value $dist_1^c(x)$ is the distance between the new frame and the reference image at pixel location x over RGB channel c.

In order to classify pixel x as background (represented by 0) or foreground (represented by 1), the proposed method thresholds the average value of $dist_1^c(x)$ over the R, G, and B channels using the average value of $Th_1^c(x)$ over the channels. This classification is given by

$$silh(x) = \begin{cases} 0 \quad if \quad Average^c(dist_1^c(x)) < Average^c(Th_1^c(x)) \\ 1 \quad if \quad Average^c(dist_1^c(x)) \geq Average^c(Th_1^c(x)) \end{cases}, \tag{11}$$

where $Average^c(dist_1^c(x))$ is the average value of $dist_1^c(x)$ over the RGB color channels, $Average^c(Th_1^c(x))$ is the average value of $Th_1^c(x)$ over the RGB color channels, and $c \in \{R, G, B\}$.

After the classification of each pixel, each video frame is represented by two values (i.e., 0 and 1). The proposed method applies a set of morphological operations to extract silhouettes. The proposed method performs an opening operation followed by a closing operation to remove noise. The closing operation is performed again to preserve the silhouette edges. Then, the silhouette edges are extracted by computing the difference between the frame and the eroded version of the frame. A dilation operation is followed to bridge the disconnected areas of the extracted silhouette contour. Finally, morphological operations are used to fill the silhouette gaps. Figure 3 illustrates the application of the morphological operations on the extracted foreground.

5. Proposed Reference Image Updating Using Spatial, Texture, and Color Information of Detected Objects

Hamad and Tsumura proposed a mechanism called OUM that relies on the shape and color features of each detected object to robustly update background and threshold images.[12,13] OUM uses the Euclidean distance to measure the similarity between the features of each detected object in the current frame and the features of all objects in the previous frame. The similarity measure is then employed to determine whether the object starts its motion inside or outside the scene. Despite the robustness of this mechanism, the considerable amount of time required is considered as a drawback of OUM, especially in crowded videos. The time consumption problem is due to the computations of the Euclidean distance between the features of detected objects and the features of all objects in previous frames. Moreover, OUM does not handle the problem of local illumination changes (e.g., shadows).

The proposed method modifies OUM to reduce the time consumption. In addition, the proposed method attempts to detect and remove shadows. For brevity, OUM after modification will be hereafter referred to as MOUM (M stands for modified). MOUM relies on the idea that the starting position for regular moving objects is outside the scene. If the detected object starts its motion inside the scene, this is considered an irregular situation. Consequently, this object is analyzed and classified as a real moving object, a ghost, or a moving background object. After the foreground extraction step, the proposed method labels each detected object (a connected set of foreground pixels) as ℓ (where $\ell \in \{1,2,3,...,W\}$ and W is the number of objects in the whole video). For each labeled object, spatial and color features are extracted and stored in a database DB. In addition, MOUM stores the number of the frame in which the detected object appears for the last time and denotes this parameter as n.

The object skeleton centroid, area, and bounding box are the spatial features extracted by MOUM to characterize each detected object shape in the binary frames (obtained after the foreground extraction step). The centroid of an object is influenced by the large amount of motion of its extremities.[7] Therefore, MOUM utilizes the object skeleton centroid, which is not influenced by the motion of extremities. The parameters of the bounding box surrounding the detected object in the binary frame are used to locate the detected object in the corresponding color frame.

Color moments are the features extracted from the detected object in the color frame. The first order (mean), second order (variance), and third order (skewness) color moments have been shown to efficiently and effectively represent the color distributions of images.[14] The three moments for each RGB channel give a feature vector that includes nine elements. The computation of the color moments was reported by Choras.[14] For simplification, the detected object in the binary frame is denoted as BO_i^j, the corresponding object in the color frame is denoted as CO_i^j, and the centroid of the BO_i^j skeleton is denoted as y_i^j (where i represents the frame number and j is the object number in frame i). Figure 4 shows an example of BO_i^j and CO_i^j.

For each new frame, each detected object (BO_i^j in the binary frame and CO_i^j in the corresponding color frame) is characterized by spatial and color features and is given a label ℓ. A feature vector that includes the color moments and the area of each labeled object is constructed and stored in *DB* (this feature vector is denoted as f_ℓ). The Euclidean distance between the feature vector of each new detected object (which is denoted as f_{new}) and f_ℓ is estimated. If the distance is less than a threshold T_1, the detected object is recognized as being found in *DB*. Otherwise, this object is considered a new object, given a label ℓ, and stored in *DB*.

5.1 Detecting ghosts and illumination changes

MOUM aims to cope with problems such as sudden illumination changes, ghost appearance, and non-stationary background objects. Figure 5 shows examples of such problems. Owing to the repeated occurrence of moving background objects (e.g., tree branches) or the repeated switching of a light on and off, MOUM applies the updates to a copy of the first reference model and denotes the second reference model as $R_2^c(x)$. The key factor of MOUM is to determine whether the detected object initiates its motion inside the scene or it exists in the previous frames. The Euclidean distance *dist* between f_{new} and the stored feature vector f_ℓ is given by

$$dist(f_{new}, f_\ell) = \sum_{\ell=1}^{W_{DB}} \sqrt{(f_{new} - f_\ell)^2}, \tag{12}$$

where W_{DB} is the number of the objects whose feature vectors are stored in *DB*.

If $dist(f_{new}, f_\ell)$ is less than or equal to T_1, the detected object is concluded to have appeared previously in *DB* and is assigned a label ℓ. The following step is to determine whether the object BO_i^j appeared in the previous frame *i*-1 by computing the difference between the current frame number and *n*. If the difference is less than or equal to 1, BO_i^j appeared in the previous frame *i*-1. If the difference is greater than 1, BO_i^j appeared before, then moved out of the scene, then reappeared. Therefore, this object starts its motion from inside the scene, which is considered an irregular situation.

If $dist(f_{new}, f_\ell)$ is more than T_1, we conclude that the object BO_i^j does not exist in the previous frames and that it initiates its motion in frame *i*. Consequently, the motion status of BO_i^j and CO_i^j is analyzed by considering the following scenarios:

(a) Non-stationary background objects: MOUM specifies the location of CO_i^j in the first reference image $R_1^c(x)$. Eight neighbour blocks to CO_i^j, centred at the location of CO_i^j, are generated in $R_1^c(x)$, where each block has the same size as CO_i^j. The color moment features are used to characterize CO_i^j and the eight neighbour blocks generated in the reference image. The Euclidean distance is computed between the CO_i^j moment features and the moment features of the eight neighbour blocks. If the distance is less than a threshold T_2 for one or more of the neighbour blocks, then CO_i^j is considered part of a non-stationary background (e.g., swaying tree branches). In this case, the block that has the closest match to CO_i^j in $R_1^c(x)$ replaces CO_i^j in $R_2^c(x)$. The statistical parameters: minimum, maximum, and standard deviation of the pixels at the block that has the closest match to CO_i^j replace those of the pixels at CO_i^j in $Th_2^c(x)$. Figure 6 shows an example of background image updating 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, we conclude that CO_i^j is a real moving object.

***(b) Ghost appearance*:** The detected object obtained after the foreground extraction step consists of a set of connected pixels. In the case that the detected set of pixels does not correspond to a real moving object, it is defined as a ghost. A ghost may appear for the following reasons:

(1) The stability of a real moving object: a real moving object may appear in a scene then stop to become short-term background as shown in Fig. 7(b). Alternatively, a real moving object may be modeled as a background object that remains stable for most of the training period and then leaves the scene as illustrated in Fig. 5(a).

(2) Sudden physical changes in the background as shown in Figs. 5(c) and 8(b).

MOUM detects a ghost in the case that the location of the detected object skeleton centroid y_i^j does not change for t s (in experiments, t is set to 0.5 s or equivalently 15 frames). Figures 7 and 8 show examples of ghost appearance as a result of the stability of a real moving object and a sudden physical change in the background, respectively. Figures 7(a) and 8(a) show the reference image $R_1^c(x)$ after the initial training period. The ghost is detected owing to the stability of the detected object skeleton centroid from frame fr_1 to frame fr_t in Figs. 7(b) and 8(b). Frame fr_0 is the frame that precedes the detection of the ghost. MOUM distinguishes the two cases shown in Figs. 7(b) and 8(b) by extracting the area and color moment features for the ghost and objects detected in frame fr_0. The Euclidean distance is used to measure the distance between the feature vector of the ghost and the feature vector of every object in frame fr_0. If the distance is less than a threshold T_3, we can conclude that the ghost is a moving object that stops to become short-term background as illustrated in Fig. 7(b). Otherwise, the ghost is detected owing to a sudden physical change that occurs in the background as shown in Fig. 8(b). MOUM adapts to the ghost appearance by recalling the learning reference image module discussed earlier in §3. The training period of the learning reference image module starts from frame fr_1 shown in Figs. 7(b) and 8(b). The aim of the learning reference image module is to obtain an updated background model $R_2^c(x)$. As part of the learning reference image module, the stationary pixels of each pixel history (which are denoted $S^c(x)$) are determined. Four statistical values, $\lambda_s^c(x)$, $\sigma_s^c(x)$, $min_s^c(x)$, and $max_s^c(x)$, are calculated to model each $S^c(x)$ and adapt the statistical parameters to the change occurring due to the ghost appearance. Figures 7(c) and 8(c) show the updated reference model $R_2^c(x)$ due to the ghost appearance.

***(c) Sudden illumination changes*:** MOUM instantaneously adapts to sudden changes in the background illumination (such as switching a light on or off). If the area of BO_i^j is greater than 80% of a scene, MOUM responds quickly and calls the reference image modeling technique to update $R_2^c(x)$ and the modeling parameters.

5.2 Cast shadow detection and removal

MOUM exploits texture and color information to detect and remove cast shadows. A cast shadow is an area projected by an object that prevents the light source from reaching this area. The parameters that determine the shadow size and orientation are the lighting conditions, camera position, reflective surfaces, and background texture.[15] A comprehensive survey with qualitative and quantitative comparisons of most of the shadow detection methods in the literature was presented by Prati *et al.*[16] Shadow detection approaches can be classified as property-based and model-based approaches.[17] Property-based approaches do not require any prior information about the parameters that affect the shadow size and orientation. Property-based approaches rely on spatial, spectral, and temporal features such as geometry, brightness, or color to identify shadowed regions. In contrast, model-based approaches require prior knowledge about the scene or the moving objects. Model-based approaches have shown less robustness than property-based approaches when used in different scenes and illumination conditions.[17]

Spectral features are the most common features used to detect shadows. Cucchiara *et al.* hypothesized that shadows reduce the background brightness and saturation while having a little effect on the hue properties in hue, saturation, and value (HSV) color space.[18] Schreer *et al.* used YUV color space instead of HSV color space to avoid the time consumption required by the HSV color transformation.[19] The segmentation of shadows from foreground objects relies on the observation that shadows reduce the YUV pixel value linearly. Horprasert *et al.* built a model in RGB color space to express the variation of normalized luminance and distortions in chromaticity.[20] However, techniques based on only chromaticity characteristics can lead to the misclassification of shadow pixels.[17]

MOUM employs both color and texture information to detect cast shadow regions. Texture is employed generally to discriminate homogeneous foreground and shadow regions. Shadow regions have two characteristics relating to the corresponding background regions: (1) Shadow-region pixel intensities are less than the corresponding background-region pixel intensities, (2) Shadow regions have similar texture to the corresponding background regions. On the basis of these characteristics, MOUM employs the intensity ratio between the extracted foreground pixels and the corresponding pixels in the reference image. In addition, MOUM divides the detected objects into blocks and exploits the entropy of such blocks as a texture feature to detect shadow regions.

The intensity ratio measures how closely the current pixel intensity matches that predicted by the background model.[21)] The intensity ratios between the pixels in CO_i^j and the corresponding pixels in reference image $R_2^c(x)$ are thresholded such that

$$\alpha_1^c < \frac{I_i^c(x)}{R_2^c(x)} < \alpha_2^c, \tag{13}$$

where $I_i^c(x)$ represents the intensity of pixel x over RGB channel c at frame i. The thresholds α_1^c and α_2^c are used to detect shadow pixels for each color channel c.

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image.[22)] MOUM divides CO_i^j into nonoverlapping blocks of 5x5 pixels. If the blocks do not perfectly fit CO_i^j, padding is added to the right and bottom borders of CO_i^j. Entropy is used as a texture feature to characterize each block as follows:

$$\epsilon_b^c = -\sum_{i=1}^{Z_b} p_b^c(I_i^c) * \log(p_b^c(I_i^c)), \tag{14}$$

where ϵ_b^c is the entropy of block b over color channel c, I_i^c is the intensity level i in block b at RGB channel c, $p(I_i^c)$ is the histogram of the intensity levels in block b at each RGB channel c, and Z_b is the number of possible intensity levels in block b.

MOUM classifies pixel x as a shadow pixel if the following two conditions are realized:

(1) The intensity ratio $\frac{I_i^c(x)}{R_2^c(x)}$ lies between the thresholds α_1^c and α_2^c.

(2) The absolute difference (which is denoted $\triangle\epsilon_{b,R}^c(x)$) between $\epsilon_b^c(x)$ of block b that contains pixel x and the entropy of the corresponding block in $R_2^c(x)$ that contains pixel x is less than a threshold β^c.

The classification of pixel x as a shadow pixel can be formulated as

$$Sh(x) = \begin{cases} shadow & if(\triangle\epsilon_{b,R}^c(x) < \beta^c) \wedge (\alpha_1^c < \frac{I_i^c(x)}{R_2^c(x)} < \alpha_2^c) \\ foreground & otherwise \end{cases}. \tag{15}$$

Pixels classified as shadow pixels are altered from 1 to 0 in each BO_i^j. After eliminating the shadow pixels, a set of morphological operations is applied to remove noise and smooth the moving object boundary. An opening operation followed by a closing operation is utilized to remove any noise resulting from the misclassification of shadow pixels. Finally, a closing operation is applied to the moving object to smooth its boundary. Figure 9 shows an example for detecting and removing shadows using MOUM. Figure 10 shows a comparison of the results obtained using OUM and MOUM methods. The comparison demonstrates the efficiency of MOUM method to detect and remove shadows.

5.3 Modification of foreground extraction process

The foreground extraction process described in §4 involves subtracting the new frame from the reference image $R_1^c(x)$ obtained after the training period. MOUM constructs updated versions of $R_1^c(x)$ and $Th_1^c(x)$. Therefore, the foreground extraction process should be modified to handle the new constructed images $R_2^c(x)$ and $Th_2^c(x)$.

MOUM subtracts the new frame from both $R_1^c(x)$ and $R_2^c(x)$ and the result is stored in $dist_1^c(x)$ and $dist_2^c(x)$, respectively. The average values of RGB channels for $dist_1^c(x)$ and $dist_2^c(x)$ are thresholded by the average values of RGB channels for $Th_1^c(x)$ and $Th_2^c(x)$, respectively. Pixel x is classified as foreground if the average values for $dist_1^c(x)$ and $dist_2^c(x)$ over the R, G, and B channels are greater than or equal to the average values for $Th_1^c(x)$ and $Th_2^c(x)$ at pixel x over the R, G, and B channels. This modification is given by

$$silh(x) = \begin{cases} 1 & if(Average^c(dist_1^c(x)) \geq Average^c(Th_1^c(x))) \wedge \\ & \quad (Average^c(dist_2^c(x)) \geq Average^c(Th_2^c(x))) \\ & \\ 0 & \qquad\qquad otherwise \end{cases} . \tag{16}$$

6. Experimental Results

Quantitative and qualitative evaluations of MOUM are presented in this section. The evaluation of reference image learning and construction is beyond the scope of this paper. Rather, we evaluate the effectiveness of the MOUM method for overcoming the problems of gradual and sudden illumination changes, ghost appearance, non-stationary background objects, and cast shadows. This work is considered as a modification of the work presented by Hamad and Tsumura[13)] with a different background modeling process. Therefore, MOUM can be employed with any background subtraction technique to overcome the above problems. The results of MOUM are assessed using various datasets with different situations and scenarios. The MOUM results are compared with two state-of-the-art background subtraction methods: MOG and W4.

6.1 Datasets and parameters selection

Although MOUM relies on several parameters, a fixed value for most of the parameters is found empirically to be suitable for a large number of studied videos (approximately 150 videos were studied; 30 videos were employed in the training of the parameters, and 120 videos were used for testing the parameters). Specifically, the number of frames utilized in the training phase ranges from N=60 to N=150, which was found to be sufficient to yield a clear reference image. The threshold T_1 is an important parameter in our proposed method. To select the value of T_1, the ground truth of each detected object in all frames of the training videos is generated. The true positive rate (TPR) is defined as the ratio between the number of correctly classified foreground pixels and the actual number of foreground pixels in the ground truth. TPR is computed for different sequences with different values of T_1, T_2, and T_3 as illustrated in Fig. 11. As shown in Fig. 11(a), $T_1 = 0.3$ (for normalized feature vectors) gives the highest TPR. Similarly, the TPR is used to select the thresholds T_2 and T_3, and it was found that $T_2 = 0.2$ and $T_3 = 0.3$ give the highest TPR as shown in Figs. 11(b) and 11(c), respectively. Figure 11 indicates that when the values of T_1, T_2, and T_3 are greater than 0.5, TPR drops to be less than 50%. In addition, TPR is employed to estimate the thresholds α_1^c, α_2^c, and β^c. We ran experiments to estimate α_1^c, α_2^c, and β^c on video sequences found on the shadow detection dataset.[23] α_1^c, α_2^c, and β^c are selected to be 0.2, 0.3, and 0.2, respectively.

We implemented the MOG method with three Gaussian components (K=3). The threshold used to identify the matching component is defined as the standard deviation of the Gaussian components scaled by 2.5. The threshold for identifying the components used to model the background is estimated to be 0.25 and the learning rate is set to 0.007. We implemented the W4 method with the parameters specified in §2.

The comparison among MOG, W4, and MOUM is performed using a variety of outdoor and indoor video sequences. These sequences have different situations, parameters (frame length, frame size, and so forth), and scene complexity. The sequences used in this comparison are samples from the following datasets: shadow detection,[23] KTH,[24] wallflower,[6] i2r developed by Liyuan *et al.*,[25] and vehicle stopping.[26] The wallflower and i2r datasets are provided with ground truth information (a manually segmented set of images that defines the foreground and background regions clearly). These two datasets are used to assess the three techniques of MOG, W4, and MOUM qualitatively and quantitatively.

6.2 Qualitative analysis

Figure 12 illustrates the effectiveness of MOUM in updating the reference image due to the appearance of ghosts. The video sequence used in this experiment is a vehicle stopping sequence provided by the VISOR repository.[26] The video consists of 2755 frames with frame size 320x256. Figure 12(a) shows the input frames at frame numbers 100, 677, 1100, and 2050. The scene is a car parking lot that remains stable until the cars within white and black rectangles reach the park starting from frame number 660. Then, the car within the black rectangle remains in the park until the end of the video, while the car within the white rectangle remains stable for some time then leaves the park at frame number 2050. As the two cars remain stable for more than 0.5 s (15 frames), this causes MOUM to update the reference image due to the appearance of two ghosts. After the leaving of the car within the white rectangle, the ghost reappears in the location of the leaving car, which makes MOUM update the reference image again. Figure 12(b) shows the reference image obtained after 100 frames, the first update of the reference image, and the second update of the reference image. The output frames shown in Fig. 12(c) result from the subtraction process between the input frames shown in Fig. 12(a) and the reference images shown in Fig. 12(b). The effectiveness of MOUM in overcoming the problem of moving object stability is thus shown in Fig. 12.

As OUM, MOG, and W4 suffer from the problem of shadow pixel misclassification, we compare MOUM with the scheme proposed by Xu *et al.* [27)] Xu *et al.* proposed a scheme for shadow detection using both color and texture cues.[27)] Their technique is based on the morphological reconstruction of shadow-removed regions based on the regions preceding the shadow-removal process. They assumed that object shapes are properly defined along most of their contours after the initial detection. Figure 13 shows a comparison between the technique of Xu *et al.* and MOUM. As shown in Figs. 13(a) and 13(b), MOUM gives better results than the technique of Xu *et al.* for both outdoor and indoor videos. Although the technique of Xu *et al.* gave reasonable results for outdoor videos, it failed to detect cast shadows for indoor videos.

Figures 14 and 15 present the comparison results of the three studied methods for two datasets: the i2r dataset and wallflower dataset. The first column in Figs. 14 and 15 displays sample frames of the datasets, the second column displays the ground truth frames, and the results of MOG, W4, and MOUM methods are depicted in the final three columns. As shown in Figs. 14 and 15, MOUM achieves better results than MOG and W4 for both indoor and outdoor scenes. The curtain, waving tree, fountain, and escalator sequences show the ability of MOUM to overcome the problem of moving background objects by updating the reference image. MOUM gives more accurate results than MOG and W4 in both light-switching sequences of the two datasets (i2r dataset and wallflower dataset) owing to the sensitivity of the MOUM method to the area of the detected object. The problem of the foreground aperture sequence is that the human remains stable for the entire period of the training phase. When this human moves afterwards, a ghost appears at the location of the human, causing the misclassification of background pixels as foreground pixels. MOUM detects the appearance of the ghost owing to the stability of the detected object skeleton centroid. Then, MOUM updates the reference image immediately. The foreground aperture sequence shown in Fig. 14 shows that W4 gives the worst result among the three methods. One of the drawbacks of MOG method is the difficulty it has in modeling a moving object that remains stable for a while as a background (as shown in the water surface sequence). MOUM overcomes this problem by detecting the reason for the ghost appearance. In the case that the reason for the ghost appearance is the stability of a real moving object, MOUM allows the application to decide to update the reference image by removing the stable moving object from the scene (as shown in Fig. 12) or keeping the stable moving object in the scene (as shown in the result for the water surface sequence in Fig. 15). In contrast, if the reason for the ghost appearance is a physical change in the scene, MOUM must update the reference image immediately.

6.3 Quantitative analysis

The robustness of MOUM compared with MOG and W4 is quantitatively measured through a pixel-based metric based on *recall, precision*, and *F-measure*. *Recall* measures the number of correctly classified foreground pixels as a percentage of the number of foreground pixels in the ground truth. *Precision* measures the number of correctly classified foreground pixels as a percentage of the total number of pixels classified as foreground. *F-measure* is given by

$$F = 2 * \frac{recall * precision}{recall + precision}. \tag{17}$$

F-measure is computed for the i2r and wallflower datasets. The background subtraction method with the highest number of pixels correctly classified as foreground has the highest *F-measure*. As MOUM updates the reference image immediately to overcome various problems, the foreground objects detected by MOUM are accurately extracted. Figure 16 shows a comparison of the *F-measure* among MOG, W4, and MOUM for the i2r and wallflower datasets. As shown in Fig. 16, MOUM has the highest *F-measure*. From the qualitative and quantitative results, we can conclude that MOUM has a better performance than MOG and W4 for outdoor and indoor environments.

Table 1 shows a comparison of the average processing time between MOUM and OUM for the wallflower dataset. The experiments were executed on a Desktop PC with a core i3 3.1 CPU and 4 GB RAM. As shown in Table 1, although OUM and MOUM have nearly the same results, MOUM is much faster than OUM. In addition, MOUM is a robust method for detecting and removing shadows as shown in Fig. 10.

7. Conclusion

In this paper, we proposed a robust method for updating background models and detecting cast shadows. The proposed method called MOUM can be applied to any background modeling technique. The proposed method mainly relies on two background models. The first model is constructed and undergoes learning during the training period based on the W4 method. The second model is an updated version of the first model that adapts to scene changes. MOUM extracts spatial and color features for each detected object and stores these features in a database. MOUM exploits the spatial and color features and Euclidean distance to classify each new detected object into moving background objects, ghosts, and real moving objects. MOUM copes with problems such as gradual and sudden illumination changes, ghost appearance, and non-stationary background objects. A shadow detection and removal scheme that is based on the color and texture features of the detected objects is also presented. Experimental results showed the efficiency and effectiveness of the proposed method in outdoor and indoor scenarios. A limitation of this method is its low performance when a moving object has the same or nearly the same color as the background (color heterogeneity problem). In future, we plan to build a statistical model to estimate the thresholds T_1, T_2, and T_3.

References

1) Y. L. Tian and A. Hampapur: Proc. IEEE Workshop Motion and Video Computing (WACV/MOTION), 2005, p. 30.

2) C. Braillon , C. Pradalier, J. L. Crowley, and C. Laugier: Proc. IEEE Symp. Intelligent Vehicles, 2006, p. 466.

3) S. C. Cheung and C. Kamath: Proc. SPIE Visual Communications and Image Processing, **5308** (2004) 881.

4) C. Stauffer and W. Grimson: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR’99), 1999, p. 246.

5) W. Grimson, C. Stauffer, R. Romano, and L. Lee: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR’98), 1998, p. 22.

6) K. Toyama, J. Krumm, B. Brumitt, and B. Meyers: Proc. 17th IEEE Int. Conf. Computer Vision (ICCV’99), 1999, p. 255.

7) I. Haritaoglu, D. Harwood, and L. Davis: IEEE Trans. Pattern Anal. Mach. Intell. **22** (2000) 809.

8) J. C. S. Jacques Jr, C. R. Jung, and S. R. Musse: Proc. XVIII Brazilian Symp. Computer Graphics and Image Processing (SIBGRAPI’05), 2005, p. 189.

9) A. Monnet, A. Mittal, N. Paragios, and V. Ramesh: Proc. 9th IEEE Int. Conf. Computer Vision (ICCV ’03), 2003, p. 1305.

10) A. Elgammal, D. Harwood, and L. S. Davis: Proc. European Conf. on Computer Vision (ECCV’00), 2000, p. 751.

11) A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu: Int. J. Mach. Vision Appl. **20** (2009) 395.

12) A. Hamad and N. Tsumura: Proc. 1st IEEE Asian Conf. Pattern Recognition (ACPR’11), 2011, p. 584.

13) A. Hamad and N. Tsumura: in preparation for publication.

14) R. S. Choras: Int. J. Biol. and Biomed. Eng. **1** (2007) 6.

15) A. Joshi and N. Papanikolopoulos: IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 2055.

16) A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara: IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 918.

17) N. Martel-Brisson and A. Zaccarin: IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1133.

18) R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti: Proc. IEEE Conf. Intelligent Transportation Systems, 2001, p. 334.

19) O. Schreer, I. Feldmann, U. Goelz, and P. Kauff: Proc. 4th IEEE Int. Symp. Video Processing and Multimedia Communication, 2002, p. 371.

20) T. Horprasert, D. Hardwood, and L. S. Davis: Proc. Int. Conf. Computer Vision Frame-Rate Workshop, 1999, p. 1.

21) A. Joshi and N. Papanikolopoulos: IEEE Int. Conf. Robotics and Automation (ICRA'08), 2008, p. 987.

22) M. C. Padma and P. A. Vijaya: Int. J. Comput. Sci. and Eng. **2** (2010) 115.

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

24) C. Schuldt, I. Laptev, and B. Caputo: Proc. 17th Int. Conf. Pattern Recognition (ICPR'04), 2004, p. 32.

25) L. Liyuan, H. Weimin, Y. Irene, and T. Qi: IEEE Trans. Image Process. **13** (2004) 1459.

26) http://www.openvisor.org

27) D. Xu, X. Li, Z. Liu, and Y. Yuan: Pattern Recognit. Lett. **26** (2005) 5.

Figure Captions

Fig. 1 Overview of the proposed method.

Fig. 2 Reference image restoration using the proposed method: (a) sample of an indoor video of a human who stops walking for a while, (b) background image restored after training the first 100 frames.

Fig. 3 Illustration of silhouette extraction steps: (a) original image, (b) separated foreground, (c) removal of noise by opening operation followed by closing operation, (d) detected foreground edge, (e) filling and dilating the silhouette of the human body.

Fig. 4 (Color online) Extraction of a detected object BO_i^j and the corresponding object in the color frame CO_i^j: (a) input frame, (b) extracted foreground, (c) detected object in the binary frame (BO_i^j), (d) corresponding object in the color frame (CO_i^j).

Fig. 5 (Color online) Examples of background-modeling difficulties: (a) ghost appearance as a result of a moving object stability during most of the training period, (b) ghost appearance as a result of a sudden illumination change, (c) ghost appearance as a result of a physical background change.

Fig. 6 (Color online) Example of updating the reference image in a non-stationary background scenario: (a) first reference image $R_1^c(x)$ after training period, (b) second reference image $R_2^c(x)$ after the updating by MOUM.

Fig. 7 (Color online) Example of ghost appearance as a result of stability of real moving object: (a) reference model $R_1^c(x)$ after the initial training period, (b) ghost appearing in the binary frames (fr_1 to fr_t) as a result of the stability of the black vehicle in color frames, where fr_0 is the frame that precedes the detection of the ghost, (c) updated reference model $R_2^c(x)$ due to the ghost appearance.

Fig. 8 (Color online) Example of ghost appearance as a result of a sudden physical background change: (a) reference model $R_1^c(x)$ after the initial training period, (b) ghost appearing in the binary frames (fr_1 to fr_t) as a result of the cabinet door opening in color frames, where fr_0 is the frame that precedes the detection of the ghost, (c) updated reference model $R_2^c(x)$ due to the ghost appearance.

Fig. 9 (Color online) Example of detecting and removing shadows using MOUM: (a) input frame, (b) reference image, (c) detected shadow regions, (d) after applying shadow removal and morphological operations.

Fig. 10 A comparison of the results obtained using OUM and MOUM methods on a video contains multiple shadow regions: (a) input frame numbers 40, 140, 200, 250, and 330, (b) output frames obtained using OUM method, (c) output frames obtained using MOUM method.

Fig. 11 (Color online) True positive rate (TPR) for different sequences and different values of thresholds T_1, T_2, and T_3: (a) TPR for different ten sequences and different values of T_1, (b) TPR for different ten sequences and different values of T_2, (c) TPR for different ten sequences and different values of T_3.

Fig. 12 (Color online) Example of updating reference image several times as a result of the appearance of ghosts: (a) input video frames at times 100, 677, 1100, and 2050, (b) first reference image obtained after 100 frames, first updated reference image, and the second updsted reference image, (c) output frames.

Fig. 13 (Color online) Comparison between the results obtained by Xu *et al.* method and the results obtained by MOUM method on: (a) highway video (shadow detection dataset), (b) outdoor walking person (KTH dataset), (c) indoor walking person (KTH dataset), (d) intelligent room (shadow detection dataset).

Fig. 14 (Color online) Comparison between MOG, W4, and MOUM using i2r dataset.

Fig. 15 (Color online) Comparison between MOG, W4, and MOUM using wallflower dataset.

Fig. 16 *F-measure* comparison between three background subtraction methods using (a) i2r dataset, (b) Wallflower dataset.

Table 1. Processing time (ms) for OUM and MOUM methods for wallflower dataset.

Method	Bootstrap	Time of day	Light switch	For. Aperture	Waving tree	Camouflage
OUM	470	523	565	369	323	345
MOUM	353	283	313	198	189	208

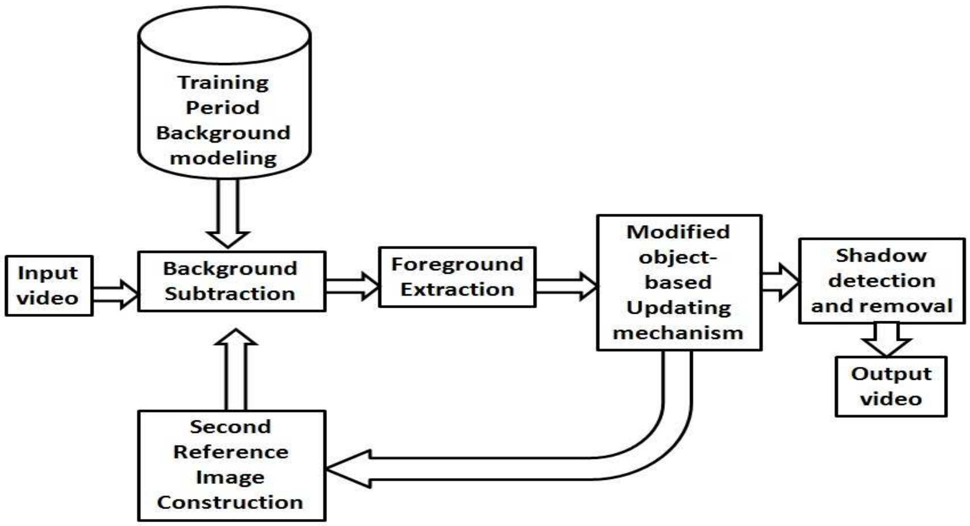

Training Period Background modeling
Input video
Background Subtraction
Foreground Extraction
Modified object-based Updating mechanism
Shadow detection and removal
Output video
Second Reference Image Construction

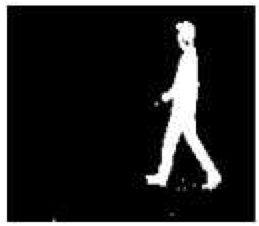
Fig. 1

(a)

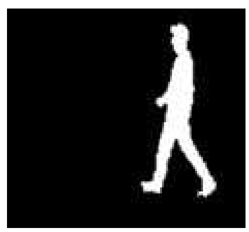
(b)

Fig. 2

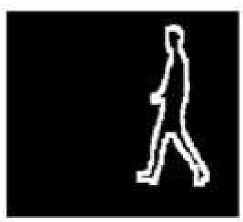
(a)



(b)



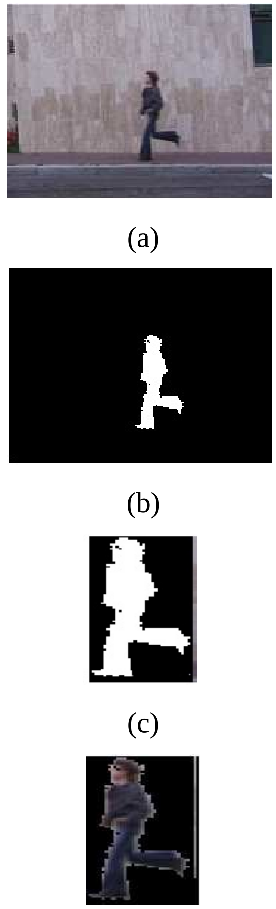
(c)



(d)

(e)

Fig. 3



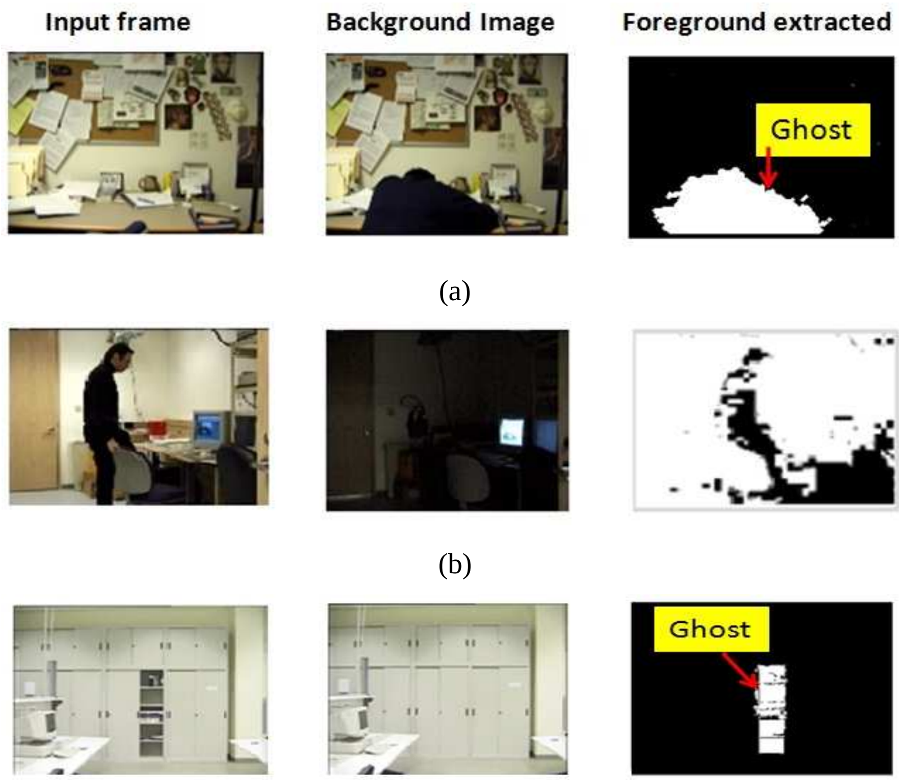
(a)

(b)

(c)

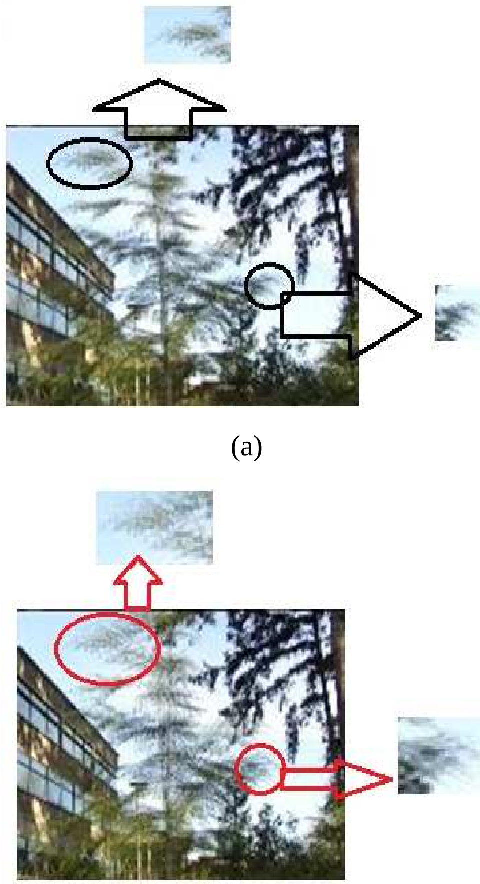
(d)

Fig. 4


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

(c)

Fig. 5

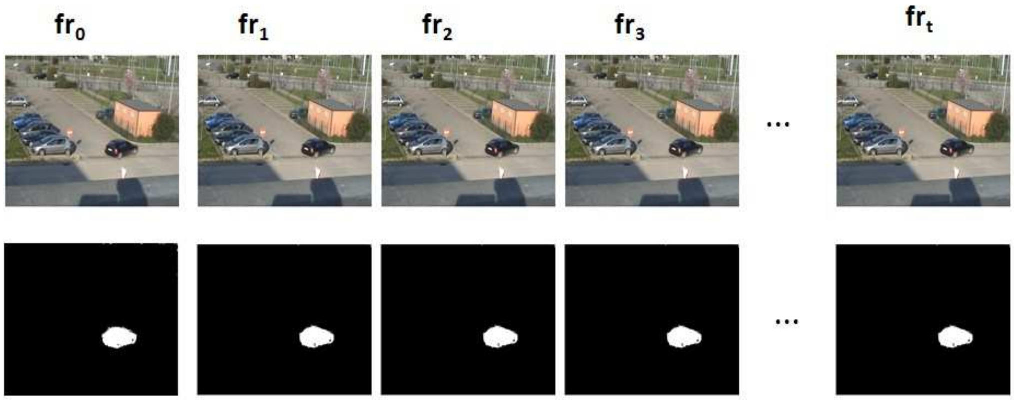


(a)

(b)

Fig. 6

(a)



(b)

(c)

Fig. 7

(a)

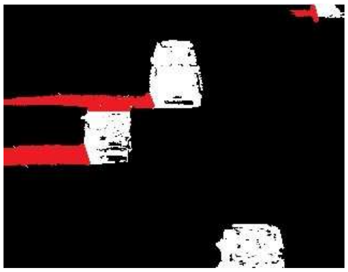
(b)

(c)

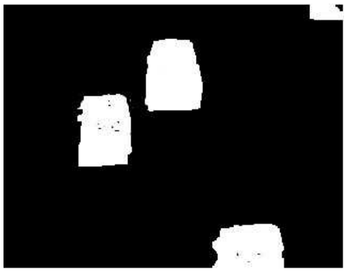
Fig. 8

(a)

(b)

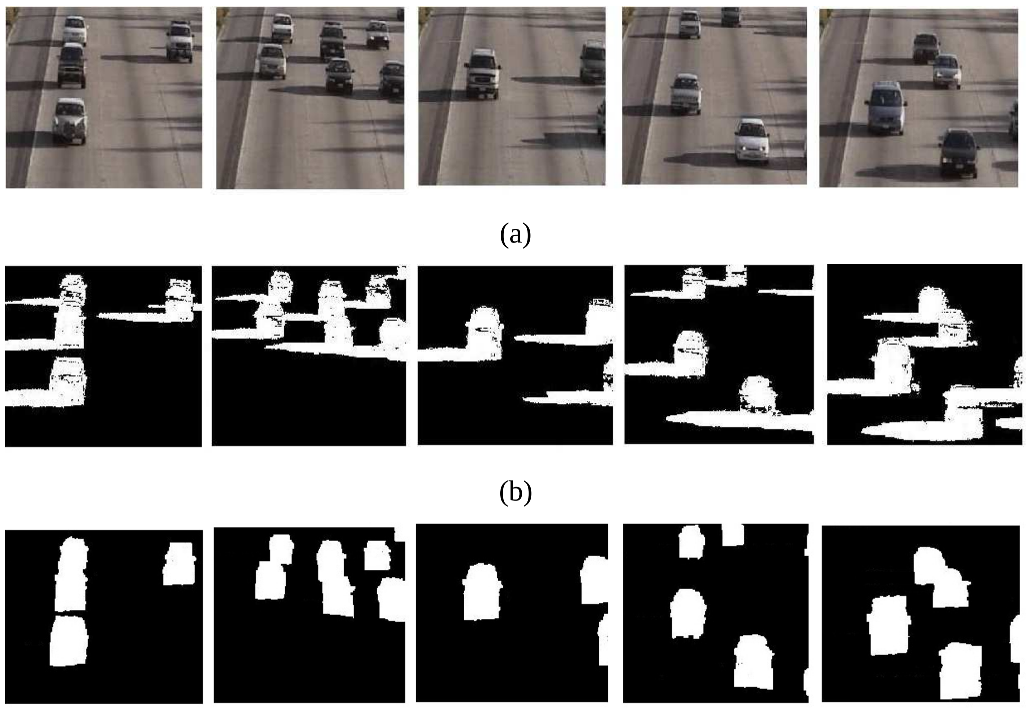


(c)



(d)

Fig. 9

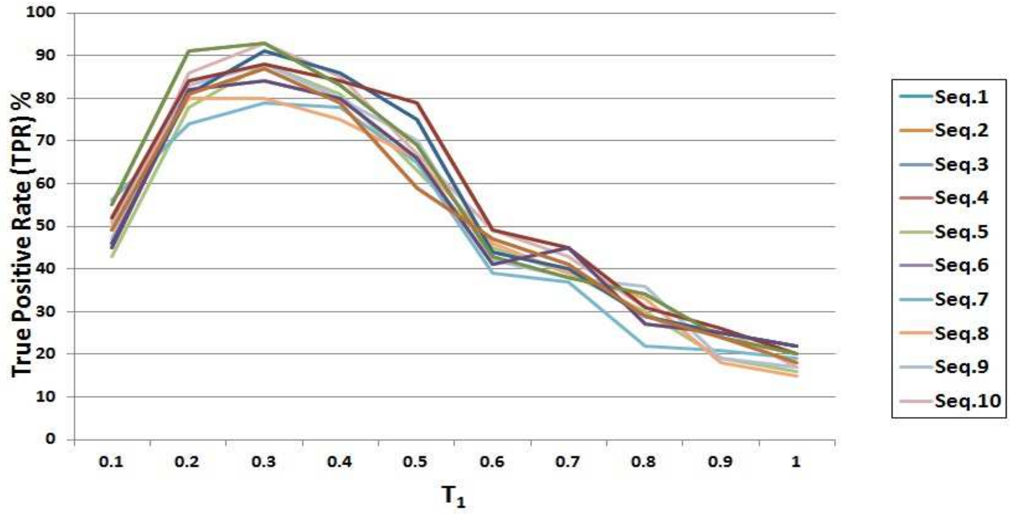


(a)

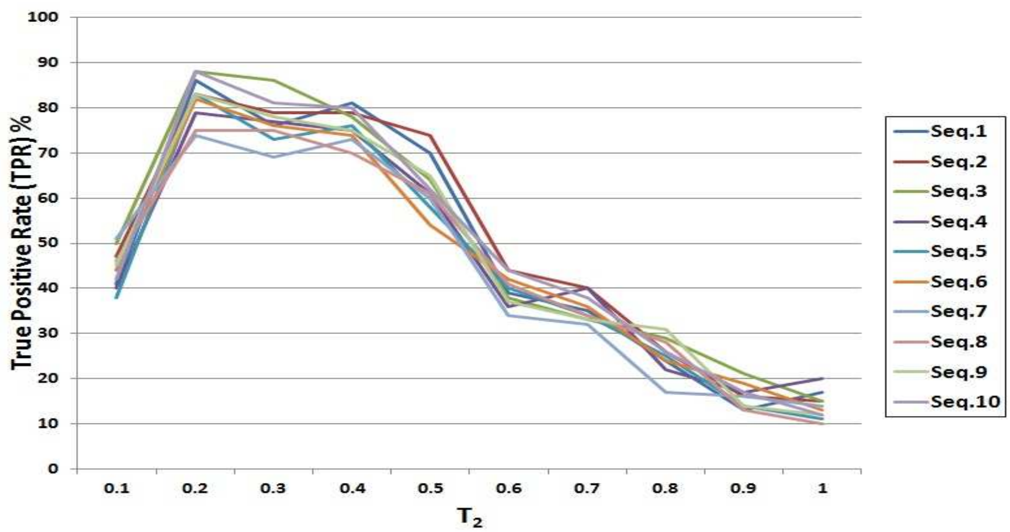
(b)

(c)

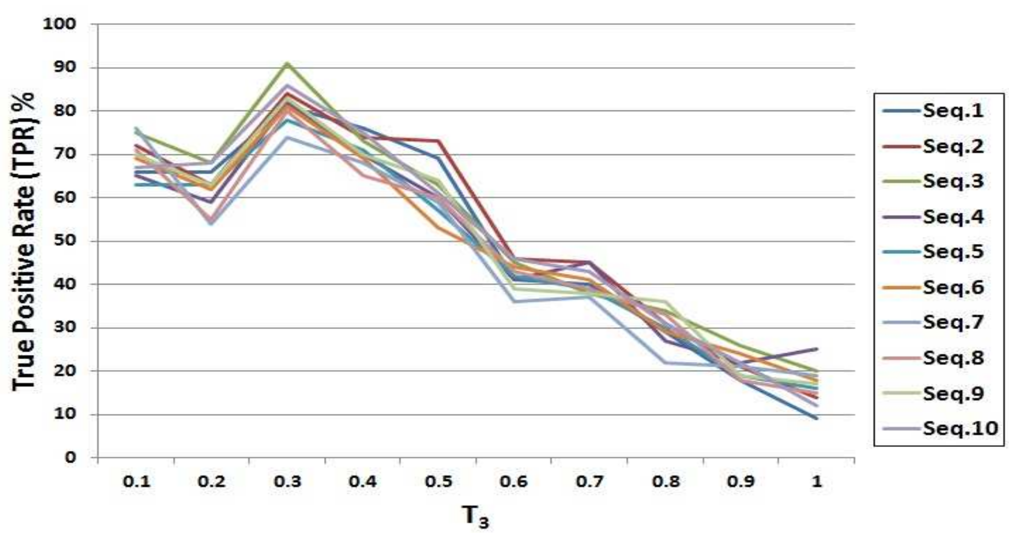
Fig. 10


True Positive Rate (TPR) %
T1
Seq.1
Seq.2
Seq.3
Seq.4
Seq.5
Seq.6
Seq.7
Seq.8
Seq.9
Seq.10

(a)


True Positive Rate (TPR) %
T2
Seq.1
Seq.2
Seq.3
Seq.4
Seq.5
Seq.6
Seq.7
Seq.8
Seq.9
Seq.10

(b)


True Positive Rate (TPR) %
T3
Seq.1
Seq.2
Seq.3
Seq.4
Seq.5
Seq.6
Seq.7
Seq.8
Seq.9
Seq.10

(c)

Fig. 11

(a)

(b)

(c)

Fig. 12

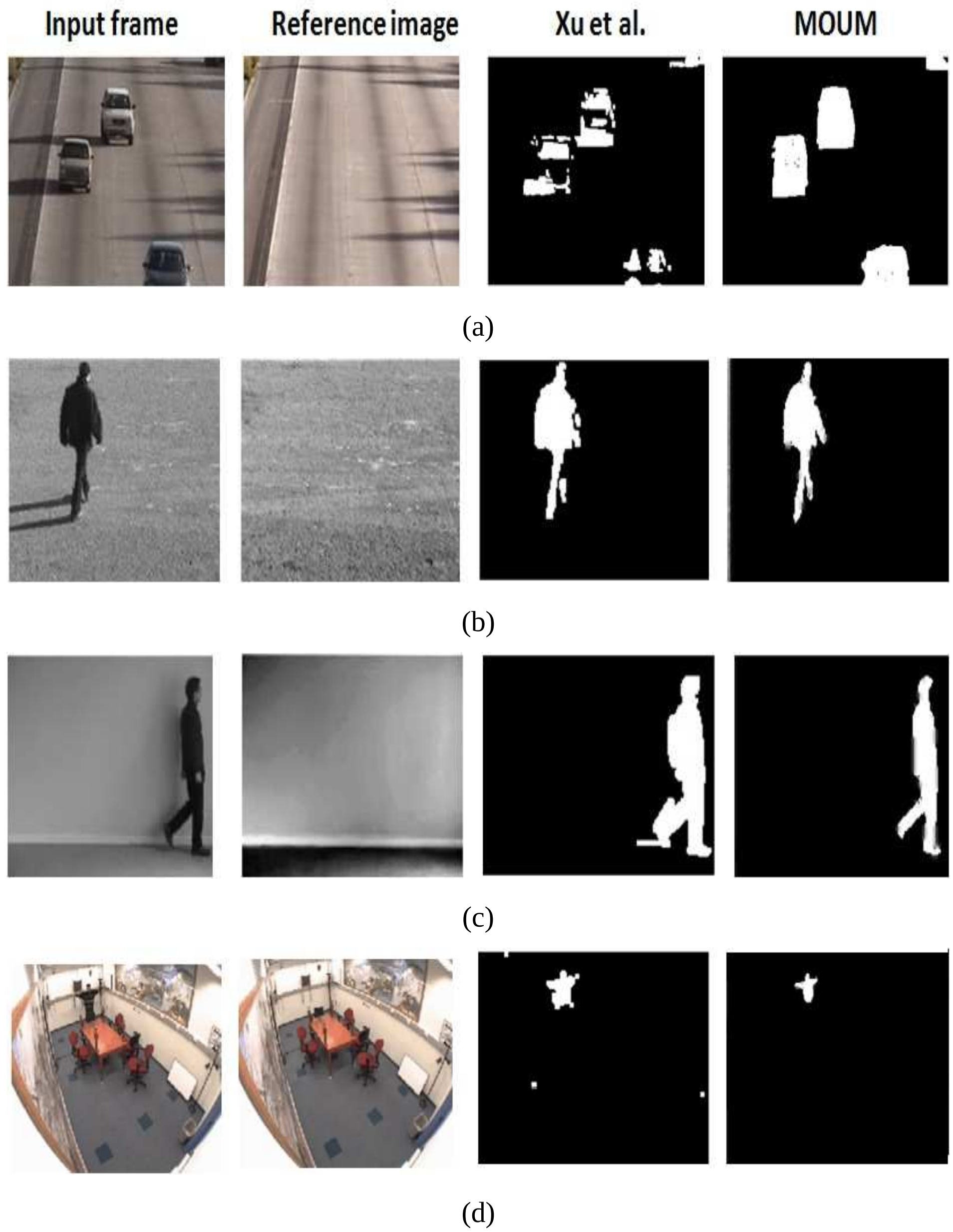


Fig. 13

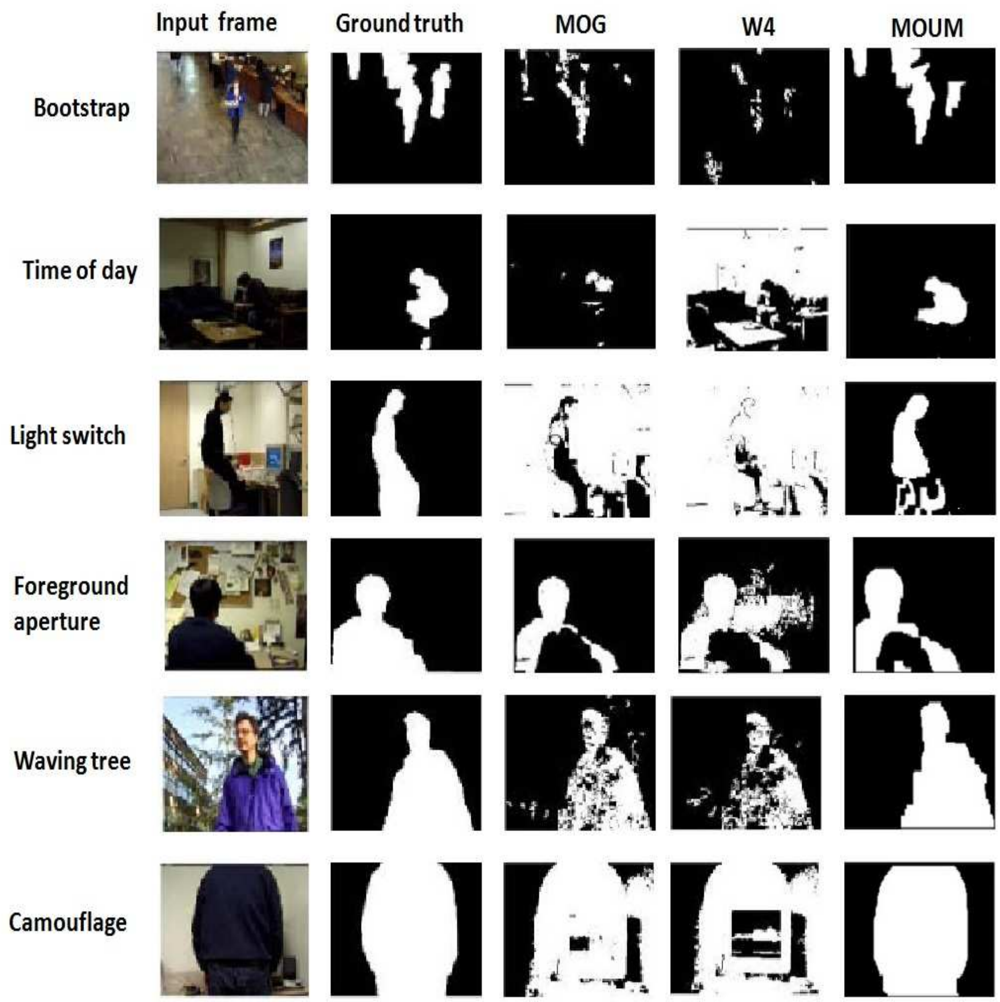

Input frame
Ground truth
MOG
W4
MOUM
Bootstrap
Time of day
Light switch
Foreground aperture
Waving tree
Camouflage

Fig. 14

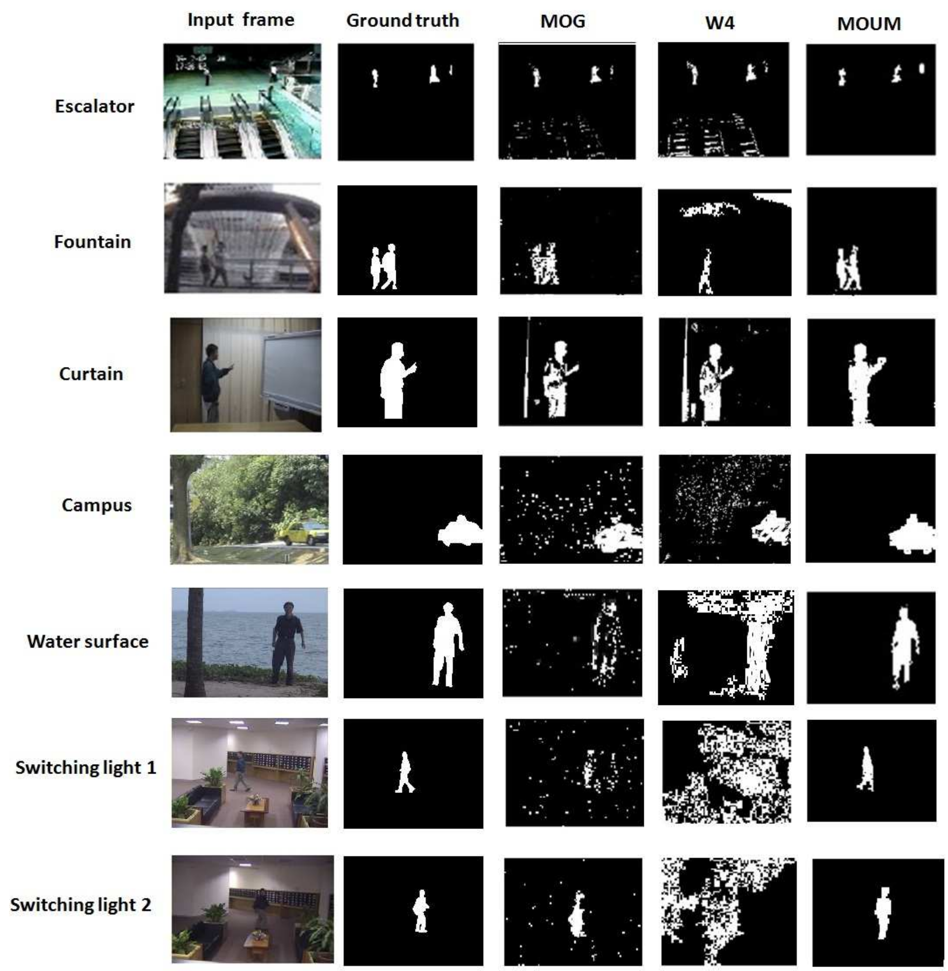
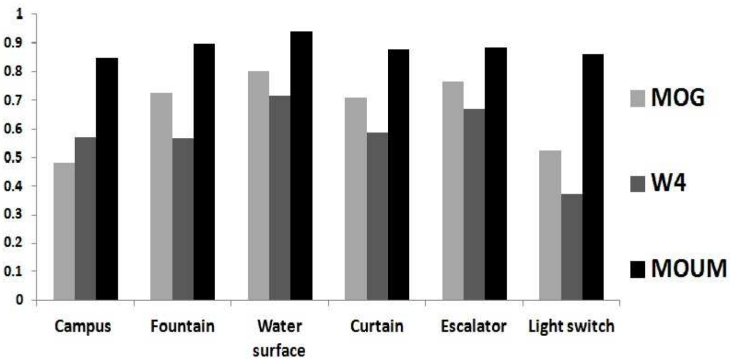
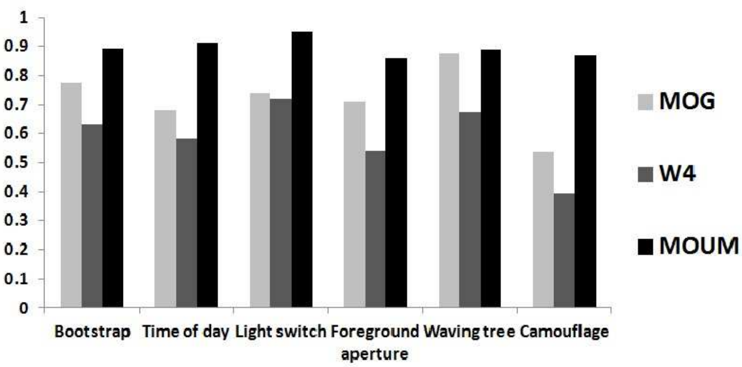

Input frame
Ground truth
MOG
W4
MOUM
Escalator
Fountain
Curtain
Campus
Water surface
Switching light 1
Switching light 2

Fig. 15


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Campus
Fountain
Water surface
Curtain
Escalator
Light switch
MOG
W4
MOUM

(a)


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Bootstrap
Time of day
Light switch
Foreground aperture
Waving tree
Camouflage
MOG
W4
MOUM

(b)

Fig. 16