10

Background Subtraction and Lane Occupancy Analysis

Erhan A. Ince, Nima S. Naraghi and Saameh G. Ebrahimi
Eastern Mediterranean University
North Cyprus

1. Introduction

In the last decade or so traffic monitoring has attracted considerable attention from intelligent transportation systems (ITS). ITSs are generally designed to collect traffic data such as vehicle count, vehicle speed, vehicle path, vehicle density, and vehicle classification. The information gathered may be used by traffic lights controllers, and toll collectors to regulate traffic flow, reduce congestion and improve safety. Traffic information can be obtained through physical devices like buried loop sensors, radars, and infrared detectors however these conventional systems have the disadvantage that they can only count but they cannot differentiate or classify. Nowadays, most ITSs essentially rely on wireless IP cameras which are either fixed or movable type. These systems employ state of the art machine vision technologies to automatically analyze traffic data collected by the camera system(s) and forward their findings to control devices or points. Video based surveillance systems (VBSS) can be categorized as indicated below:

1. Tripwire Systems,
2. Tracking Systems,
3. Spatial Analysis based systems.

In Tripwire systems the camera is used to simulate usage of a conventional detector by using small localized regions of the image as detector sites. Such a system can be used to detect the state of a traffic light (red, yellow, green) check if a reserved section has been violated, and detect wrong-way traffic etc.

Tracking systems detect and track individual vehicles moving through the camera scene. They provide a description of vehicle movements (east bound, west bound, etc.) which can also reveal new events such as sudden lane changes and help detect vehicles travelling in the wrong direction. Tracking systems can also compute trajectories and conclude on accidents when different trajectories cross each other and then motion stops.

Spatial analysis based systems on the other hand concentrate on analyzing the two-dimensional information that video images provided. Instead of considering traffic on a vehicle-to-vehicle basis, they attempt to measure how the visible road surface is being utilized. Whichever approach is employed, the segmentation of mobile objects present in frames of a video sequence is a fundamental step for many video based applications. In the literature this step is referred to as the background subtraction. The process includes creation of a background model and then updating it with each newly arriving frame from a sequence. Newly arriving frame from a video sequence. The updating of the background model can be done exploiting various predictive, non-predictive, recursive and non-recursive algorithms. Current frame pixels with considerable deviation from the background model are considered to belong to the moving objects (vehicles, people etc). Over the years many background estimation algorithms have been proposed. This is mainly because no single algorithm is able to cope with all the challenges in this area. While some of the proposed algorithms are best for indoor applications, others may be better for outdoors. The section that follows provides a general introduction on the classification of background subtraction algorithms and then gives a comparative study of five selected background subtraction algorithms.

2. Classification of background subtraction algorithms

In visual surveillance applications, a common approach for differentiating moving objects from the static part of the video frames is detection by background subtraction. According to (Christani, Bicego & Murino), a background modeling process has three phases:

1. Model representation,
2. Model initialization,
3. Model adaptation.

The first part describes the kind of model used to represent the background; the second part is about the initialization of the assumed model; and the third part is the mechanism used for adapting to illumination changes in the background. (Mittal & Paragios, 2004) state that existing state of the art methods for background adaptation may be classified as either predictive or non-predictive. Predictive algorithms are known to model the scene (background) as a time series and they would make use of a dynamic model to recover the current input based on past observations. The absolute error between the predicted and the actual observation can then be used as a measure of change. On the other hand non-predictive methods do not rely on the order of the input observations but rather try to build probabilistic representation for the observations at a particular pixel.

An alternative way for classifying background adaptation methods is to group them as either non-recursive or recursive. This was first proposed by (Cheung &Kamath, 2004). A non-recursive technique uses a sliding-window approach for background estimation. For non-recursive estimation the *L* previous video frames are first stored in a buffer and then a background image is constructed making use of the temporal variation of each pixel in the buffer. One disadvantage of non-recursive methods is that for slow moving objects a large storage may be required. Recursive techniques do not rely on a buffer for estimating the scene. Instead, they recursively update a single or multiple background model(s) based on each input frame (Elhabian, El-Sayed & Ahmed,2008). Even though recursive techniques have much lower memory requirements, any error in the background model can remain around for a longer time. To alleviate this problem exponential weighting and/or positive decision feedback can be used.

Non-recursive adaptation techniques include temporal differencing (frame differencing), average filter, Median filtering and Minimum-Maximum filtering. Recursive techniques on the other hand include Approximated Median filtering, Single Gaussian, Kalman Filtering, Mixture of Gaussians (MoG), Clustering based segmentation methods, and Hidden Markov Models.

2.1 Study of selected adaptation algorithms

Although many background subtraction methods are listed in the literature, foreground detection specially for outdoor scenes is still a very challenging problem. The performance of video based surveillance systems will vary based on several environmental changes like the ones listed below:

1. Variable lighting conditions, during sunset and sunrise,
2. Adverse weather conditions such as fog, rain, snow, etc,
3. Non-stationary backgrounds such as swaying grass, leaves and branches,
4. Presence of camera vibration due to wind and heavy vehicles.

Another important consideration while trying to choose an appropriate background estimation method is the time required for processing a frame. If a system has to run in real-time, its computational complexity should not be too high.

In this section we will first provide a summary for each of the five different adaptation algorithms that we have chosen to discuss and follow with the segmentation quality performance evaluations.

2.2 Temporal median filtering

(Nixon & Aguado, 2005) et al., state that finding the background given a sequence of frames is an example application of statistical operators. For conventional median filtering the median is the center of a rank-ordered list of values usually taken from a template centered on the point of interest. A temporal median filter (TMF) on the other hand is different from a conventinal median filter in that each point in the TMF produced image is the median of the points in a mask placed over the same positions in the N sample frames.

TMF computes the median intensity for each pixel from all the stored frames in a buffer. Considering the computation complexity and storage limitations it is not practical to store all the incoming video frames and make the decision accordingly. Hence the frames are stored in a limited size buffer. In some cases the number of stored frames is not large enough (buffer limitations), therefore the basic assumption will be violated and the median will estimate a false value which has nothing to do with the real background model. An example where temporal median filtering algorithm fails to extract a proper foreground mask is shown in Fig. 1.

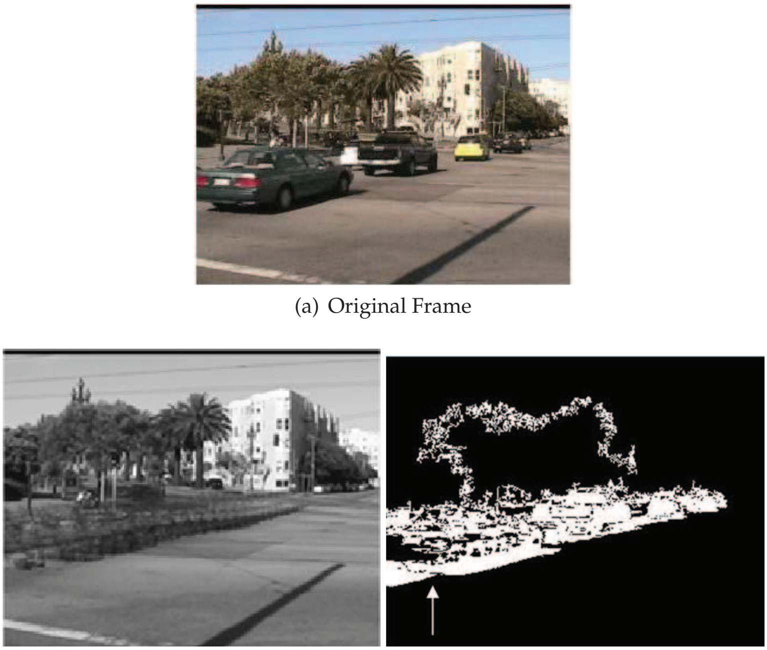
In comparison to average or temporal average filters a background image that is obtained using TMF would have more detail and less blur.

2.3 Approximated median filtering

Shortly after the non-recursive median filtering became popular among the background subtraction algorithms, (McFarlane & Schofield, 1995) presented a simple recursive filter for estimating the median of each pixel over time. This filtering process which is named as the approximated median filtering (AMF) would simply increment the background model intensity by one, if the incoming intensity value (in the new input frame) is larger than the previous existing intensity in the background model. The reverse is also true, meaning that when the intensity of the new input is smaller than that of the background model the corresponding intensity will be decreased by one. Over time this process will converge to the median of the observed intensities.

Unlike TMF, this approach does not require storing any frames in a buffer and tries to update the estimated background model online. Hence it is extremely fast and suitable for real time applications. AMF used for estimating the background scene in a simulated indoor environment is shown in Fig. 2.

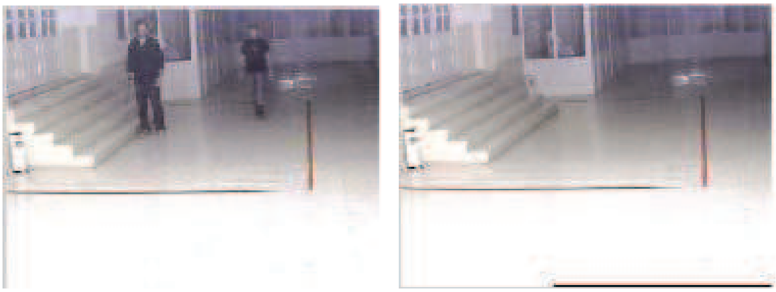
This method has also been adopted by some for background subtraction for urban traffic monitoring due to its considerable speed. A sample output showing the foreground background separation for an outdoor scene has been provided in Fig. 3.



(a) Original Frame

(b) Estimated Background (c) Mask of Extracted Foreground

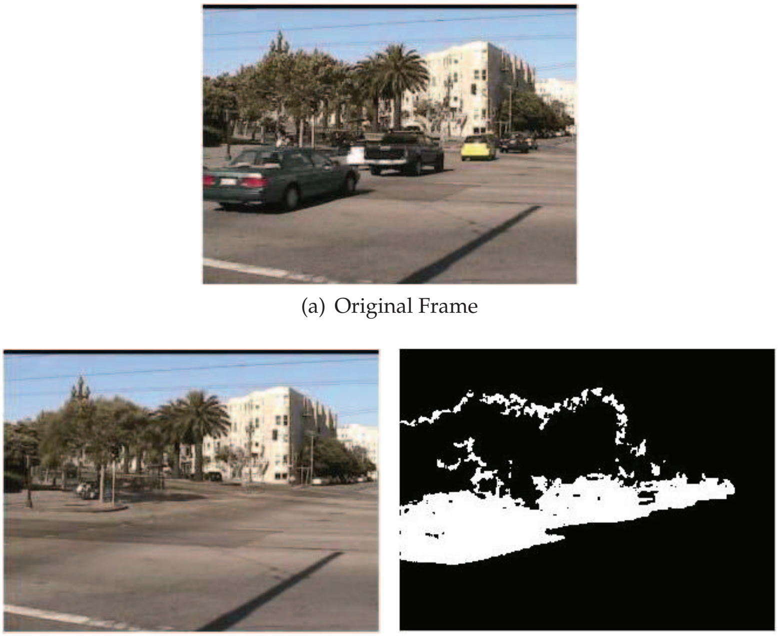
Fig. 1. Foreground-Background detection using temporal median filtering



(a) Original Scene

(b) Estimated Background

Fig. 2. Background detection using approximated median filtering for indoors



(a) Original Frame

(b) Estimated Background

(c) Mask of Extracted Foreground

Fig. 3. Foreground-Background separation using approximated median filtering

Note that even though the AMF produced background is more correct the technique still has some difficulty in handling shaking leaves and swaying branches.

2.4 K-Gaussian mixture model

The Mixture of Gaussians technique was first introduced by (Stauffer & Grimson, 1999). It sets out to represent each pixel of the scene by using a mixture of normal distributions so that the algorithm will be ready to handle multimodal background scenes. In the mixture model each pixel is modeled as a mixture of K Normal distributions. Typically values for K varies from 3 to 5. For $K < 3$, the mixture model is not so helpful since it cannot adapt to multimodal environments and if K is selected a value over 5, often the disadvantage of processing speed reduction (not able to be performed in real time) outweighs the improvement in quality of background model. At any time t, K Gaussian distributions are fitted to the intensities seen by each pixel up to the current time t:

$$P(X_{i,t}) = \sum_{i=1}^{K} w_{i,t} \cdot \eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t}) \tag{1}$$

In the adaptive K-MoG model $X_{i,t}$ is the current pixel value vector which consists of Red, Green and Blue components, $w_{i,t}$ is an estimate of the weight of the i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$ and $\Sigma_{i,t}$ are the mean value and the covariance matrix of the i^{th} Gaussian in the mixture. $P(X_{i,t})$ denotes the probability of observing the current pixel value vector given the mixture of K Gaussian distributions and $\eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t})$ is a Gaussian probability density function.

$$X_{i,t} = (x_{i,t}^{R}, x_{i,t}^{G}, x_{i,t}^{B}) \tag{2}$$

$$\mu_{i,t} = (\mu_{i,t}^{R}, \mu_{i,t}^{G}, \mu_{i,t}^{B}) \tag{3}$$

$$\Sigma_{i,t} = \begin{pmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_G^2 & 0 \\ 0 & 0 & \sigma_B^2 \end{pmatrix} \tag{4}$$

$$\eta(X_{i,t}, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{-1/2}} \exp^{1/2 \cdot (X_{i,t}-\mu_{i,t})^T \cdot \Sigma^{-1} \cdot (X_{i,t}-\mu_{i,t})} \tag{5}$$

Background/foreground separation consists of two independent steps: 1) estimating the parameters of *K*distributions; and 2) evaluating the likelihood of each distribution to represent the background.

2.4.1 Parameter updating

Since at the start of modelling all the Gaussians have an equal probability for representing the background the weights $w_{i,t}$, i ϵ (1...K), are all set to the value $\frac{1}{K}$ and the variances are set randomly to high values. Then every new pixel value vector $X_{i,t}$ is checked against the existing *K* Gaussian distributions until a match is found (a match is defined as a pixel value vector whose Euclidean distance is within 1.5 standard deviations of a distribution). The parameters of the matched component are then updated using the recursive equations below:

$$\mu_{i,t} = (1-\rho) \cdot \mu_{i,t-1} + \rho \cdot X_{i,t} \tag{6}$$

$$\Sigma_{i,t} = (1-\rho) \cdot \Sigma_{i,t-1} + \rho \cdot diag\{(X_{i,t} - \mu_{i,t})^T (X_{i,t} - \mu_{i,t})\} \tag{7}$$

$$\rho = \alpha \cdot (X_{i,t} | \mu_{i,t-1}, \Sigma_{i,t-1}) \tag{8}$$

Here α represents the user-defined learning rate and has a value in the range $0 < \alpha < 1$. ρ on the other hand is a learning rate for the parameters.

For the case when there are no matches the Gaussian distribution with the least weight is replaced by a new component with a mean equal to the current pixel vector. The variance for this new distribution is set high and the weight is set to a low prior value. Finally, the weight of all the *K* Gaussians at time t are updated and normalized using:

$$w_{i,t} = (1-\alpha) \cdot w_{i,t-1} + \alpha \cdot M_{i,t} \tag{9}$$

$$w_{i,t} = \frac{w_{i,t}}{\Sigma_{m=1}^{K} w_{m,t}}$$

When there is a match $M_{i,t}$ can be assumed as 1 and 0 otherwise.

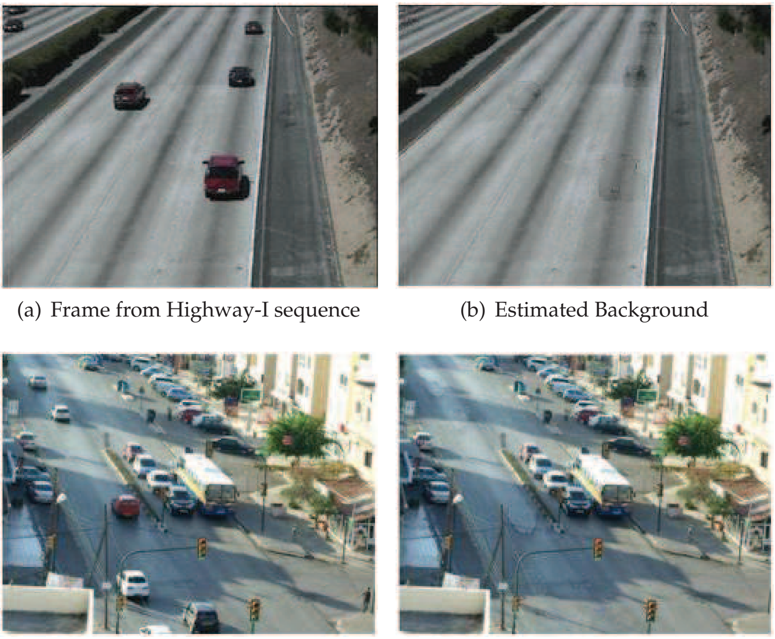
2.4.2 Background estimation

Once the parameters for all the Gaussian distributions are updated the ones that are most likely produced by background processes are determined. First, the *K* Gaussians are sorted in descending order by the value of $\frac{w_{i,t}}{\Sigma_{i,t}}$ and then the first *B* distributions are chosen to be in the background model using the value of *B* as given by:

$$B = \arg\min_{b} \{\Sigma_{k=1}^{b} w_k > T\} \tag{10}$$

where, T can assume any value from the interval 0.5 - 1.

Generally the segmented foreground would contain some noise. It is possible to get rid of this noise by making use of standard morphological operations as suggested by (Gonzales, 2002). Figure 4 depicts some estimated backgrounds using the K-Gaussians mixture model. For all video sequences $K = 3$, $\alpha = 0.05$, $\beta = 1.5$, and $T = 0.85$ values were used in the adaptive K-MoG model.



(a) Frame from Highway-I sequence

(b) Estimated Background

(c) Frame from Yeni Izmir Sequence

(d) Estimated Background

Fig. 4. MoG based Background Generation

2.5 Progressive background estimation

This method was first introduced by (Chung et al., 2002). A progressive background image is generated by utilizing a histogram to record the changes in intensity for each pixel of the image, however, unlike its other histogram based background generator counterparts, progressive method does not directly use the input frames to create the histogram. The progressive method constructs the histograms from the preprocessed images also referred to as the partial backgrounds.

In order to generate the partial backgrounds, the progressive method follows the following steps. First, the current frame $I(t)$ at time t of an input video sequence $S(t)$ is captured into the system and this image is compared with the previous frame image, $I(t-1)$ to generate a current partial background $B(t)$. Each pixel at location i at time t of the corresponding partial background is called $b_i(t)$ and is computed using;

$$b_i(t) = \begin{cases} bg & ,|p_i(t) - b_i(t-1)| < \epsilon \\ non - bg & ,\text{otherwise.} \end{cases} \tag{11}$$

Here bg stands for pixels related to the background image whose intensity value difference from the previous partial background $b_i(t-1)$ does not exceed a small predefined threshold ϵ. If the incoming intensity varies from the partial background more than the selected threshold, the corresponding pixel will be classified as $non - bg$. There are several possible ways to assign value to bg pixels; one is to take the minimum intensity between the new $b_i(t)$ and $b_i(t-1)$, another way is to average these two values and yet another is by simply taking the new value as $b_i(t)$. This last approach requires less computational time and hence is more suitable for real-time processing. For $non - bg$ pixels a specific value should be assigned, so that it will be possible to distinguish them since we are not interested in them. To separate them from bg pixels, usually they are assigned 0 or -1. After all the pixels have been classified and the numbers are assigned to them, the whole partial background at time t can be created as;

$$B_i(t) = \bigcup b_i(t) \qquad , i \epsilon I(t) \tag{12}$$

By creating the partial background images, the moving objects are discarded due to their intensity differences from the background and only the pixels which are more likely to be a part of background will be kept. In some cases slow moving objects or similarity among foreground objects and background scene may cause some parts of moving objects to be misclassified as background related pixels. This problem can be avoided if color information is used as shown below:

$$b_i(t) = \begin{cases} bg & , \cap_c |p_i{}^c(t) - b_i{}^c(t-1)| < \epsilon^c \\ non - bg & , \text{otherwise.} \end{cases} \tag{13}$$

Here, c is the different components of the RGB. In other words the classification is done separately for each color channel and then their intersection is obtained in order to set aside the pixels that vary in all channels in comparison to previous partial background.

The next step of the progressive background estimation method would be generating a histogram called $h_p(t)$ using the partial backgrounds obtained from the previous step. The index p indicates that there is a histogram for every pixel of the image and t stands for time. For each pixel at time t a certain number of generated partial background depending on the size of our buffer are processed and then the histograms are created per pixel location in time. This process is shown by Fig. 5.

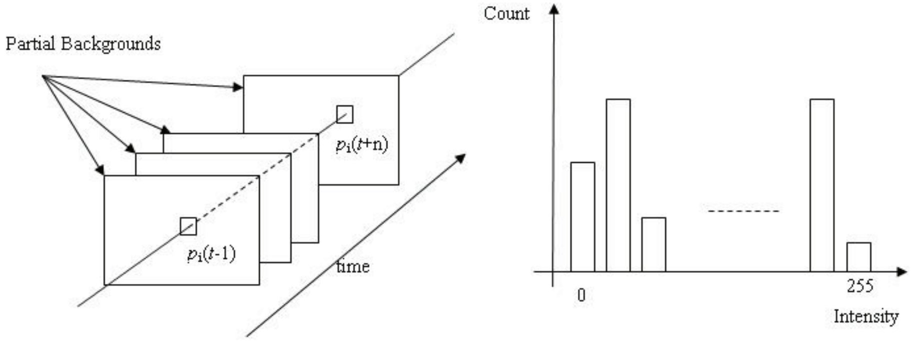


Fig. 5. Partial Background Images and Histogram

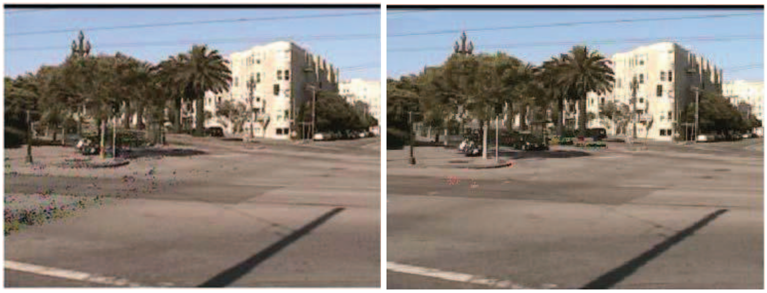
The histogram updating procedure is done simultaneously with the generation of histograms. For each pixel the incoming intensity from partial background is checked by the algorithm to discover whether the new intensity is within the local neighborhood of the previous background intensities or not. If the mentioned condition is satisfied (the intensity belongs to the neighborhood) then the frequency of that intensity is incremented by a constant factor. If the constraint is violated and the newly gained intensity is located outside the boundaries of our neighborhood domain, the recorded frequency for corresponding pixel in the histogram will be decreased by a factor less than mentioned incrementing factor. The preceding discussion can be summarized as in:

$$v = v + A\delta(b_i(t), a) - D \tag{14}$$

Here, v is the count (frequency) of the intensity index a, in the histogram. A represents the rising factor while on the contrary D is the descending factor. The δ function in equation 14 is defined as:

$$\delta(l, r) = \begin{cases} 1 & , |l - r| < \lambda \\ 0 & , \text{otherwise.} \end{cases} \tag{15}$$

After the histograms are generated and updated, the maximum frequency of each histogram along with its corresponding intensity for each pixel in the image are recorded in a table. The histogram table can be utilized as a reference for intensities which are responsible for background generation at any time. Whenever the background image is required, the recently updated intensity values in the table are used to generate the desired background.
At the beginning of the processing some cells of the table may not have a value and hence the background image contains leakages (undesired black dots). This problem occurs because the histograms are built over partial backgrounds which include black parts in the position of moving objects but as time passes, intensities related to the background image come to the pixel view more and more. Therefore this leakage effect will be gradually removed. A stable background image would be expected when the counts recorded in the histogram table are approximately 75-80 % of a predetermined upper limit.
Figure 6, depicts an example where leakage problem is resolved after 5 frames of the video sequence.



(a) Existence of leakage (b) Leakages removed after 5 frames

Fig. 6. Estimated background using progressive method

2.6 Group-based histogram estimation

Group Based Histogram (GBH) algorithm construct background models by using histogram of intensities for each pixel on the image. However, unlike the other histogram based methods, in group based histogram, each of the individual intensities is considered along with its neighboring intensity levels and forms an accumulative frequency. The frequency of coming intensity is summed up with its neighboring frequency to create a Gaussian shape histogram.

The accumulation can be done by using an average filter of width $2w+1$ where w stands for half width of the window. The output $n^*_{u,v}(l)$ of the average filter at level l can be expressed as:

$$n^*_{u,v}(l) = \Sigma^{w}_{r=-w} n_{u,v}(l+r) \qquad ,0 \leq (l+r) \leq (L-1) \tag{16}$$

Here $n_{u,v}(l+r)$ is the count of the pixel having the intensity $(l+r)$ at the location (u,v), and L is the total number of possible intensity levels. The maximum probability density $p^*_{u,v}$ of a pixel can be computed through a simple division of the occurrence for a pixel by the total frequency N^*:

$$p^*_{u,v} = \frac{\max_{0 \leq l \leq L-1}\{n^*_{u,v}(l)\}}{N^*} \tag{17}$$

Since the filter smoothens the histogram curve, if the width of the averaging window is chosen to be less than a preset value, the location of the maximum will be closer to the center of the Gaussian model (which corresponds to background value). Therefore the mean intensity of the background model will be:

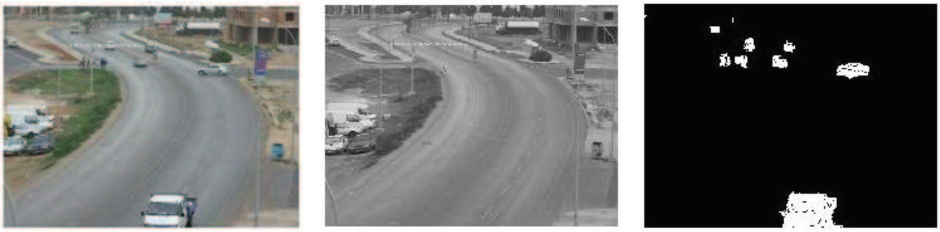
$$\mu_{u,v} = \arg\max_{l}\{n^*_{u,v}(l)\} \tag{18}$$

Choice of the window size is a critical task since a smaller window width can save the processing time, while a larger window will lead to smoother GBH and therefore more accurate estimation of the real value of the pixel related to the background model. The mean intensity can be computed by selecting the maximum frequency of the smoothened histogram. When a new intensity$l$ is captured, the algorithm does not process all the possible intensities, just the new one and its adjacent intensities which fall in the selected window.

If the current pixel intensity is represented by $I_{(u,v)}$ where (u,v) corresponds to the location of pixel on the image,then background objects are extracted by using:

$$BG(u,v) = \begin{cases} 1 & ,if|I(u,v) - \mu(u,v)| < 3\sigma(u,v) \\ 0 & ,\text{otherwise.} \end{cases} \tag{19}$$

Figure 7, depicts an estimated background using GBH method for a video frame taken in Famagusta city.



(a) Original Frame (b) Estimated background (c) Extracted foreground mask

Fig. 7. Background / Foreground separation using group based histogram method

3. Segmentation performance

According to (Mezaris et al., 2003), comparison of algorithms trying to achieve the same task are possible either using standalone evaluations or by the application of relative evaluation methods. In this work the latter approach was used.

To have a precise comparison between BE/FS algorithms video sequences with ground truths are necessary. These are video sequences that are created by first recording a scene without any foreground objects and then superimposing animated moving objects on the recorded background manually. Therefore, the exact location of the pixels related to foreground items is known (in other words the ground-truths of these sequences are available). A second advantage of using a test sequence with ground truth is that the superimposed objects would not contain shadows and hence the focus will be on the BE/FS performance only.

The comparisons of the afore mentioned algorithms were based on a synthetically generated video sequence ,video7long.avi, which was developed for the *Background Competition of the 4th ACM International Workshop on Video Surveillance Sensor Networks* and a custom recorded video taken in the electrical and electronics engineering department of Eastern Mediterranean University. The custom video sequence contains an indoor scene showing students walk through the corridor, stop for a while, then continue walking again. To compare our achieved results with the ground truths two well known scales *recall* and *precision* were employed for each pixel. Recall is a measure of completeness and is defined as the number of correctly identified pixels (true positives) divided by the total number of pixels that actually belong to the foreground objects (pixels in ground truth). On the other hand precision is defined as the ratio of correctly detected pixels in the region of interest to the number of all pixels in relevant detection regions.

$$\begin{aligned} R &= \tfrac{TP}{TP+FN} \\ P &= \tfrac{TP}{TP+FP} \end{aligned} \tag{20}$$

where, TP, FN and FP stand for true positive, false negative and false positive respectivelt.

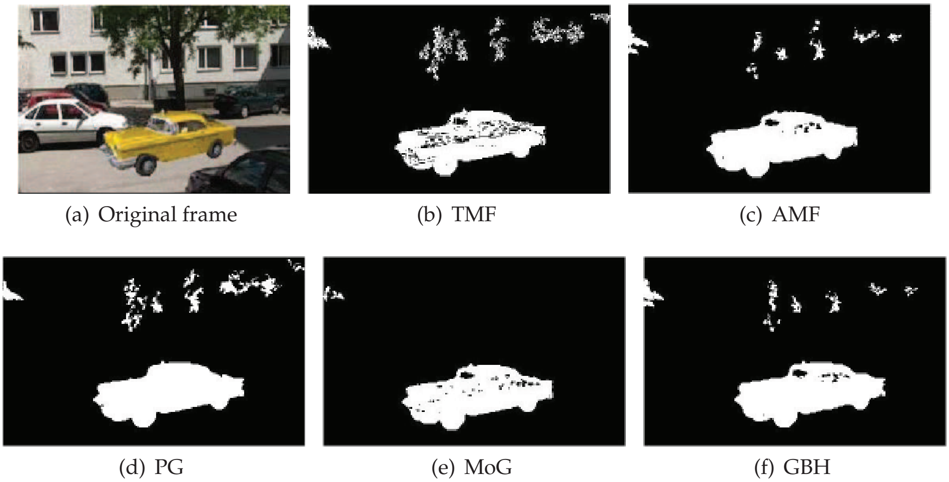
During evaluation of the five different algorithms introduced in section 2, all the frames belonging to video7long were used and average recall and precision percentages were computed for each technique separately. These results have been summarized in Table 1. To test which one of the algorithms generate the background model faster, we have also applied them to a video sequence which does not start with an empty frame and average processing times required to process a single frame have been recorded for each method. During the simulations also the number of frames required to generte an acceptable estimate of the foreground mask has been noted. These values are summarised in Table 2 .

A visula comparison showing how well each algorithm can cope with multi-modal background scenes (shaking leaves, swaying branches etc.) is depicted in Fig. 8. Similarly, Fig. 9 depicts how well each algorithm perform under indoor environment with transient stops.

A quick look at the results indicate that the MoG Model is not robust against transient stops but it suppresses the multi-modal backgrounds best. Also for indoor environments with transient stops the AMF technique surpasses the PG, MoG and GBH methods.

BG Est. Method	Recall (%)	Precision (%)
TMF	77.88	49.65
AMF	82.34	58.19
PG	72.30	60.92
MOG	85.38	77.96
GBH	86.18	74.42

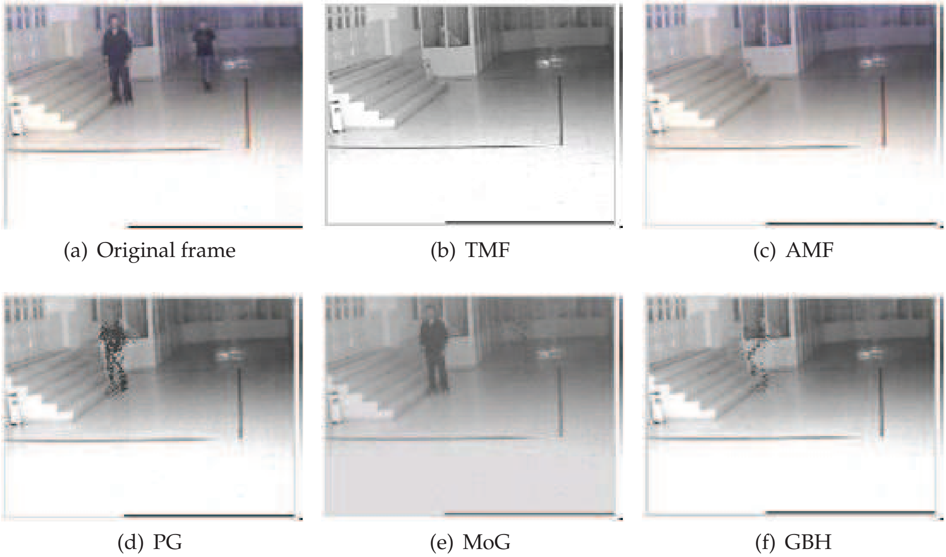
Table 1. Average recall and precision results for five background estimation algorithms



(a) Original frame (b) TMF (c) AMF

(d) PG (e) MoG (f) GBH

Fig. 8. Visual comparison between algorithms in handling multi-modal background scenes



(a) Original frame (b) TMF (c) AMF

(d) PG (e) MoG (f) GBH

Fig. 9. Visual comparison between algorithms in handling transient stops

Method	Average Processing Time Per Frame (sec)	Frames to Generate Acceptable FG
TMF	1.8570	12
AMF	0.0490	44
PG	2.7422	10
MOG	1.4152	9
GBH	4.0214	6

Table 2. Comparison of algorithm with respect to time

4. Cast shadow detection and removal

Cast shadows are generated due to occlusion of sun light by moving objects. The resultant shadows are the projected areas on the scene which move along side of the moving object. From a camera's point of view, cast shadows have many of the same characteristics as vehicles. They move in similar patterns and directions, and they are considerably different from the background. The cast shadows that are projected on the road surface can change in size based on how high or how low the illuminating light source might be. In cases when the cast shadows stretch, two or more independent objects can appear to be connected together and this makes classification a more difficult job. In fact, incorrect detection of moving cast shadows as part of the foreground scene will cause serious problems in all applications that deal with recognition, classification and traffic analysis.

4.1 Classification of shadow detection algorithms

As stated by (Prati et al., 2003), cast shadow detection algorithms can be classified using a two layer taxonomy. On the first layer are the *deterministic* and *statistical* methods. In the second layer, the statistical approaches can be subdivided into *parametric* and *non-parametric*. Similarly, deterministic methods can be sub-classified as *model-based* and *non-model based*.Choosing a model-based approach undoubtedly achieves the best results, but is, most of the time, too complex and time consuming compared to the nonmodel-based. In this chapter we will summarize the HSV color space and Shadow Confidence Score (SCS) based shadow removal algorithms and provide results based on custom and/or standard video sequences.

4.1.1 Shadow detection in the HSV space

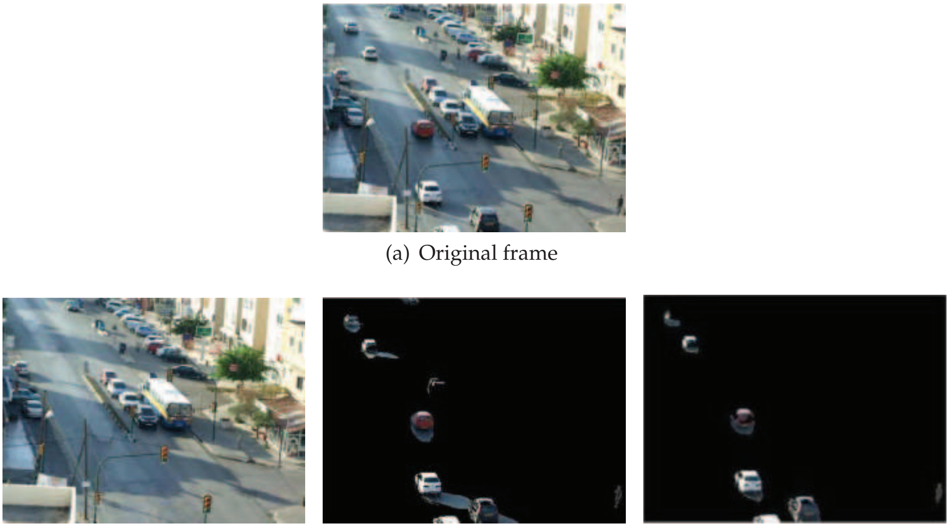
The HSV system described by (Cucchiara et al., 2001) is an example of the deterministic nonmodel-based approaches. HSV color space corresponds closely to human perception of color and it has high accuracy in detecting shadow pixels. The shadow point mask defined by the HSV method is as follows:

$$SP_k(x,y) = \begin{cases} 1 \ , \left\{ \alpha \leq \frac{I_k{}^V(x,y)}{B_k{}^V(x,y)} \leq \beta \right\} \cap \left\{ (I_k{}^S(x,y) - B_k{}^S(x,y)) \leq \tau_S \right\} \cap \\ \quad \left\{ (I_k{}^H(x,y) - B_k{}^H(x,y)) \leq \tau_H \right\} \\ \\ 0 \ , \text{otherwise.} \end{cases} \tag{21}$$

where $I_k{}^H(x,y)$, $I_k{}^S(x,y)$ and $I_k{}^V(x,y)$ are the HSV components of the input frame at time instant k and location (x,y) and $B_k{}^H(x,y)$,$B_k{}^S(x,y)$ and $B_k{}^V(x,y)$ are the HSV components of the background frame.

The lower bound α is used to define a minimum value for the darkening effect of shadows on the background and it is almost proportional to the light source intensity and the upper

bound β prevents the system from identifying noise which slightly changes the background in the shadow regions. It has been shown that the chrominance values for both the shadow and non-shadow pixels would vary only slightly. The choice of τ_H and τ_S is done according to this assumption. This choice is complicated and the threshold values have to be chosen by trial and error.
Figure 10 , shows the background subtraction and shadow removal processes applied to a frame from a custom video taken at the Yeni Izmir junction of Famagusta city.



(a) Original frame

(b) BG estimate

(c) FG with cast shadows

(d) FG objects

Fig. 10. Shadow detection and removal using HSV method

4.1.2 Shadow confidence score based shadow removal

The shadow confidence score (SCS) based shadow removal was first proposed by (Andrew et al., 2002). This method requires that firstly, a background subtraction algorithm is used to generate the moving foreground mask (MFM) and then extracted blobs corresponding to binary mask locations in the color image are converted to YC_bC_r . For creating the SCSs one needs to combine the characteristics of the cast shadow in the luminance, chrominance and gradient density domains. The characteristics of the cast shadow in the luminance, chrominance and gradient density domain dictates that:

1. Luminance values of the cast shadow pixels are lower than those of the corresponding pixels in the background image,
2. The chrominance values of the cast shadow pixels are identical or only slightly different from those of the corresponding pixels in the background,
3. The difference in gradient density values of the cast shadow pixels and the corresponding background pixels is relatively low. The difference in gradient density values between the vehicle pixels and the corresponding background pixels is relatively high.

The three scores $S_{L,i}(x,y)$, $S_{C,i}(x,y)$ and $S_{G,i}(x,y)$ can be calculted using the equations below:

$$S_{L,i}(x,y) = \begin{cases} 1 \; L_i \leq 0 \\ \frac{(T_L - L_i(x,y))}{T_L} \; , 0 < L_i(x,y) < T_L \\ 0 \; L_i(x,y) \geq T_L \end{cases} \tag{22}$$

where, $L_i(x,y) = l_{I,i}(x,y) - l_{B,i}(x,y)$.

$$S_{C,i}(x,y) = \begin{cases} 1 \; C_i \leq T_{C1} \\ \frac{(T_{C2} - C_i(x,y))}{T_{C2} - T_{C1}} \; , T_{C1} < C_i(x,y) < T_{C2} \\ 0 \; C_i(x,y) \geq T_{C2} \end{cases} \tag{23}$$

where, $C_i(x,y) = |Cb_{I,i}(x,y) - Cb_{B,i}(x,y)| + |Cr_{I,i}(x,y) - Cr_{B,i}(x,y)|$.

$$S_{G,i}(x,y) = \begin{cases} 1 \; GD_i(x,y) \leq T_{G1} \\ \frac{(T_{G2} - GD_i(x,y))}{T_{G2} - T_{G1}} \; , T_{G1} < GD_i(x,y) < T_{G2} \\ 0 \; GD_i(x,y) \geq T_{G2} \end{cases} \tag{24}$$

where, $GD_i(x,y) = GD_{I,i}(x,y) - GD_{B,i}(x,y)$.

Figures 11 and 12, depict the shadow removal process applied to frames extracted from a custom and a standard video sequence. The threshold values used by the SCS calculator have been summarized in Table 3 for each video used.

Video Sequence	T_L	T_{C1}	T_{C2}	T_{G1}	T_{G2}
Yeni Izmir Junction	180	9.5	19	0.3	0.6
Highway I	200	7.5	15	0.5	1.0

Table 3. SCS Algorithm Parameters

It is possible that sometimes parts of the objects can be misclassified as shadows (incorrect decisions led to undesired erosion on the foreground mask). To fix this problem a convex hull can be fitted to the remaining shadow free foreground mask as described by (Ince et al., 2009). Generating a polygon that completely and closely surrounds a given set of points in 2D is called convex hull fitting. In the literature there are many algorithms for convex hull generation. Some well-known ones include incremental, gift wrapping, divide and conquer and quick hull algorithms. The one adopted here is the incremental algorithm. The processing starts with a single point and then using two more points a triangle is created. Next a new point is selected. If the new point is inside the hull there is nothing to do. Otherwise one must delete all the edges that the new point can see and add two new edges to connect the new point to the remainder of the old hull. This process is then repeated for all the remaining new points.

5. Analysis of lane occupancy

In a conventional traffic lights controller, the lights either change at constant cycle times or at times proportional to the length of each leg of the intersection. Such approaches clearly are not perfect for optimizing traffic flow. Waiting times proportional to lane length may work well for a single-lane road but when roads with multiple lanes are considered this solution would not be optimal. Assuming that in real life each leg of an intersection is being monitored simultaneously by fixed surveillance cameras, this section presents a framework as suggested by (Ince et al., 2009) to analyse the lane fullness for individual legs of an intersection. This way adaptive signalling based on the computed values would become possible. During simulations the segmentation of foreground objects from frames of the surveillance video

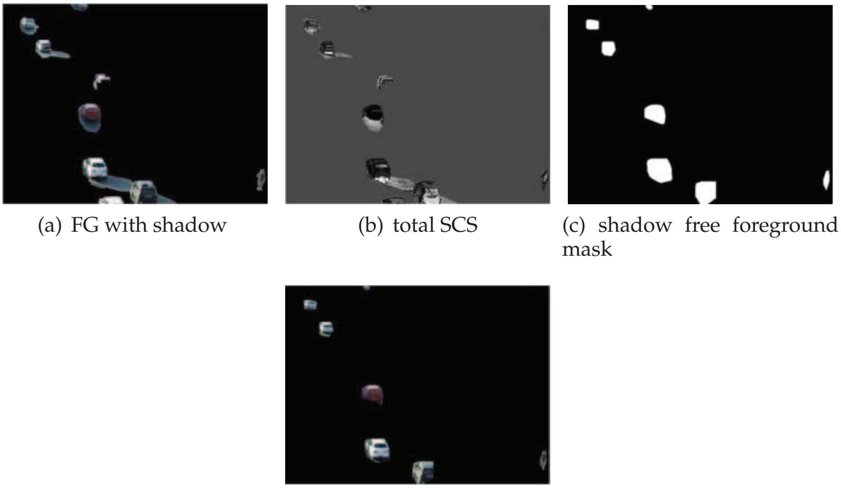

(a) FG with shadow
(b) total SCS
(c) shadow free foreground mask
(d) FG objects

Fig. 11. Shadow detection and removal using SCS method

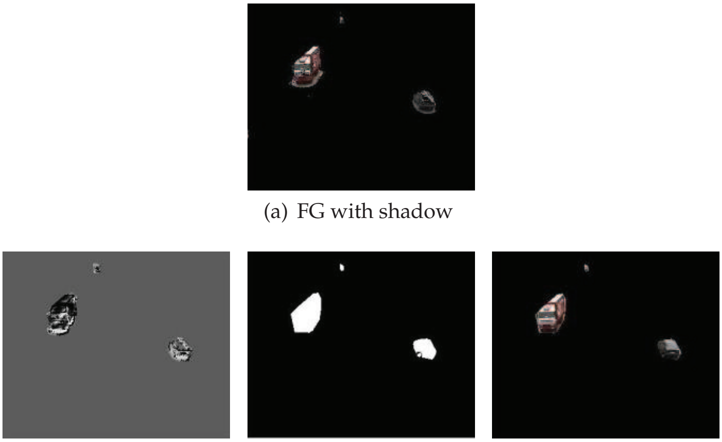
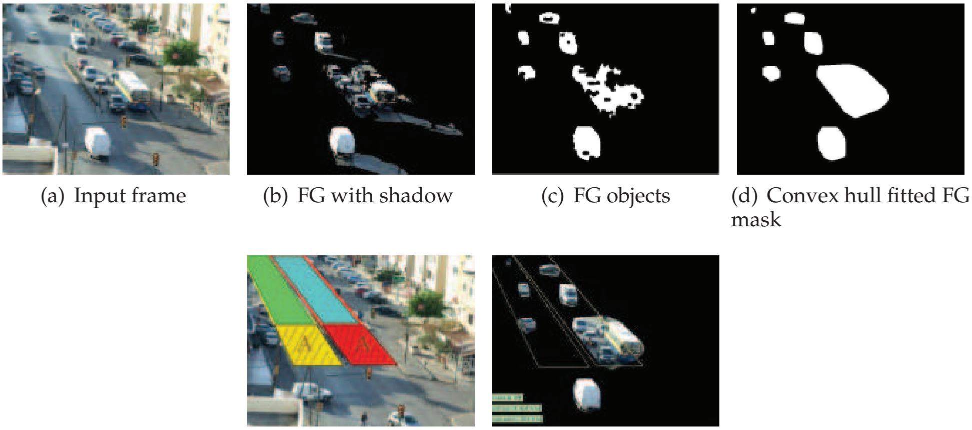

(a) FG with shadow
(b) total SCS
(c) shadow free foreground mask
(d) FG objects

Fig. 12. Shadow detection and removal using SCS and Highway I sequence.

are done using an adaptive K-Gaussian mixture model and cast shadows present in the segmented foregrounds are removed using the shadow confidence score earlier discussed.
In systems using fuzzy logic each leg houses two sensors behind traffic lights separated by a distance D. The sensor at distance D from the light counts the number cars coming to the intersection and the second counts the cars passing the traffic light. The amount of cars between the sensors is determined by the difference of the readings. However, this approach can not differentiate between a truck, a bus or a car. Hence determining what percent of the road is full based on size becomes fairly difficult.
A better approach that would not require any information on the type of cars present behind the traffic lights would be the use of the foreground mask(with shadows removed) together with two lane masks for determining how much each lane and the detected foreground overlap outside a designated region A. Afterwards we test to see if any of the foreground objects fall in this designated region. If region A contains no moving objects it is assumed 100% full. Otherwise the overlap between the extracted FG over region A and the ground truth mask of region A can be computed.
The application of the fullness analysis to the north leg of the intersection for frame #1890 of the video sequence shot at Yeni Izmir junction is depicted in Fig. 13.



(a) Input frame (b) FG with shadow (c) FG objects (d) Convex hull fitted FG mask

(e) lane masks (f) degree of fullness

Fig. 13. Lane fullness analysis.

While computing percentage fullness of left and right lanes, the depth for the designated region A which is close to the traffic lights, can be adaptively changed after tracing every five minute of the video sequence and obtaining a speed for the traffic flow. This will allow estimates taken during different times of the day to be more realistic.

6. Summary

The simulation results indicate that critical tradeoffs are always present between the accuracy of estimated background model and the real time performance of the method. The choice of algorithm for background modeling should be made according to the desired application. For instance if it is desired to monitor an indoor scene environment, one of the most suitable choices would be the AMF, however, the same algorithm is not a proper choice when it comes to outdoor scenes due to the fact that it cannot deal with multi-modal background scenes or cope with changes in weather condition. Among the five algorithms discusses in section 2, the MoG algorithm is best in handling the multi-modal backgrounds.

Both the HSV and the SCS based shadow removal algorithms need to use different thresholds and this constitudes a disadvantage since for each video sequence the set of thresholds have to be optimized empirically. On the other hand the HSV runs fast and accurately and if the selection of thresholds can be automized based on the content of each frame and its layers then it would constitude a good solution for real time systems.

For various examples it was observed that after shadow detection and removal step, applying a convex hull to the shadow free FG mask will help enhance the final FG mask by fixing errors like partial erosions and/or holes. This holds in general regardless of the shadow detection and removal method adopted.

7.References

Christani, M.; Bicego, M. & Murino, V. (2003). Multi level background initialization using Hidden Markov models, *In first ACM SIGMM Int. workshop on video surveillance*, pp. 11-20, 2003.

Mittal, A. & Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation, *In Proceedings of the Int. Conf. on Computer Vision and Pattern Recognition*, pp. 302-309, 2004.

Cheung, S-C. & Kamath, C. (2004). Robust techniques for background subtraction in urban traffic video, *In Proceedings of Electrical Imaging: Visual Communications Image Processing* , pp. 881-892, 2004.

Elhabian, S. Y.; El-Sayed, K. M. & Ahmed, S.H. (2008). Moving Object Detection in Spatial Domain using Background Removal Techniques-State-of-Art, *Recent Patents on Computer Science* , Vol. 1, pp. 32-54, 2008.

Nixon, M. & Aguado, A. (2005). *Feature Extraction and Image Processing*, Elsevier Inc., 0-7506-5078-8, UK.

McFarlane, N. & Schofield, C. (1995). Segmentation and tracking of piglets in images, *In Proc. of Machine Vision Applications*, 8(3), pp. 187-193, 1995.

Stauffer, C. & Grimson, W. (1999). Adaptive background mixture models for real-time tracking, *In Proceedings of the Int. Conf. on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

Gonzales, R.C. (2002). *Digital Image Processing*, Prentice Hall Inc.,0-20-118075-8, New Jersey USA.

Chung, Y., Wang, J. & Chen, S. (2002).Progressive Background Images Generation, *In Proc. of 15th IPPR Conf. on Computer Vision*, 2002.

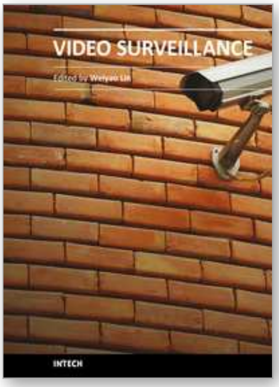
Mezaris, V., Kompatsiaris, I. & Strintzis, M.G. (2003). Still Image Objective Segmentation Evaluation using Ground Truth, *In 5th COST 276 Workshop*, pp. 9-14, 2003.

Prati, A., Mikic, I., Trivedi, M.M.& Cucchiara, R. (2003). Detecting Moving Shadows: Algorithms and Evaluation, *In IEEE tracsactions on pattern analysis and machine intelligence* , Vol. 25, No. 7, pp. 918-923, 2003.

Cucchiara, R., Grana, C., Neri, G., Piccardi, M. & Prati, A. (2001). The Sakbot System for Moving Object Detection and Tracking, *In Video-Based Surveillance SystemsâĂŤComputer Vision and Distributed Processing* , pp. 1458-157, 2001.

George, S.K.F., Nelson, H.C.Y.,Grantham, K.H.P. & Andrew, H.S.L. (2002). Effective moving cast shadow detection for monocular colour traffic image sequences, *In Optical Engineering. 41(6)* , pp. 1425-1440, June 2002.

Seifnaraghi, N., Ebrahimi, S.G.& Ince, E.A. (2009). Novel traffic lights signalling technique based on lane occupancy rates, *In ISCIS 09* , pp. 592-596 , Sept 2009.



Video Surveillance

Edited by Prof. Weiyao Lin

ISBN 978-953-307-436-8
Hard cover, 486 pages
Publisher InTech
Published online 03, February, 2011
Published in print edition February, 2011

This book presents the latest achievements and developments in the field of video surveillance. The chapters selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. Besides the introduction of new achievements in video surveillance, this book also presents some good overviews of the state-of-the-art technologies as well as some interesting advanced topics related to video surveillance. Summing up the wide range of issues presented in the book, it can be addressed to a quite broad audience, including both academic researchers and practitioners in halls of industries interested in scheduling theory and its applications. I believe this book can provide a clear picture of the current research status in the area of video surveillance and can also encourage the development of new achievements in this field.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821