Fast and Robust Background Updating for Real-time Traffic Surveillance and Monitoring

Suchendra M. Bhandarkar and Xingzhi Luo
Department of Computer Science, The University of Georgia
Athens, Georgia 30602-7404, USA

Abstract

Background updating is an important aspect of dynamic scene analysis. Three critical problems: sudden camera perturbation, sudden or gradual illumination change and the *sleeping person* problem, which arise frequently in real-world surveillance and monitoring systems, are addressed in the proposed scheme. The paper presents a multi-color model where multiple color clusters are used to represent the background at each pixel location. In the proposed background updating scheme, the updates to the mean and variance of each color cluster at each pixel location incorporate the most recently observed color values. Each cluster is assigned a weight which measures the time duration and temporal recurrence frequency of the cluster. The *sleeping person* problem is tackled by virtue of the observation that at a given pixel location, the time durations and recurrence frequencies of the color clusters representing temporarily static objects are smaller compared to those of color clusters representing the true background colors when measured over a sufficiently long temporal history. The camera perturbation problem is solved using a fast camera motion detection algorithm that allows the current background image to be registered with the background model maintained in memory. Sudden illumination changes are handled by using an adaptive histogram template whereas gradual illumination changes are automatically resolved with the adaptive background model. The background updating scheme is shown to be robust even when the scene is very busy and also computationally efficient, making it suitable for real-time traffic surveillance and monitoring systems. Experimental results on real traffic monitoring and surveillance videos are presented.

1 Introduction

Separating foreground from background (also known as figure-ground discrimination) is an important though difficult problem in computer vision. In this paper we propose a background updating scheme for a real-time traffic surveillance and monitoring system. In particular, we address three critical problems that are confronted by real-world surveillance and monitoring systems. The first problem is sudden camera perturbation, which occurs occasionally but causes typical background updating schemes to fail. In the context of traffic monitoring and surveillance, cameras mounted on bridges or overpasses are typically subject to structural vibrations caused by especially heavy moving vehicles, resulting in sudden and random camera perturbations. The second problem is dealing with sudden and gradual changes in ambient illumination. The third problem is the *sleeping person* problem [9] where a moving object stops in the scene and becomes motionless for some duration of time, causing it to be improperly merged with the background image. The sleeping person problem arises frequently in the context of automated traffic monitoring when moving vehicles stop temporarily at traffic lights or intersections.

2 Background and Previous Work

Broadly speaking, there are two categories of online methods to model the background image. The first category models the background image using a single color value per pixel [3, 4, 5, 6, 7, 9], whereas the second category uses multiple color values per pixel. For dynamic scenes with high levels of random noise, a single value is not sufficient to represent the background color at a given pixel location. Stauffer and Grimson [8] propose an adaptive on-line multi-color model in which the background color of each pixel is modeled using multiple clusters each with a Gaussian distribution. If an observed color matches or falls within the kth cluster, then the mean (i.e., centroid) and variance of the cluster are updated using the equations $\mu_{k,t} = (1-\rho_k)\mu_{k,t-1} + \rho_k X_t$ and $\sigma^2_{k,t} = (1-\rho_{k,t})\sigma^2_{k,t-1} + \rho_{k,t}(X_t - \mu_{k,t})^T(X_t - \mu_{k,t})$, where $\mu_{k,t}$ is the kth cluster centroid at time t, X_t is the new color observed at time t at that pixel location, $\sigma^2_{k,t}$ is the variance of the cluster at time t, $\rho_{k,t}$ is a parameter that is measured from the distribution of the kth cluster and is given by $\rho_{k,t} = \alpha(\mathcal{N}(x = X_t|\mu_{k,t-1}, \sigma^2_{k,t-1}))$ and α is the learning rate. Note that $\mathcal{N}(\mu_k, \sigma^2_k)$ is a Gaussian (normal) distribution with mean μ_k and variance σ^2_k. An observed color X_t matches or falls within the kth cluster if $X_t \in [\mu_{k,t-1} - 2.5\sigma_{k,t-1}, \mu_{k,t-1} + 2.5\sigma_{k,t-1}]$. If none of the existing clusters match the observed color, then a new cluster is created. Each cluster is assigned a weight, which is updated as $\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t})$, where $M_{k,t} = 1$ if the observation X_t matches the kth cluster and $M_{k,t} = 0$ otherwise. Variations of Stauffer and Grimson's technique that are computationally more efficient have also been proposed in the literature [1, 2].

Typical background updating methods can adapt rapidly to changes in the background. However, when a moving object becomes temporarily motionless, for example, when a moving vehicle stops at a traffic light, these adaptive methods respond to this change by eventually merging the colors of the temporarily static object with the background color. In the context of a traffic monitoring system, a vehicle may be temporarily stationary at a traffic light for as long as 2 minutes, i.e., for 3600 frames at a video sampling rate of 30 frames/second. In the case of Stauffer and Grimson's method [8], the weight of a newly created and matched cluster (corresponding to a temporarily static object) increases rapidly at the learning rate α whereas the weights of unmatched clusters, some of which may represent the true background color, decrease at the same rate. Let T be the minimum fraction of the image frame data that could be accounted for by the background. For a temporarily static object, the initial weight w_s of its corresponding cluster could be as low as 0, whereas the total weight of all the other clusters w_b could be at most 1. After n frames, $w_b = (1-\alpha)^n$ and $w_s = 1-(1-\alpha)^n$, and if $w_s > (1-T)$ then the temporarily static object will be merged with the background. Thus, the number of elapsed frames n_s for a temporarily static object to be merged with the background can be shown to be $n_s = \log(T)/\log(1-\alpha)$ frames. If we require that $T = 0.6$ and $n_s > 3600$, then $\alpha < 0.00014$ which is a very small learning rate (note that $\alpha = 0.002$ in [8]). However, too small a learning rate results in a low background convergence rate, delaying the adaption of the background to gradual changes in ambient illumination. The longer the temporarily static object stays in the scene, the smaller the value of α needs to be in order to avert the sleeping person problem. In very busy scenes where the true background color is not exposed long enough, the time taken to converge to the true background image is excessive when α is small. Thus, reducing the learning rate to tackle the sleeping person problem is not practically feasible. Conversely, if the background model is highly adaptive to the presence of temporarily static objects (i.e., α is large), an object tracking system may cease to detect a moving object using background subtraction after the object has been temporarily motionless and consequently merged with the background. Therefore, a reasonable solution is to be able to differentiate between a temporarily static object and an actual change in the background color.

3 Overview of the Approach

In the proposed approach, the camera perturbation is modeled as a Euclidean transformation. A background image with dimensions greater than the actual image frame is created and continuously updated. The first image frame is aligned with the center of the background image. For each successive image frame, a fast algorithm is used to estimate the camera motion parameters and consequently determine the alignment of the new image frame with the stored background image. The background updating procedure is performed on those locations within the background image that overlap with the new image frame.

Sudden changes in ambient illumination cause the color histogram of the observed image to be skewed. A histogram correction technique, similar to histogram equalization, is used to restore the distorted image. However, in contrast to histogram equalization which modifies the pixel color values in an input image to ensure a uniform color distribution in the output image, the designed histogram correction technique adjusts the pixel color values in the input image such that the resulting color distribution matches a predefined histogram template.

For each pixel location in the background image, a multiple Gaussian mixture (MGM) model for the pixel color values is adopted. A novel weight updating scheme for the color clusters is used to address the sleeping person problem. Once every T frames, the number of color values that fall into each cluster is computed and stored in a counter variable. The corresponding cluster weight is updated once every T frames based on the counter value and the previous history of the cluster. The weight evaluation takes into consideration both the cluster duration and the recurrence frequency. The key idea behind the proposed approach is to use the cluster weight to approximate the cluster duration thus enabling one to decide whether or not to adapt a new cluster into the background model. A significant advantage of the proposed approach is that since the background updating is done once every T frames, it is computationally very efficient and very well suited for real-time traffic surveillance and monitoring applications.

4 Proposed Background Model

A background image that is larger than the actual image frame in the video stream is continuously maintained. Initially, the first image frame is aligned with the center of the background image. For each successive frame, a fast algorithm is used to estimate the Euclidean transformation parameters and thereby determine the alignment of the new image frame with the stored background image. The background image is updated in those areas that overlap with the new image frame.

4.1 Camera Motion Estimation

In the context of traffic monitoring and surveillance, cameras mounted on bridges or overpasses are typically subject to structural vibrations caused by especially heavy moving vehicles, resulting in sudden and random camera perturbations. Since the range of the perturbations is typically small, it is modeled as a simple Euclidean transformation given by $\hat{v}_x = x\cos(\theta) - y\sin(\theta) + v_{x_0}$ and $\hat{v}_y = x\sin(\theta) + y\cos(\theta) + v_{y_0}$, where $\hat{v}_x$ and $\hat{v}_y$ are the velocity components at point (x, y) along the x and y axes respectively. The center of the rotation is $(0, 0)$ and is chosen to be the center pixel in the image. In order to determine $(v_{x_0}, v_{y_0}, \theta)$, the velocity (v_x, v_y) at each point is computed using the pyramidal Lucas-Kanade optical flow technique [10]. Since θ is usually a small value, the Euclidean transformation can be rewritten as $\hat{v}_x = x - y\theta + v_{x_0}$ and $\hat{v}_y = x\theta + y + v_{y_0}$.

Let $E = \sum_{(x,y)}(v_x - x + y\theta - v_{x_0})^2 + (v_y - x\theta - y - v_{y_0})^2$. By letting $\partial E/\partial v_{x_0} = 0$, $\partial E/\partial v_{y_0} = 0$ and $\partial E/\partial\theta = 0$, the perturbation parameters can be solved as $v_{x_0} = \sum_{(x,y)} v_x/A$, $v_{y_0} = \sum_{(x,y)} v_y/A$ and $\theta = \sum_{(x,y)}(v_y x - v_x y)/\sum_{(x,y)}(x^2+y^2)$ where A is the area of the symmetric region R over which the computation is performed. Note that for the symmetric region R, $\sum_{(x,y)} x = 0$ and $\sum_{(x,y)} y = 0$.

4.2 Handling Sudden Illumination Changes

In the context of traffic surveillance and monitoring, sudden changes in ambient illumination are caused by moving cloud cover. For every image, its histogram is first normalized to reduce the effect of sudden ambient illumination changes. Let $h_t(g)$ be the histogram of an observed image at time t, where $g \in [0, G)$. Let $H_t(g) = \int_0^g h_t(x)dx$, where $H_t(g) \in [0, 1]$. Before normalization, parameters G_1 and G_2 are determined, such that $H_t(G_1) = \alpha$ and $H_t(G_2) = 1 - \alpha$. The normalization is a transformation given by the following equation:

$$H'(g) = \begin{cases} H_t(\frac{gG_1}{\alpha G}) & g < \alpha G \\ H_t(\frac{(g-\alpha G)(G_2-G_1)}{(1-2\alpha)G} + G_1) & \alpha G < g < \alpha_1 G \\ H_t(\frac{(g-\alpha_1 G)(G-G_2)}{(\alpha G)} + G_2) & g \geq \alpha_1 G \end{cases} \tag{1}$$

where $\alpha_1 = 1 - \alpha$. The histogram template $H_0^T(g)$ is initialized with the normalized histogram of the first image $H'_0(g)$. For a new observed normalized histogram $H'_{t+1}(g)$, the histogram template is adjusted as $H_{t+1}^T(g) = H_t^T(g) + \beta(H'_{t+1}(g) - H_t^T(g))$. The new image is then adjusted to match $H_{t+1}^T(g)$ by mapping each grayscale value g in the new image $H'_{t+1}(g)$ to g', such that $H'_{t+1}(g) = H_{t+1}^T(g')$.

4.3 Background Image Assumptions

Two key assumptions are made about the background. The first assumption is that the background color persists for a longer time duration than any foreground color at any given pixel location. This assumption is implicit in background updating schemes based on temporal averaging [3], median filtering [5] and multi-color models [2, 8]. Although this assumption holds over the long run, it might not be true over a short time duration, especially when the underlying scene is very busy. The second assumption is that the background color has a higher frequency of recurrence [1]. This assumption is true when moving objects pass over a pixel location periodically. Thus, it is logical to assign to each cluster center a weight that takes into account both, the time duration of the cluster and its recurrence frequency as an alternative to the simple weight updating scheme described in [8].

In the proposed background model, the two aforementioned assumptions about the background color are exploited to make the background updating procedure more robust. A multi-color model similar to one described in [8] is used. The background color at each pixel location is modeled using an ensemble of k color clusters. The weight assigned to each cluster incorporates both, its time duration and its recurrence frequency. The weighting scheme is designed to distinguish between color clusters that correspond to temporarily stationary objects and those that correspond to the actual background. Instead of using recent history to learn the cluster parameters, we use the first frame in the video sequence to initialize the model. Initially, each pixel is assigned a single cluster center whose mean is the pixel color value and variance is set to a default value σ_0^2. Two parameters N and R are used to characterize the two aforementioned properties of a cluster. N, the cluster cardinality, is the total number of observations that comprise (i.e., have matched) the cluster. N is a measure of the time duration of the cluster since the beginning of the video sequence. R is the total number of times that the cluster is matched and represents the recurrence frequency of the cluster. In order to compute N and R, we use tl to denote the last time instant that the cluster is matched and n to record the number of colors that have matched this cluster in recent history.

The graph in Figure 1 depicts an idealized plot of the observed gray level at a pixel location as a function of time and serves to explain the parameters tl, n, R and N. Figure 1 contains four time stamps: $t_1, \ldots, t_4$. Let C_i denote a cluster center that did not exist at $t = 0$ but is initialized at $t = t_1$. At $t = t_1$, the last time instant that this cluster is matched is $tl = t_1$, the number of colors that have matched this cluster in recent history $n = 1$ and the total number of colors that have matched this cluster $N = 1$. At $t = t_2$, tl is updated as $tl = t_2$ whereas n and N are both updated to reflect the total number of colors that have fallen into this cluster during the time interval $[t_1, t_2]$. At time $t = t_3$, the

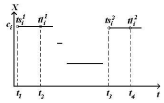

Figure 1: Cluster parameter analysis

cluster with center C_i is matched again. At time $t = t_4$, n is updated to account for the number of colors that have fallen into this cluster during the interval $[t_3, t_4]$ whereas N is updated to account for the total number of colors that have fallen into this cluster during the interval $[t_1, t_4]$ or $[0, t_4]$. Since the cluster with center C_i is matched again after $t = t_2$, $R = 2$ which means that there is greater evidence to support the hypothesis that the cluster with center C_i does represent the actual background. However, we do not explicitly measure R in our scheme. Instead, we modify the value of N to reflect the impact of R, and use N as the cluster weight. In contrast to Stauffer and Grimson's scheme [8], in the proposed scheme the weight assigned to each cluster is indicative of both, the cluster duration and the cluster recurrence frequency. Thus each cluster is characterized with the following parameters:

- C_i: Centroid or mean of the ith color/gray level cluster.
- σ_i^2: Variance of the ith color/gray level cluster.
- N_i: Total number of colors/gray levels that have matched the ith cluster. Initially, $N_i = 1$ for all clusters.
- tl_i: The most recent time that the ith cluster has been updated. Initially, $tl_i = 0$ for all clusters.
- n_i: The number of colors/gray levels that have matched the ith cluster in recent history.

The details of the background updating scheme are given in the following section.

4.4 Background Updating

Given color X_k at a certain pixel location in the current frame, we compare it to the existing cluster centroids associated with this pixel location. If $X_k \in [C_i - 2.5\sigma_i, C_i + 2.5\sigma_i]$ then X_k is deemed to match the ith cluster. The centroid and covariance of the ith cluster are updated as follows:

$$C_i = C_i + (X_k - C_i)/L \qquad (2)$$
$$\sigma_i^2 = \sigma_i^2 + ((X_k - C_i)^2 - \sigma_i^2)/L \qquad (3)$$

where L is an integer representing the inverse of the learning rate. The advantage of using an integer L instead of the learning rate α in equations (2) and (3), is that the need for floating point computation at each update is averted. For example, in equation (2) we can accumulate the difference $(X_k - C_i)$ and decrement C_i by 1 if $(X_k - C_i) < -L$ and increment C_i by 1 if $(X_k - C_i) > L$. If color X_k does not match an existing cluster then a new cluster is created replacing an existing cluster j with minimum weight N_j.

In order to efficiently compute the duration and recurrence frequency of a cluster, we quantize the time series $x(t)$ into time slices of interval T. For all clusters in a given time slice, if the number of colors that have been assigned to cluster i in that time slice is n_i then the duration of the cluster is updated as: $N_i = N_i + T$ if $n_i > T/2$, otherwise $N_i = N_i + n_i$. Thus, if a cluster at a given pixel location is assigned more than $T/2$ colors in a given time slice, then this cluster is deemed to dominate this time slice. Consequently, we reward this cluster by adding T to N_i else we update its duration by adding the actual number of matched colors n_i to N_i.

We also check for the recurrence frequency of clusters. If a cluster has not been matched for some period of time and then matched again, it is probable that the cluster does represent the real background. If this cluster has been deemed to be sufficiently exposed during the current time slice, i.e., $n_i \geq \delta$ then we increase its weight by increasing its value of N. On the other hand, if $n_i < \delta$ then the cluster is deemed insufficiently exposed and its recurrence frequency ignored during the current time slice. We measure the recurrence frequency of the ith cluster by checking the last time tl_i that the cluster was matched. If $t - tl_i > 2T$, then $N_i = N_i + T/2$, that is N_i is incremented by an extra duration $T/2$ to account for the recurrence and t_l is set to t. Checking for recurrence frequency is useful when the underlying dynamic scene is very busy.

The clusters at each pixel location are ranked on the basis of their N values, the higher the value of N, the higher the priority for the corresponding color/gray level of that cluster to be considered as part of the background. However, it is necessary to set an upper limit for the value of N_i since too large a value of N_i will make it difficult for an actual new background color cluster to be considered as part of the background. We set the upper limit of N_i to Δ. At any pixel location, if $N_{max} > 1.25\Delta$ where $N_{max} = \max_i(N_i)$, we scale down all the N_i values by multiplying them by $4/5$. If $N_i = 0$ then the ith cluster is deleted. All the clusters which satisfy the condition $N_i > N_{max}/3$ are deemed as representing valid background colors/gray levels. If a cluster has not been updated for a time period Δ, then it is deleted. The background updating is performed once every T frames.

The background updating algorithm is summarized as follows:

1. Given an observed color/gray level X_k at a pixel, check all of the pixel's clusters. If X_k matches cluster i, then update the centroid and the variance of cluster i using equations (2) and (3). Set $n_i = n_i + 1$.
2. If there is no match, create a new cluster and replace an existing cluster with the smallest N_i value. For the new cluster, set $C = X_k$, $N = 1$, $n = 1$, $\sigma^2 = \sigma_0^2$ and $tl = k$.
3. If $(t \mod T) \equiv 0$, then for each cluster i at each pixel,
   (a) Check the value of n_i and update N_i as : if $n_i > T/2$,

then $N_i = N_i + T$; otherwise, $N_i = N_i + n_i$.

(b) Check for recurrence: If $n_i > \delta$ and $t - tl_i > 2T$, then $N_i = N_i + T/2$ and $tl_i = t$.

(c) Reset all n_i values to zero.

(d) Check which clusters are deemed as belonging to the background. All clusters i such that $N_i > N_{max}/3$ where $N_{max} = \max_i(N_i)$ are considered to belong to the background.

(e) If $N_{max} > 1.25\Delta$, then $N_i = N_i * 4/5$, for all i.

(f) For any i, if $k - tl_i > \Delta$, then delete this cluster.

The background updating scheme described above was incorporated into a real-time traffic monitoring system and tested on color (RGB) and grayscale video sequences of real traffic scenes. All video sequences were sampled at a constant rate of 30 frames per second (fps). We chose values of $L = 1024$ and $k = 4$ in our implementation. Since the background is constantly refreshed, the value of N_i for the cluster corresponding to the actual background eventually increases to 1.25Δ. We chose $\Delta = 12000$. We also chose $T = 2$ seconds (60 frames) and $\delta = 20$. Thus, if a cluster that has not been updated for the past $2T = 120$ frames, has received more than 20 updates in the current time slice, then it is treated as an instance of a recurring background color/gray level and the N_i value of the cluster is incremented by $T/2$.

5 Experimental Results

To simulate camera perturbation, the video streams were gathered while the tripod mount of the camera was being manually shaken. A sudden global illumination change is simulated in the laboratory by manually switching on and off half of the lights. Figure 2(a) shows the background image from the moving camera. The actual size of the image frame is 360×240 whereas the background image size is 400×280. Figure 2(b) shows the image after a sudden change in ambient illumination resulting from half the lights in the laboratory being switched off. Figure 2(c) shows the image after histogram template adjustment has been performed. Experimental results on the captured videos show that the proposed scheme for camera motion compensation and handling sudden global illumination change works well and that the background updating scheme is capable of recovering from abrupt camera motion and sudden global illumination change.

A comparison between the proposed background updating scheme and that of Stauffer and Grimson [8] is performed using video streams, captured by a static camera, of a busy traffic scene containing several vehicles and with traffic lights present. Figures 3(a), and 3(b) summarize the results of our scheme for an RGB color video sequence.

(a) Camera Perturbation (b) Sudden Illumination Change (c) Histogram Adjustment Result

Figure 2: Handling camera perturbation and illumination change.

(a) Frame 3841. (b) Background. (c) Binarized foreground.

Figure 3: Results of the proposed technique.

Figure 3(a) shows the color image frame at time $t = 3841$ where a *sleeping person* problem is evident at pixel location $(180, 134)$. Figure 3(b) depicts the background image generated by our scheme from the color associated with the maximum weight cluster centroid at each pixel location at time $t = 3841$.

For the sake of comparison we also implemented the background updating scheme of Stauffer and Grimson [8]. As previous analysis has shown, with the same learning rate of $1/1024$ as the proposed scheme, Stauffer and Grimson's scheme suffers periodically from the *sleeping person* problem. Videos of the background image frames generated with the proposed scheme and Stauffer and Grimson's scheme are provided along with this paper for visual comparison. From these videos it can be seen that, the proposed scheme suffers from the sleeping person problem initially when little information about the background is known. However, the proposed scheme is observed to eventually overcome this problem and converge to a stable background image. However, Stauffer and Grimson's scheme is observed to suffer from the sleeping person problem from time to time. Note that with a learning rate of $1/1024$, Stauffer and Grimson's scheme can adapt quickly to ambient illumination changes caused by the moving cloud cover. However, lowering the learning rate to address the sleeping person problem is not practically feasible in the case of Stauffer and Grimson's scheme, since it renders the scheme less adaptive to changes in ambient illumination.

The results of moving object extraction using the background image generated by our scheme are compared with those obtained using the background image generated by the scheme of Stauffer and Grimson [8]. Figure 3(c) depicts the result of moving object extraction using the background image generated by our scheme where the dark pixels repre-

(a) Background.

(b) Binarized foreground.

Figure 4: Results using Stauffer and Grimson's method.

sent the pixels with motion. For the purpose of comparison, the result of moving object extraction using the background image generated by the scheme proposed by Stauffer and Grimson [8] is depicted in Figure 4(b). It is evident that in the case of Stauffer and Grimson's scheme, the objects that become temporarily motionless are partially merged into the background and cannot be extracted. However, with our background updating scheme, moving vehicles that are temporarily static can still be extracted. Real videos confirming these results are also provided with this paper.

6 Conclusions

In this paper we proposed a background updating scheme for a real-time traffic monitoring system. Specially, we have addressed the camera perturbation problem, sudden or gradual ambient illumination change problem and the *sleeping person* problem. To make the background updating scheme adaptive to gradual changes in the background, we used a multi-color model where multiple color clusters are associated with each pixel location in the background image. The cluster parameters were updated periodically to adapt to gradual changes in the background. To make the proposed background updating scheme robust in the face of the *sleeping person* problem, the color clusters at each pixel location were assigned weights based on the observation that the clusters corresponding to the real background colors are likely to persist for a longer time duration and also have a higher recurrence frequency compared to the clusters that correspond to colors from temporarily motionless objects. The camera perturbation was modeled as an Euclidean transformation and the camera perturbation parameters determined using the pyramidal Lucas-Kanade optical flow technique. Sudden ambient illumination changes are resolved using an adaptive histogram template, which corrects the skewed image color/grayscale histogram resulting from sudden illumination change. Experimental results on grayscale and color video sequences obtained from real traffic scenes show that the proposed background updating scheme can adapt to gradual or long-term changes in the background while ignoring short-term changes arising from the sleeping person problem and sudden ambient illumination changes. The proposed scheme is also computationally efficient since most of the computation is performed once every 60 frames. Moreover, the background update equations are optimized by greatly reducing floating point computation, thus making the background updating scheme well suited for real-time traffic surveillance and monitoring. Although the proposed scheme is specifically designed for a real-time traffic monitoring system, it is nevertheless applicable to most surveillance systems, in which *sleeping person* problem is seen to occur but the time period for which a moving object is temporarily stationary has a definite upper bound.

References

[1] D. Butler, S. Sridharan and V.M. Bove, Jr., Real-time Adaptive Background Segmentation, *Proc. IEEE ICME*, Baltimore, MD, July 2003.

[2] P. KaewTraKulPong and R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, *Proc. Wkshp. Adv. Vision-based Surveillance Sys.*, Kingston, UK, Sept. 2001.

[3] S. Kamijo, Traffic Monitoring and Accident Detection at Intersections, *IEEE Trans. Intell. Transport. Sys.*, Vol. 1, No. 2, June 2000, pp. 108-118.

[4] D. Koller, J.W. Weber and J. Malik, Robust Multiple Car Tracking with Occlusion Reasoning. *Proc. ECCV*, Stockholm, Sweden, 1994, pp. 189-196.

[5] M. Massey and W. Bender, Salient stills: Process and practice, *IBM Sys. Jour.*, 1996, Vol. 35, Nos. 3&4, pp. 557-573.

[6] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfield, H. Wechsler, Tracking Groups of People, *CVIU*, Vol. 80, 2000, pp. 42-56.

[7] C. Ridder, O. Munkelt, and H. Kirchner, Adaptive Background Estimation and Foreground Detection Using Kalman Filtering, *Proc. of Intl. Conf. Recent Adv. Mechatronics*, Istanbul, Turkey, 1995, pp. 193-199.

[8] C. Stauffer and W.E.L. Grimson, Adaptive Background Mixture Models for Real-time Tracking. *Proc. IEEE Conf. CVPR*, Ft. Collins, CO, June 1999, pp. 246-252.

[9] K. Tooyama, J. Krumm, B. Brumit, and B. Meyers, Wallflower: Principles and Practice of Background Maintenance. *Proc. ICCV*, Corfu, Greece, Sept. 1999, pp. 255-261.

[10] J. Bouguet, Pyramidal Implementation of the Lucas-Kanade Feature Tracker: Description of the Algorithm, *OpenCV Documentation*, http://www.intel.com/research/mrl/research/opencv/