Robust and Efficient Foreground Analysis in Complex Surveillance Videos

YingLi Tian[1], Andrew Senior[2], and Max Lu[3]

[1]Electrical Engineering Department
The City College and Graduate Center
The City University of New York, New York, NY 10031 USA
ytian@ccny.cuny.edu

[2]Google Research
76 Ninth Ave, New York, NY 10011 USA
andrewsenior@google.com

[3]IBM Global Technology Services,
19 Skyline Dr., Hawthorne, NY 10532 USA
maxlu@us.ibm.com

Abstract

Mixture of Gaussians based background subtraction (BGS) has been widely used for detecting moving objects in surveillance videos. It is very efficient and can update the background model with slow lighting changes, however, it suffers from a number of limitations in complex surveillance conditions such as quick lighting variations, heavy occlusion, foreground fragments, slow moving or stopped object etc. To address these issues, this paper first focuses on foreground analysis within the Mixture of Gaussians BGS framework in long term scene monitoring to handle 1) quick lighting changes, 2) static objects, 3) foreground fragments, 4) abandoned and removed objects, and 5) camera view changes. Then, we propose a framework with interactive mechanisms between BGS and processing from different high levels (i.e. region, frame, and tracking) to improve the accuracy of moving object detection and tracking to handle 1) objects that stop for a significant period of time and 2) slow-moving objects. The robustness and efficiency of the proposed mechanism is tested in IBM Smart Surveillance Solution on a variety of sequences, including standard datasets. The proposed method is very efficient and handles 10 video streams in real-time on a 2GB Pentium IV machine with MMX optimization.

Keywords: *Background subtraction (BGS), foreground analysis, interaction of BGS and tracking, video surveillance.*

I. INTRODUCTION

Automatic video surveillance is a rapidly expanding field, driven by increases in the affordability of technology and the perceived need for security. Demand and the constrained domain make it one of the most commercially viable application areas for computer vision technology. Many applications in the field require the tracking of moving objects (usually people and vehicles), so that events (such as

entering a secure zone) can be detected or those objects can be found through a search interface.

Video surveillance systems which run 24 hours a day and seven days a week create a large amount of data including videos, extracted features, alerts, etc. There are thousands of surveillance cameras in a typical city. For example, London has about 500,000 security cameras and Manhattan has more than 10,000 cameras [45, 46]. These numbers are continuing to increase. Robustness and efficiency are the two key factors for successful video surveillance systems due to the large scale data processing and complex video analysis.

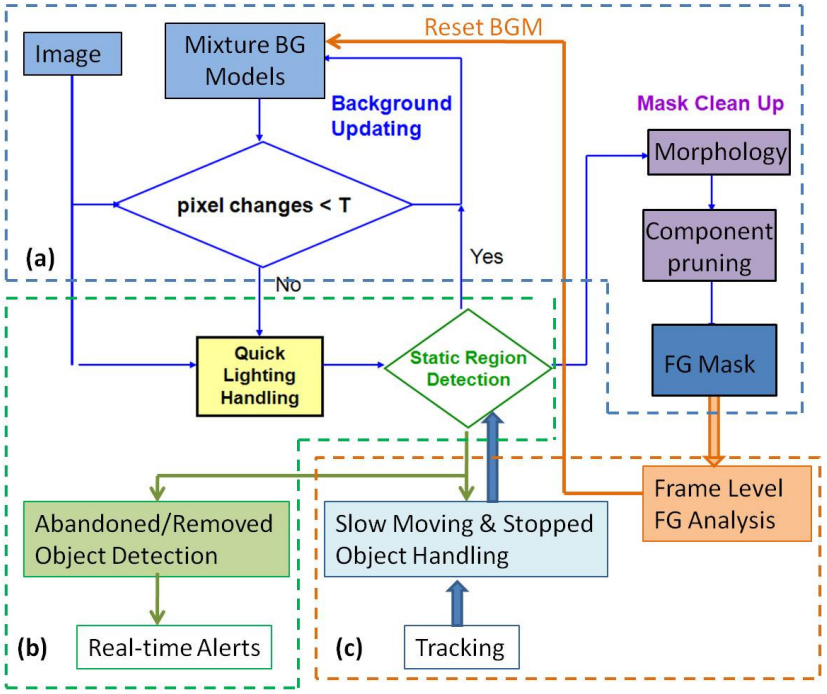

Fig. 1: Overview of the background subtraction and foreground analysis system: (a) mixture of Gaussians based background subtraction; (b) region level foreground analysis to handle quick lighting change, detect static regions, reduce foreground fragments, and detect object type (abandoned or removed); (c) higher level foreground analysis to handle camera move (frame level) and slow moving/ stopped object (tracking level).

In most automatic surveillance systems, objects of interest are first detected, usually by background subtraction which will find moving objects. Detected objects are then tracked by a tracking module. Such a system provides an efficient mechanism for detecting moving objects, but practical implementations suffer from a number of limitations in complex surveillance video conditions such as quick lighting variations, severe weather, heavy occlusion, crowding, non-rigid objects, etc. A slow moving or stopped object can result in the object being adapted piecemeal into the background. This leads to errors in tracking, as the object dissolves into multiple fragments, and false "ghost" fragments appear where the background contains the object after it moves away. In this paper, we focus on problems of background subtraction and foreground analysis in long term scene monitoring to handle quick lighting changes, static objects, foreground fragments, abandoned and removed objects, slow-moving objects, and objects that stop for significant periods of time. In general, quick lighting changes will cause false foregrounds.

Figure 1 shows an overview of our system. The system includes three main components: (a) mixture of Gaussians based background subtraction; (b) region level foreground analysis to handle quick lighting change, detect static regions, reduce foreground fragments, and detect object type (abandoned or removed); and (c) higher level foreground analysis to handle global scene variations such as a camera view change (frame level) and slow moving/ stopped object (tracking level). Overall, the work introduced in this paper offers the following main contributions to robustly and efficiently analyze foreground in complex videos for surveillance applications:

- We propose a new framework to analyze the foreground as *moving objects*, *abandoned objects*, or *removed objects (ghosts)* while detecting the background by employing a mixture of Gaussians method [25] as the basic framework.
- We integrate the Phong shading model [20] into the framework to robustly handle quick lighting changes.
- We design a new mechanism by considering interactions between BGS and processing at several higher levels (i.e. region, frame, and tracking) to improve the accuracy of moving object detection and tracking. Region level information is employed for foreground analysis, to handle quick lighting changes, detect static regions, reduce foreground fragments, and detect object type (abandoned or removed). For camera movement or global scene variation (lights are turned off), frame level information will be used to reset the BGS models. To handle slow moving/stopped objects, we create two feedback mechanisms that allow interactions between BGS and tracking to handle stopped and slow-moving objects.

This paper is organized as follows. The next section describes previous work in BGS to handle quick lighting changes, fragments, slow moving or stopped objects, and previous systems that use feedback to assist in background subtraction. Section 3 briefly summarizes the adaptive BGS method based on mixture of Gaussian models. Section 4 introduces how our system handles quick lighting changes, static region detection, fragment reduction, and abandoned/removed object classification. Section 5 describes the interaction between BGS and tracking to deal with slow moving and stopped objects respectively. Section 6 describes experiments to evaluate the robustness of the proposed method and to assess the impact of the feedback mechanisms on tracking performance and presents the results. Section 7 presents conclusions and discussion.

II. RELATED WORK

Background subtraction (BGS) is a conventional and effective approach to detect moving objects for video surveillance systems with stationary cameras. To detect moving objects in a dynamic scene, many adaptive background subtraction techniques have been developed [5-19, 22, 24-27, 29, 31, 35-41]. Stauffer and Grimson [25] modeled each pixel as a mixture of Gaussians and used an on-line approximation to update the model. Their system can deal with slow lighting changes and introducing or removing objects from the scene. Monnet *et al.* [19] proposed a prediction-based online method for the modeling of dynamic scenes.

Their approach has been tested on a coast line with ocean waves and a scene with swaying trees. However, they need hundreds of images without moving objects to learn the background model, and moving objects cannot be detected if they move in the same direction as the ocean waves. Mittal and Paragios [18] presented motion-based background subtraction by using adaptive kernel density estimation. In their method, optical flow is computed and utilized as a feature in a higher dimensional space. They successfully handled complex backgrounds but the computational cost is relatively high. Some hybrid change detectors have been developed which combine temporal difference imaging and adaptive background estimation to detect regions of change [6, 13]. Huwer *et al.* [13] proposed a method of combining a temporal difference method with an adaptive background model subtraction scheme to deal with lighting changes. However, none of these methods can adapt to quick image variations such as a light turning on or off. Li *et al.* [16] proposed a Bayesian framework that incorporates spectral, spatial, and temporal features to characterize the background appearance at each pixel. Their method can handle both the static and dynamic backgrounds and good performance was obtained on image sequences containing targets of interest in a variety of environments, e.g., offices, public buildings, subway stations, campuses, parking lots, airports, and sidewalks. Recently, some methods have been developed which integrate discriminative features with tracking for object tracking and scene segmentation [12, 39-41] and a few researchers start to work on developing background subtraction methods from moving cameras [11, 27].

The mixture of Gaussians BGS method is becoming popular in recent years because of its robustness and efficiency. However it cannot adapt to quick lighting changes. A number of techniques have been developed to handle quick lighting changes or to improve the performance of the mixture of Gaussians method [8-10, 14-15, 26, 29].

Although many researchers have examined background subtraction, few papers can be found in the literatures for foreground analysis [5, 7]. Cucchiara *et al.* [7] analyzed the foreground as moving object, shadow, and ghost by using the optical flow based motion information. The computational cost is relatively expensive for real-time video surveillance systems because of the computation of optical flow.

In many video surveillance systems, object detection is followed by object tracking [2, 5] to associate the detections across time and describe the behavior of objects. Most of these systems operate in a feed-forward manner to pass detections from background subtraction to the tracker and then tracks are stored or processed further, for instance by behavior analysis modules. Many object tracking techniques focus on handling occlusions but neglect how to track slow moving or stopped objects for long term scene monitoring. Boult *et al.* [2] describe a system that performs well at detecting slow moving objects.

There have been a few systems that have investigated the possibility of feedback from tracking to background subtraction. Some papers [14, 29, 32] introduced feedback from the frame level and some papers employed feedback from tracking [1, 4, 10, 20, 36]. Abbott *et al.* [1] proposed a method to reduce computational cost in visual tracking systems by using track state estimates to direct and constrain image segmentation via background subtraction and connected components analysis. Harville [10] used application-specific high level feedback (frame level, person detector and tracker, and non-person detector)

frame to locally adjust sensitivity to background variation. Senior [23] suggests recalculating the background/foreground segmentation using the model of the tracked object after the background subtraction stage. Wang *et al.* [34] proposed a unified framework to address detection and tracking simultaneously to improve the detection results. They feed the tracking results back to the detection stage.

The interaction between the tracking and background subtraction can also be used to improve the tracking of slow moving and stopped objects. Venetianer *et al.* [32] examine a way of pushing foreground objects into the background and vice versa. Yao and Odobez [37] use a similar layered background mechanism to remember stopped objects. Taycher *et al.* [28] proposed an approach that incorporates background modeling and object tracking to prevent stationary objects fading into the background. Pnevmatikakis and Polymenakos [21] overcame the problem of stationary targets fading into the background by combining BGS and a Kalman tracker in a feedback configuration.

III. ADAPTIVE BACKGROUND MIXTURE MODELS

Stauffer and Grimson [25] first introduced a mixture Gaussians for BGS by modeling the background model as *K* Gaussian mixtures. For each pixel, the mixture weights at time *t* are updated based the weights at time *t-1* as:

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha(M_{k,t}). \qquad (1)$$

where α is the learning rate. For every new pixel value, *Xt*, is checked against the existing *K* Gaussian distributions, until a match is found. Here, a match is defined as a pixel value within 2.5 standard deviations of a distribution. $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. Assuming the red, green, and blue pixel values are independent and have the same variances, we write: $\sum_{k,t} = \sigma_k^2 I$. After the Gaussians are ranked in descending order of ω/σ, the first *B* distributions are chosen as the background model, where

$$B = \arg\min_b \left(\sum_{k=1}^{b} \omega_k > T \right), \qquad (2)$$

and *T* is the minimum fraction of the data that should be accounted for by the background. Here, σ remains same for unmatched distributions. In implementation, two significant parameters, α and *T,* need to be set. For more details, see Stauffer and Grimson [25]. In our system, we set *K = 3* (three Gaussians), α=0.005, and *T* = 0.4. We implement the method for both grayscale and RGB video inputs. All the test results in our system are from the same set of parameters.

The original mixture of Gaussians method is robust to slow lighting changes, periodic motions from a cluttered background, and camera noise. However it cannot handle: 1) quick lighting changes; 2) detect static regions; 3) fragments; 4) classify abandoned or removed objects; 5) camera view changes; 6) objects that stop for a significant period of time; and 7) slow-moving objects. We describe some solutions for these problems in the following sections.

IV. FOREGROUND ANALYSIS

A. Integrating Phong shading model to handle quick lighting changes

To handle quick light changes, researchers have developed different methods [3, 5, 26, 29, 35, 36]. Watanabe *et al.* [35] proposed a pre-processing step for background subtraction by applying an interesting radiometric model based on knowing the position of the sun and building models to remove strong shadows of buildings due to direct light from the sun, prior to applying a change detection algorithm. Javed *et al.* [14] proposed a hierarchical approach by combining color and gradient information with the mixture of Gaussians BGS method to handle quick lighting changes. Tian *et al.* [29] proposed a method to handle quick lighting changes by integrating texture information into the BGS. The basic idea is that the texture in false positive foreground areas caused by lighting changes should be similar to the texture in the background. Xie *et al.* [36] observed that the sign of the difference between corresponding pixel measurements in the current image and the background image is invariant during quick lighting changes. They developed a BGS method based on this observation which is able to discriminate false position foregrounds caused by lighting changes from the real moving objects. The algorithm of Durucan and Ebrahimi [8] detects foreground changes with respect to sudden illumination variations by voting a pixel as changed or unchanged in a given spatial window centered at the pixel based on a linear parametric model. Stefano *et al.* [26] proposed a visual correspondence measure together with a tonal registration procedure to handle quick lighting changes.

According to the Phong shading reflection model [20], for an image of a scene with Lambertian surfaces, the image intensity of a pixel $I(x, y)$ at an object is modeled as the product of the illumination from light source(s) $I_l(x, y)$ and the reflectance of the object surface $I_o(x, y)$:

$$I(x, y) = I_l(x, y) I_o(x, y) \tag{3}$$

Here, we integrate the Phong shading model into the mixture of Gaussians BGS method. Assuming $I_F(x, y)$ is the intensity of pixel (x, y) in the current frame of the input video for foreground analysis and $I_B(x, y)$ is the intensity of the same pixel (x, y) of the background model which is built based on previous frames, we have:

$$\begin{aligned} I_F(x, y) &= I_{lF}(x, y) I_{oF}(x, y) \\ I_B(x, y) &= I_{lB}(x, y) I_{oB}(x, y) \end{aligned} \tag{4}$$

where $I_{lF}(x, y)$ and $I_{lB}(x, y)$ are the illumination component from light source(s) in the current frame and the background image. $I_{oF}(x, y)$ and $I_{oB}(x, y)$ are the reflection component from object(s) in the current frame and the background image.

Since only the reflectance component from objects $I_o(x, y)$ contains information about the objects in the scene, for the pixels which do not belong to real moving objects (i.e. which belong to the background), the object reflection component from the current image $I_{oF}(x, y)$ and the background image $I_{oB}(x, y)$ should be equal. Therefore, for the false foreground pixels which are caused by

quick lighting changes, the ratio of intensity between the current image and the background image will become:

$$\frac{I_F(x,y)}{I_B(x,y)} = \frac{I_{lF}(x,y)I_{oF}(x,y)}{I_{lB}(x,y)I_{oB}(x,y)} = \frac{I_{lF}(x,y)}{I_{lB}(x,y)} \quad (5)$$

Hence, for the detected foreground pixels, if the ratio of intensity $R(x,y) = I_F(x,y)/I_B(x,y)$ in a small window *W(x, y)*, centered at pixel (x, y) remains constant (i.e. the difference of the intensity ratio is less than a threshold), these foreground pixels are eliminated as false foreground pixels caused by lighting changes. The difference of the ratio in the window *W(x, y)* is calculated by:

$$R_{Diff} = \frac{1}{N} \sum_{i \in W(x,y)}^{N} (R(x,y) - \mu(x,y))^2 \quad (6)$$

where N is the total number of pixels in the window W(x, y) and the mean of the intensity ratio $\mu(x, y)$ is obtained by:

$$\mu(x,y) = \frac{1}{N} \sum_{i \in W(x,y)}^{N} R(x,y) \quad (7)$$

In our implementation, we set the window size to *3x3* and the threshold on the difference in intensity ratio to 0.05. A post-processing is performed to fill holes in the foreground by applying morphological operations. As shown in Figure 2, the difference of the intensity ratios in real foreground regions are much bigger than that in the false foreground regions caused by lighting changes.

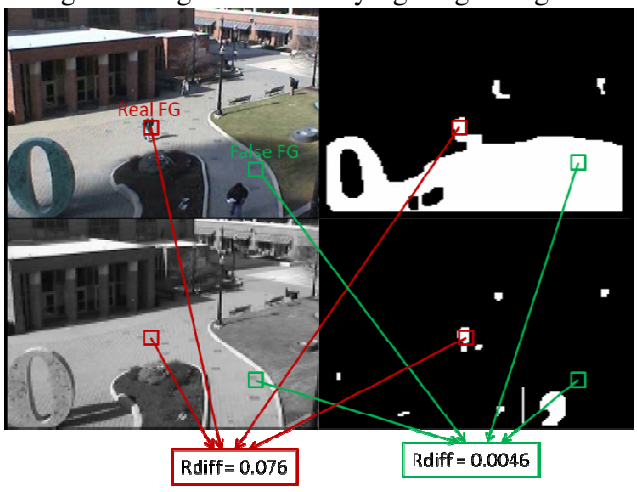

Fig. 2: Removing false foreground regions caused by quick lighting changes by integrating the Phong shading model. The difference of the intensity ratios of the current frame and the background image are illustrated for two foreground regions, one real (red arrows) and one false (green arrows). **Upper-left**: current frame; **Bottom-left**: background image; **Upper-right**: foreground image before integrating Phone shading model. **Bottom-right**: foreground image after rejecting the false positives by using Phong shading model.

B. Static Region Detection

If an object in a scene becomes stationary or an object starts moving, the pixels belonged to the stopped object will be adaptively updated to the mixture of Gaussians of the background model. Ideally, each object should correspond to one

connected component in the foreground image. However, due to the uneven intensity distribution of the object and the background, different pixels will be updated to the background model at different times to produce many connected components for one object. In this paper, we define areas corresponding to objects which change state either from stationary to moving or from moving to stationary as "static regions."

a) Static region is detected based on the 2nd Gaussian component.

b) The static region is pushed back to the 1st Gaussian component.

Fig. 3: Static region (the chair inside the box) detection. (a) The static region mask is visualized on the image (top left). The 1st Gaussian component (top right), the 2nd Gaussian component (bottom left), and the 3rd Gaussian component (bottom right) of the background model are shown respectively. (b) After pushing the static region (the chair) to the background image (top right) from the 2nd component (bottom left) when the size of the static region is biggest.

Here, we discuss how to detect static regions (top left in Figure 3(a)) by using the same framework of Gaussians mixtures of BGS. Figure 3(a) shows an example of a detected static object and three Gaussian mixtures of the background model. Generally, the 1st Gaussian distribution (top right in Figure 3(a)) models

the persistent pixels and represents the background image. The repetitive variations and the relatively static regions are updated to the 2nd Gaussian distribution (see bottom left in Figure 3(a)). The chair is updated to the 2nd Gaussian distribution after it stays stationary. The 3rd Gaussian distribution (bottom right in Figure 3(a)) represents the pixels with quick changes (e.g. foreground objects). In our system, we use 3 Gaussian distributions in the background model. The 2nd Gaussian distribution is used to detect if a pixel belongs the static region:

$$pixel \in static \quad region, \quad if\ \omega_2 > T. \tag{8}$$

where T is the same threshold as in Equation (2).

C. Foreground Fragment Reduction

Foreground fragments are a common problem for many background subtraction methods. In the mixture of Gaussians BGS method, the different parts of a static region are often updated to the background model at different speeds based on the similarity of the pixel intensities between the static region and the background model. Hence many foreground fragments are caused by static regions.

As we discussed in above subsection of static region detection, the 1st Gaussian distribution models the persistent pixels and represents the background image. The repetitive variations and the relatively static regions are updated to the 2nd Gaussian distribution. By "pushing back" the static region to the background model when the static region is biggest, namely switching the weights of the 1st and 2nd Gaussian distributions for the static region pixels. Therefore, we can avoid fragmentation of the foreground. Here, the process of "pushing back" will heal the static region to the first Gaussian distribution of the background models at this moment. In our implementation, we compare the area of the static region between the current frame and the previous frame. Once the area begins to shrink, the static region has reached its biggest size and needs to be pushed back to the background model. In the pushing back process, we reset the weight of all the pixels in the static region to the maximum weight which was defined in the program. The mean and variance of the 2nd Gaussian distribution are exchanged with those of the 1st Gaussian distribution for each pixel in the static region mask. Figure 3(b) shows that the static region detected in Figure 3(a) has been pushed back to the background image (top right in Figure 3(b)). Notice that the region corresponding to the static region in the 2nd distribution (bottom left in Figure 3(b)) has been exchanged with the region in the background image (top right in Figure 3(b)).

D. Abandoned and Removed Objects Classification

Detecting abandoned and removed objects is very important for video surveillance and security. After static regions are detected and healed (i.e., pushed back into the first Gaussian distribution of the background models), we need to classify whether the healing corresponds to an abandoned or removed object event. In our system, we developed two methods to detect the type of the static regions as abandoned or removed (ghost) objects: an edge energy-based method [5] and a region growing-based method [31].

For the edge energy-based method, we analyze the change in the amount of edge energy associated with the boundaries of the static foreground region

between the current frame of the original image and the background image. The intuition is that, in many cases, covering the background with an object will introduce more edges in the background image along with the object boundaries. The static region is an abandoned object if there are significantly more edges in the original image. Conversely, the static region is a removed object if there are more edges in the background image. More details can be found in paper [5].

For the region growing-based method, we first erode the static foreground region to make sure its boundaries fall completely inside the object. Then, these boundary points are employed as seeds to perform a region growing toward outside based on the similarity of intensities. The region growing process stops at the boundaries of the object, leading to a smaller segmented region which is not compatible with its surroundings. The same segmentation process is then applied in the background image. In this case, we can see that the resulting segmented. The heal type is finally determined by just comparing the size of the two segmented regions. If the background segmentation is larger than the current frame segmentation, the foreground region is classified as abandoned object. Otherwise, it is classified as a removed item. If the segmented regions have similar sizes, the heal type is set to "unclear", which may occur when the static foreground blob corresponds to lighting changes or other artifacts. More details can be found in paper [5].

Compared with the edge energy-based method, the region growing-based method is more robust but slightly slower.

E. Frame Level Foreground Analysis for Camera Move/Blind

In our system, frame level analysis is useful for two situations: camera move/blind and large area quick lighting changes (e.g., turn on or turn off lights). If the camera was moved or blinded (i.e. covered or disconnected), the size of the bounding box of the foreground region is very close to the size of the whole image (> 95% of the area). If there are large area quick lighting changes, the foreground area will be a large part of the whole image (> 70% of the area). In both cases, the background will be reset to the current frame.

V. INTERACTION BETWEEN BGS AND TRACKING

Background subtraction algorithms are generally designed to be adaptive to be able to deal with scene changes (changing lighting; backgrounds whose appearance changes, such as trees and water; static objects). In practice repeated observation of similar values of a pixel end up with such values being considered as background and dissimilar values treated as foreground. However, a slow moving, or stopped, object can lead to just such repeated observations, and result in the object being adapted piecemeal into the background. This leads to errors in tracking, as the object dissolves into multiple fragments, and false "ghost" fragments appear where the background contains the object after it moves away.

However, the tracking process usually treats groups of pixels collectively, as unitary objects, and this higher-level information derived by the tracker can be used to inform the process of background subtraction. The tracker explicitly models the objects, whose behaviors are subject to physical constraints (such as rigid motion) in ways different to the physical constraints that control the appearance of individual pixels. In this section, we introduce how our system handles slow moving and stopped objects respectively. The methods of the

previous sections can be applied without any tracking being carried out, when high speeds are needed on video for which BGS-based applications (such as abandoned object) are sufficient and tracking incurs an unnecessary burden (our system runs at 200fps with BGS alone, 70fps with tracking enabled). This section describes further improvements that can be obtained by incorporating information from tracking.

Our approach is most closely related to that of Pnevmatikakis and Polymenakos [21], who to overcome the problem of stationary targets fading into the background, propose a system combining a mixture of Gaussians background subtraction algorithm and a Kalman tracker in a feedback configuration. They control the learning parameters of the background adaptation on a pixel level in elliptical regions around the targets, based on the tracking states from the Kalman tracker. A smaller learning parameter was used for slow moving objects. However, this mechanism will fail when the targets stay stationary for a long period. They will gradually fade into the background even with very small learning parameters.

In contrast, we create two feedback mechanisms that allow the tracker to suppress background updating for slow moving objects that are being tracked. Further, we introduce an active, tracker-driven, object-level healing process where whole objects are pushed to the background to solve the challenges in tracking caused by the stopped objects. For both slow objects and stopped objects, we do not change the adaptation rate of the background model update.

A. Feedback Mechanism for Interactions between BGS and Tracking

In order to improving tracking accuracy, we create a feedback mechanism that allows interactions between BGS and tracking. Figure 4 shows the diagram of the interaction between BGS and tracking together with the metadata messages passed between the modules. The feedback required to handle slow and stopped objects is implemented by adding information to metadata of tracking observations which are accessible to the BGS processing through following three requests: 1) “heal request”—tracking requests BGS to push the region back to background model; 2) “unheal request” ”—tracking requests BGS to convert the background model of a healed region back to that before the heal happened; and 3) “hold in foreground”—tracking requests BGS to hold a region without updating.

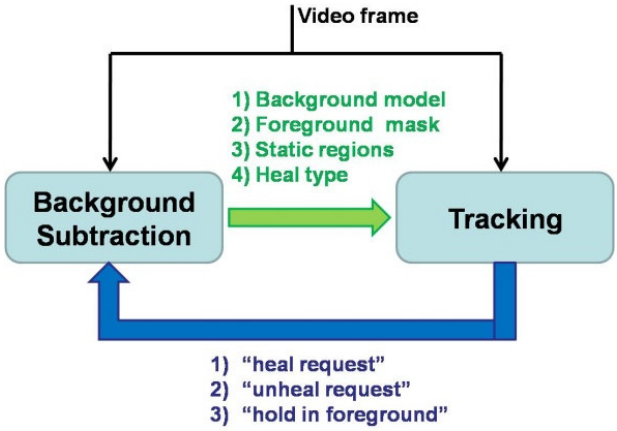

Fig. 4: Diagram of the interaction of background subtraction and tracking, showing the passing of metadata messages.

B. BGS Adaption Suppression for Tracking Slow Moving Objects

Slow Moving Objects Tracking Problem: Slow moving objects can present a significant problem to many background subtraction algorithms. In the algorithm of Stauffer and Grimson [25], each pixel is modeled by a mixture of Gaussians distribution. When an object moves very slowly or stops, any pixel on the object will eventually be updated to background model. If multiple pixels are affected in the same way, parts of the object will progressively be "lost". In Section IV-C, we partially address this problem by pushing the whole static region into the background model. However the problem is that the detection may come only after some pixels have already been adapted into the background, and may only affect part of the object. Thus, while the switch to background is no-longer independent for each pixel, it may still occur in several fragments, and results in part of an object being background and part being foreground.

BGS Adaption Suppression: To deal with this situation, the tracker suppresses background updating for slow-moving objects by feedback. When the tracker detects a slow-moving object based on speed, it flags the object observations as "slow moving" and the background subtraction algorithm suppresses the adaptation in the region where the slow moving object is observed (as indicated by a mask passed in the metadata).

Typically adaptation will already have been carried out by the background subtraction (as the video frame was received). Therefore, adaptation is suppressed in the region of a slow-moving object by copying pixels from the object model saved before adaptation, or by carrying out the inverse operation on those pixels (for instance decreasing the observation counts).

Suppressing adaptation in this way has the effect of maintaining the tracked object in the foreground, and uses object-level information from the tracker — that the pixels belong to a known object that is moving slowly and has been reliably detected and tracked for some period — to which the background subtraction module by itself does not have access.

A drawback of this mechanism is that it inhibits the process by which false alarm foreground objects are removed. For instance a shadow or a reflection which appears but is tracked for a while, would ordinarily quickly be forgotten as the background model adapts, but, if the "hold in foreground" method engages then these objects can be preserved indefinitely. However, the following mechanism can prevent this from happening.

C. Tracking-based BGS Healing for Stopped Objects

Stopped Objects Tracking Problem: Stopped objects lead to a different problem, and a dilemma for the design of a tracking system. Background modeling needs to adapt to changes in order to ignore "irrelevant" changes such as lighting changes. In a simple adaptive background subtraction system, when an object stops, as with slow moving objects above, then it will become part of the background and cease to be tracked. However the object is still present in the scene, and for some purposes (for instance the query "show me all cars present at 3p.m.") the system needs to explicitly represent that presence. A further problem is the fragment problem as described in Section IV-C. Since traditional

background subtraction algorithms typically operate independently on each pixel, different pixels of the object will be declared background at different times and result in a progressive fragmentation as the object is incorporated into the background.

When a static object starts moving, the background subtraction algorithm detects difference regions around the edges of an object, and as the original background is revealed, those pixels are detected as "foreground regions" and a "ghost" of revealed background is detected as foreground along with the true moving object, as shown in Figure 5. Toyama *et al.* describe this as the "waking person" problem, and conclude that it is not solvable in a self contained background subtraction module [32]. This presents several challenges to a tracking algorithm: (1) the object appears as many small foreground fragments; (2) the growing object is made up of a moving component and a static region; (3) the true object eventually separates from the static "ghost" region.

Fig. 5: Selected frames demonstrate ghosting by using mixture of Gaussians BGS. The car starts in the background and moves forward, leading to multiple foreground fragments and ultimately a large "ghost" or "hole" where it had been covering up the "true" background.

Tracking-based BGS Healing: With the adaptation-inhibition described in Section V-B, slow moving and stationary objects are not adapted into the background at all, so healing and fragmentation are no longer a problem. However static objects will now be held indefinitely in the foreground. As a parking lot fills slowly with cars, the number of "tracked" objects increases and their interactions and mutual occlusions become progressively more complex and unmanageable.

Consequently, we introduce an active, tracker-driven, object-level healing process where whole objects are pushed to the background. In this process, the tracker tracks whole objects and monitors their movement. When an object is stationary for a sufficient period (dependent on the scene context, for example dependent on the amount of activity in the scene and typical behaviors — whether objects stop for long or short periods) then the tracker determines that the object can be pushed to the background. The tracker sends a "heal request" message to the background subtraction processing, including a foreground mask indicating which pixels belong to the object.

On receiving the "heal request" message, the BGS algorithm takes the selected pixels and adjusts the background model so that the currently observed pixels become categorized as background. The original contents of the region's background model are sent back to the tracker in a "heal" message. The heal message includes a static region mask and heal type which indicating whether the

healed region is an abandoned object or a removed object, based on integral of the edges in the object perimeter as described in Section IV-D. On receiving the heal message from BGS, the tracker can optionally keep the track in a suspended state, ready to be reactivated if the object moves again. Alternatively the entire track can be discarded as a false positive if the region was classified as a removed object.

In this manner, stopped objects are quickly pushed to the background and cease to need active tracking. This reduces the complexity of the tracking problem since fewer tracked objects leads to fewer occlusions and difficult tracking situations, and also reduces the computational load by not "tracking" objects once they are stationary.

When the stopped object begins to move, the background subtraction will detect motion in the region and generate one or more foreground regions in or around the object. Any otherwise unexplained foreground region is compared to the stack of suspended tracks and if a matching track is found it is popped. Otherwise an unexplained foreground region leads to a new track. The tracker send an "unheal" request to BGS, with the old, stored background appearance, which is pushed into the background model, causing the entire object to again be detected as foreground in the following frame, and thus avoiding the "ghost" shown in Figure 5.

VI. Experimental Results

The proposed framework has been tested in the IBM Smart Surveillance Solution [30] product. In this section, we describe the experiments and display some results to demonstrate the effectiveness of our algorithm for background subtraction and foreground analysis in a variety of environments. Notice that the **same parameters** were used for **all sequences**. The BGS algorithm runs about 150 fps for color images and 200 fps for grayscale images at size 160x120 on a 2GB Pentium IV machine with MMX optimization. More quantitative results for the performance evaluation of our system can be found at paper [3].

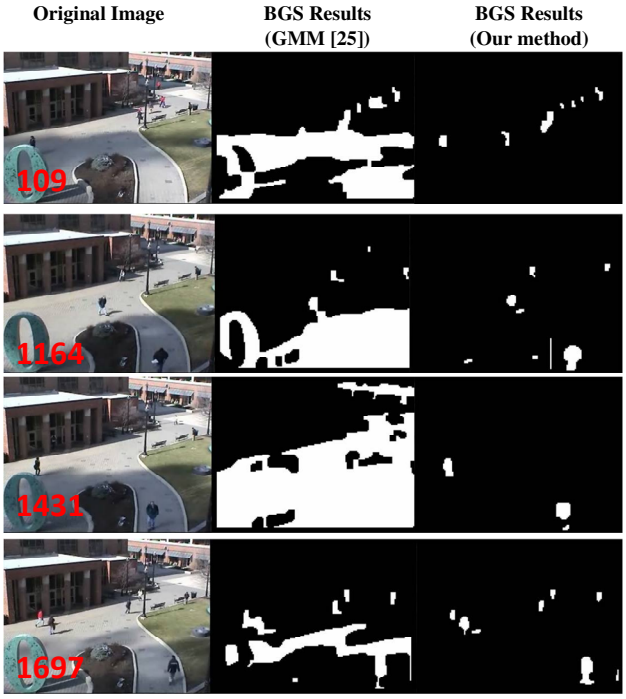

Fig. 6: Example results of our proposed BGS method for a sequence with quick lighting changes [42]. The left column shows the original images. The middle column displays the BGS results from the method of Stauffer and Grimson [25]. Large areas of false positive foreground are detected due to fast moving clouds. The right column demonstrates that our method successfully handles the quick lighting changes by integrating Phong shading reflection model.

A. BGS Results for Sequences with Quick Lighting Changes

Figure 6 shows example results of background subtraction for a sequence with fast-moving clouds on a sunny day which result in large shadow areas on the ground. The sequence is from OTCBVS Benchmark Dataset Collection: OSU Color-Thermal Database [38]. The left column shows the original images. The middle column displays the BGS results from the method of Stauffer and Grimson [25]. Large areas of false positive foreground are detected due to the cloud shadows. The right column demonstrates that our method successfully handles the quick lighting changes by integrating Phong shading reflection model.

Similarly, Figure 7 shows BGS result examples on one sequence with fast lighting changes from the IEEE Performance Evaluation of Tracking and Surveillance Workshop (PETS) 2001 dataset [42]. Same as in Figure 6, the middle column results include large areas of false positive foreground which detected by the mixture of Gaussians method [25]. The right column shows that our method is robust to quick lighting changes.

We further compare the proposed Phong shading reflection model based method with existing methods of [5, 29] to handle quick lighting changes. We observe that Phong shading reflection model based BGS method is more robust at handling swaying trees than [5] and is four-time faster. One example is demonstrated in Figure 8. Paper [29] proposed a texture similarity measure between the current frame and the background image to handle quick lighting changes. Compared to [29], the proposed Phong shading reflection model is more robust at handling strong edges, with comparable speed (see example frames in Figure 9.)

Original Image	**BGS Results (GMM [25])**	**BGS Results (Our method)**

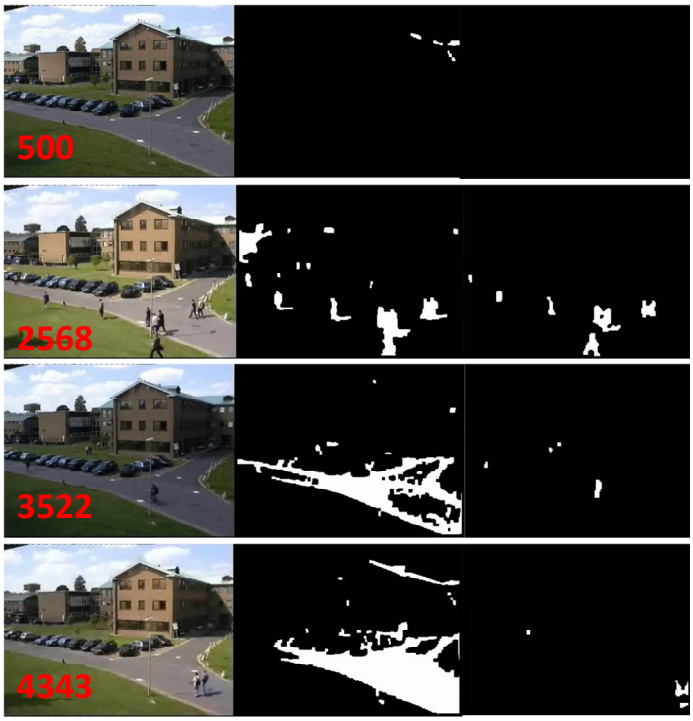

Fig. 7: Example results of our proposed BGS method for a sequence (D3C1) with quick lighting changes from the PETS2001 dataset. The left column shows the original images. The middle column displays the BGS results from method of Stauffer and Grimson [25]. Large areas of false positive foreground are detected due to fast moving clouds. The right column demonstrates that our method successfully handles the quick lighting changes by integrating Phong shading reflection model.

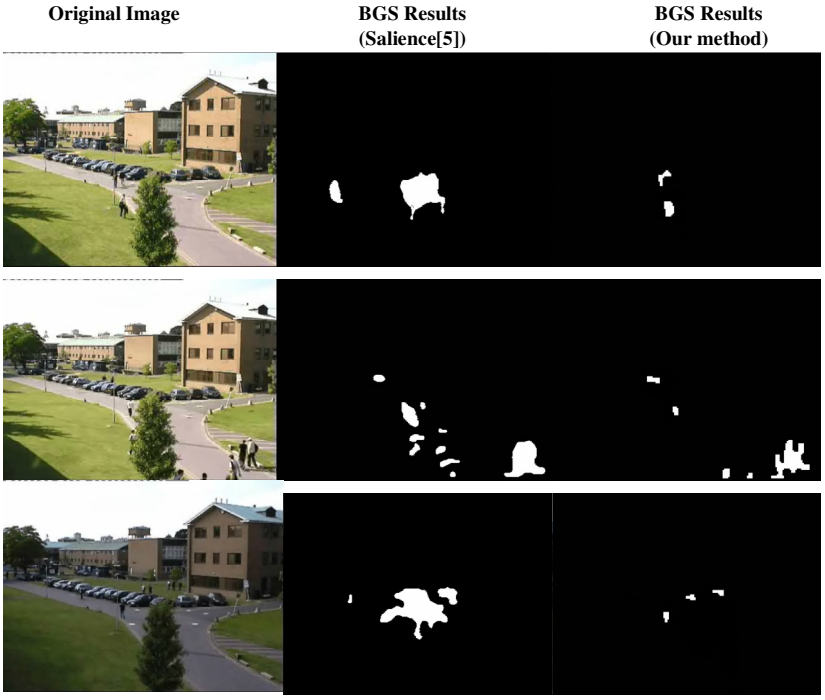

Fig. 8: Example results of our proposed BGS method and salience-based method [5].

Fig. 9: Example results of texture-based quick lighting change handling [29].

B. Results of Static Region Detection and Foreground Fragment Reduction

Figures 10 and 11 demonstrate results of static region detection, heal type classification (abandoned/removed), and foreground fragment reduction. In the test sequence, a chair is left in the center of the camera view at about frame 230. As shown in Figure 10, the chair is detected as a static object at frame 343. The static region is then pushed back to the background model in the next frame (frame 344) to avoid fragments. Further, the heal type is correctly detected as "abandoned" at frame 343. Figure 11 shows that many foreground fragments caused by the static region detected at frame 343 without pushing them back to the background model by using the original mixture of Gaussians method [25]. The fragments had been adapted to the background model at frames of 410. The fragments lasted about 65 frames and made the tracking more difficult.

(a) The chair is detected as a static object at frame 343.

(b) The whole static object is pushed back to BG model at frame 344.

Fig. 10: Examples of static object detection, foreground fragment reduction, and abandoned and removed object discrimination. The chair was detected as a static object at frame 343, and then was pushed back to BG model at frame 344 to avoid fragment problem. The whole static region adaption is finished in one frame.

C. Abandoned and Removed Object Classification

In Figure 10, the static object (chair) was classified as an abandoned object. We evaluate our approach on the PETS 2006 dataset [43] which was designed to test abandoned object detection algorithms in a public space and the i-LIDS video library [44] based on two scenarios: abandoned baggage and parked vehicles.

Figure 12 demonstrates some testing results of abandoned object detection. Figure 13 shows removed object detection.

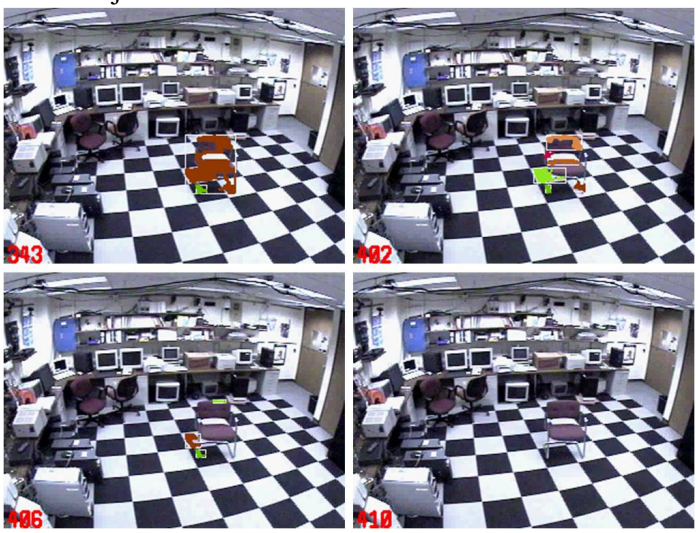

Fig. 11: Examples of the fragment problem without pushing the static region back to the background model. The chair is fully adapted to the background model at frame 410.

Fig. 12: Examples of the abandoned object detection in complex situations. 1st row: abandoned baggage detection in PETS 2006 dataset [43]. 2nd row: detection of parked cars and abandoned objects in i-LIDs dataset [44].

Fig. 13: Examples of detected removed objects in our dataset. The first row shows a moved laptop and the second row demonstrates a parked car driven away.

D. Results of Frame Level Foreground Analysis for Camera Move/Blind

Examples of camera move/blind detection by frame level processing are shown in Figure 14. Figure 14(a) demonstrates the camera moves to a new position and Figure 14(b) displays frame level detection when the power of the camera was turned off. In both situations, larger area of foreground (> 70% of the whole image area) appears.

(a) Camera moved

(b) Camera blinded

Fig. 14: Examples of the frame level processing for camera move detection. (a) Camera moved, (b) Camera power off.

E. Results of Interactions between BGS and Tracking

The feedback mechanism of interactions between BGS and tracking is evaluated on a set of six video sequences include four videos from the PETS2001 dataset [42] of cars and pedestrians crossing a university campus (about 2800 frames each) and two of our own sequences: a top-down view of a four way intersection with cars stopping and waiting for a traffic light to change and an overhead view of a retail store taken through a fish-eye lens.

The feedback mechanism was tested using simple tracking performance metrics comparing the tracker output to hand-labeled ground truth. The ground truth for each sequence consists of bounding boxes drawn around each object approximately every 30 frames, with labeling to associate a particular object's bounding boxes over time. Since the task requires tracking, evaluation is track–based rather than at the BGS level.

The performance analysis processing matches each ground truth track to the tracker's outputs by comparing the distance between the object centroids at each frame (linearly interpolating between the sparse ground truth points), with hysteresis. When at any time t, an object lies close to a ground truth track (within r, here 20, pixels) then the tracks are considered to match for the entire period around t where the tracks lie within $2r$ pixels. Trivial matches (where the match interval between an output track and ground truth track is a subset of the match for another output track, for instance when two tracks cross) are removed.

TABLE I: TRACKING PERFORMANCE RESULTS ON 4 SEQUENCES FROM THE PETS2001 DATASET AND TWO OTHER DATASETS. "UNDER" IS THE PERCENTAGE OF GROUND TRUTH FRAMES MISSING AND "FRAG" IS THE AVERAGE NUMBER OF TRACKS MATCHED TO A GROUND TRUTH TRACK

Sequences	Without feedback	With feedback

	Under%	Frag	Under%	Frag
PETS D1 C1	21.2	1.56	8.1	1.22
PETS D1 C2	27.8	1.36	12.3	1.36
PETS D2 C1	20.3	1.93	17.6	2.27
PETS D2 C2	8.1	1.50	4.6	1.50
Intersection	33.7	1.04	24.3	1.00
Retail Store	13.1	2.38	14.5	1.90

The track matching was verified to correctly match intervals of output tracks to ground truth tracks. The performance tool produces a variety of statistics, including the number of false positives (output tracks not corresponding to any ground truth track) and false negatives (ground truth tracks that have no corresponding output track); the "underrepresentation"—the proportion of ground truth track frames with no correspondence in an output track (e.g. because the object was not detected); and the "fragmentation" — the average number of output tracks matched to each ground truth track (because of gaps in detection, or identity confusion during occlusions).

Quantitative analysis results of performance on six video sequences, from PETS 2001 and two proprietary datasets for particular scenarios, are shown in Table 1. The comparison between experimental results and ground truth averaged across all the six sequences shows that there is a 39% reduction in false negatives (ground truth tracks that are not matched in the tracker output) with a 2.7% increase in the number of false positives (tracker output tracks that do not match any ground truth).

Errors come from a variety of sources: (1) objects that are too small to be detected, particularly in the store and PETS sequences D1C2 and D2C1 which have distant objects labeled; (2) in the intersection sequence several cars are in the scene at the beginning and ghosting effects mean that their tracks are not matched. (3) Failure to resolve occlusions correctly leads to multiple matches for some ground truth tracks.

Qualitative results are shown in Figure 15. This shows how the interaction between BGS and tracking prevents adaptation and fragmentation of the slowly moving and stopped vehicles, and prevents "ghosts" when they move away.

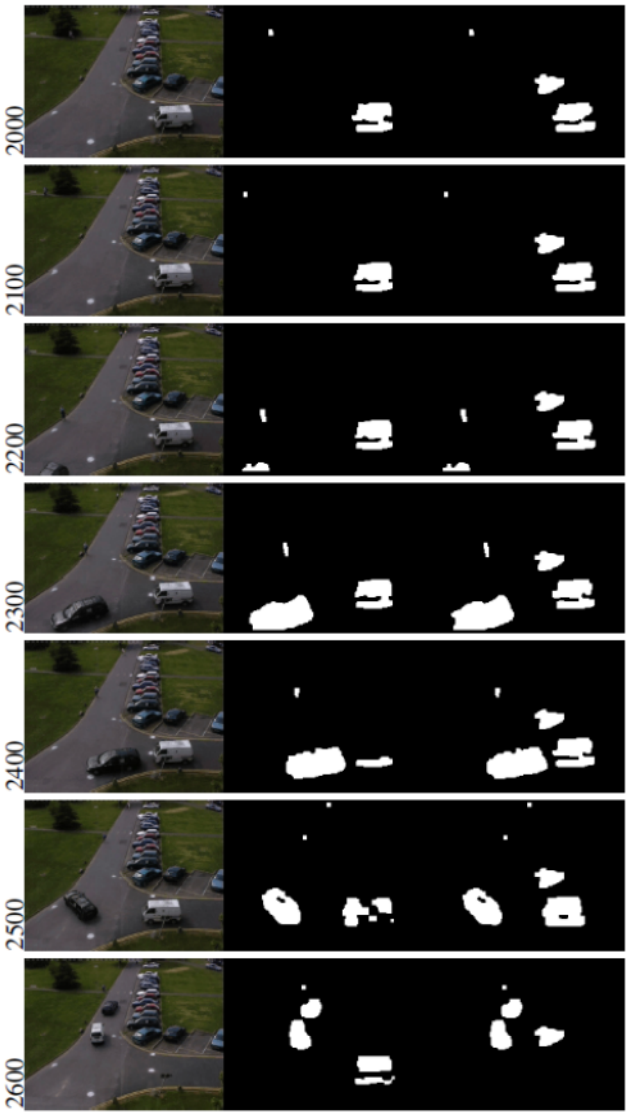

Fig. 15: Selected frames from a PETS2001 video sequence and corresponding foreground regions, demonstrating BGS adaptation without feedback from tracking (middle column) and with the feedback mechanisms (right column). Note the fragmentation (fr.2500) and ghosting (fr.2600) on the middle column. The central stopped car is lost on the middle, but maintained on the right.

VII. Discussion and Conclusion

We have presented a new framework to robustly and efficiently analyze foreground at multiple levels for video surveillance applications. The foregrounds are detected at the pixel level by using a mixture of Gaussians BGS method. We have enhanced the method to handle quick lighting changes by integrating the Phong shading model. Static objects are detected by using the same Gaussian mixture models and are further classified as abandoned or removed objects by two methods: 1) analyzing the change in the amount of edge energy associated with the boundaries of the static foreground regions, and 2) checking the compatibility of the static region with its surroundings. At the region level, whole static regions are pushed back to the background model to avoid fragmentation. Foreground analysis at the whole frame level is developed to deal with camera move/blind and extreme lighting changes (e.g., turned on or turned off lights).

To improve the tracking accuracy, we have created a feedback mechanism of interactions between BGS and tracking to handle slow moving and stopped objects. The feedback mechanism allows that BGS processing receives "heal request", "unheal request", and "hold in foreground" from tracking at object (foreground blobs) level. The proposed algorithm works well in our real-time video surveillance system for most situations. However, it may fail in low contrast situations where the color of the object is very similar to the background, e.g., black bag on a black background.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their constructive comments and suggestions that help to improve the quality of this manuscript.

REFERENCES

[1] R. Abbott and L. Williams, "Multiple target tracking with lazy background subtraction and connected components analysis", Tech. Rep., University of New Mexico, June 2005.

[2] T. Boult, R. Micheals, X. Gao, and M. Eckmann, "Into the woods: Visual surveillance of non-cooperative and camouflaged targets in compex outdoor settings", Proceedings of the IEEE, vol. 89, no. 10, October 2001.

[3] L. Brown, A.W. Senior, Y. Tian, J. Connell, A. Hampapur, Chiao-fe Shu, Hans Merkl, and Max Lu, "Performance Evaluation of Surveillance Systems Under Varying Conditions," IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2005.

[4] S. Cheung and C. Kamath, "Robust background subtraction with foreground validation for urban traffic video", EURASIP Journal of Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems, 2005.

[5] J. Connell, A. Senior, A. Hampapur, Y. Tian, L. Brown, and S. Pankanti, "Detection and tracking in the IBM PeopleVision system", in IEEE ICME, June 2004.

[6] G. Cristani, M. Bicegi, and V. Murino, "Integrated Region- and Pixel-based Approach to Background Modeling", Proceedings of the MOTION, 2002.

[7] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," IEEE Trans. on PAMI, 25: (10), October 2003.

[8] E. Durucan and T. Ebrahimi, Change detection and background extraction by linear algebra, Proc. IEEE, 89(10):1368–1381, 2001.

[9] H. Eng, J. Wang, A. Kam, and W. Yau, "Novel Region-based Modeling for Human Detection within High Dynamic Aquatic Environment," Proceedings on CVPR, 2004.

[10] M. Harville, "A Framework for High-level Feedback to adaptive, per-pixel, Mixture-of-Gaussian Background Models", Proceedings on ECCV, 2002.

[11] E. Hayman and J. Eklundh, Statistical Background Subtraction for a Mobile Observer, IEEE ICCV, 2003.

[12] K. Huang, D. Tao, Y Yuan, X. Li, and T. Tan., "Biologically Inspired Features for Scene Classification in Video Surveillance," IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2010.

[13] S. Huwer and H. Niemann, "Adaptive Change Detection for Real-time Surveillance applications," Proc. of the 3rd IEEE Workshop on Visual Surveillance, pp.37-45, 2000.

[14] O. Javed, K. Shafique, and M. Shah, "A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information," IEEE Workshop on Motion and Video Computing, 2002.

[15] P. KaewTraKulPong and R. Bowden, “An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection,” In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, 2001.

[16] L. Li and M.K.H. Leung, “Integrating Intensity and Texture Differences for Robust Change Detection”, IEEE Transactions on Image Processing, Vol. 11, No. 2, 2002.

[17] L. Li, W. Huang, I. Gu, and Q. Tian, “Statistical Modeling of Complex Backgrounds for Foreground Object Detection”, IEEE Transaction on Image Processing, Vol. 13, No. 11, 2004.

[18] A. Mittal and N. Paragios, “Motion-based Background Subtraction using Adaptive Kernel Density Estimation,” Proceedings on CVPR, 2004.

[19] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, “Background Modeling and Subtraction of Dynamic Scenes”, Proc. on ICCV, Pages 1305–1312, 2003.

[20] B. Phong, “Illumination for computer generated pictures,” *Commun. ACM*, vol. 18, pp. 311–317, 1975.

[21] A. Pnevmatikakis and L. Polymenakos, “Kalman tracking with target feedback on adaptive background learning”, in Workshop on Multimodal Interaction and Related Machine Learning Algorithms, 2006.

[22] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, Image change detection algorithms: a systematic survey, IEEE Transactions on Image Processing, Volume: 14, Issue: 3, 2005.

[23] A. Senior, “Tracking with probabilistic appearance models”, in Third International workshop on Performance Evaluation of Tracking and Surveillance systems, June 2002.

[24] Y. Sheikh, O. Javed, and T. Kanade, Background Subtraction for Freely Moving Cameras, IEEE ICCV, 2009.

[25] Stauffer and W.E.L. Grimson, “Adaptive Background mixture Models for Real-time Tracking”, CVPR99, June, 1999.

[26] L. Stefano, F. Tombari, S. Mattoccia, and E. Lisi, Robust and accurate change detection under sudden illumination variations, Workshop on Multi-dimensional and Multi-view Image Processing, 2007.

[27] Y. Sugaya and K. Kanatani, Extracting Moving Objects from a Moving Camera Video Sequence, Memoirs of the Faculty of Engineering, Okayama University, Vol.39, pp.56-62, January, 2005.

[28] L. Taycher, J.W. Fisher III, and T. Darrell, “Incorporating object tracking feedback into background maintenance framework”, in IEEE Workshop on Motion and Video Computing, 2005.

[29] Y. Tian, Max Lu, and A. Hampapur, “Robust and Efficient Foreground Analysis for Real-time Video Surveillance,” IEEE CVPR, San Diego. June, 2005.

[30] Y. Tian, A.W. Senior, A. Hampapur, L. Brown, C. Shu, and M. Lu, "IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework", Machine Vision and Applications, 19:315-327, 2008.

[31] Y. Tian, R. S. Feris and A. Hampapur. “Real-Time Detection of Abandoned and Removed Objects in Complex Environments”. IEEE International Workshop on Visual Surveillance (in conjunction with ECCV'08), Marseille, France, 2008.

[32] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance”, in Proc. IEEE International Conference on Computer Vision, 1999, vol. 1, pp. 255–261.

[33] P. Venetianer, Z. Zhang, W. Yin, and A. Lipton, “Stationary target detection using the objectvideo surveillance system”, in Advanced Video and Signal-based Surveillance, 2007.

[34] J. Wang, G. Bebis, and R. Miller, “Robust video-based surveillance by integrating target detection with tracking”, in Conference on Computer Vision and Pattern Recognition, 2006.

[35] S. Watanabe, K. Miyajima, and N. Mukawa, “Detecting changes of buildings from aerial images using shadow and shading model,” in *ICPR 98*, 1998, pp. 1408–1412.

[36] B. Xie, V. Ramesh, and T. Boult, “Sudden illumination change detection using order consistency,” *Image and Vision Computing*, vol. 22, no. 2, pp. 117–125, February 2004.

[37] J. Yao and J.-M. Odobez, “Multi-layer background subtraction based on color and texture”, in Proc. IEEE Conference on Visual Surveillance, 2007.

[38] J. Davis and V. Sharma, "Background-Subtraction using Contour-based Fusion of Thermal and Visible Imagery," *Computer Vision and Image Understanding*, Vol 106, No. 2-3, 2007. IEEE OTCBVS WS Series Bench: http://www.cse.ohio-state.edu/otcbvs-bench/

[39] Y. Yuan, Y. Pang, J. Pan, and X. Li, "Scene Segmentation Based on IPCA for Visual Surveillance," Neurocomputing (Elsevier), vol. 72, nos. 10-12, pp. 2450-2454, 2009.

[40] H. Zhou, Y. Yuan, Y. Zhang, C. Shi, "Non-rigid Object Tracking in Complex Scenes," Pattern Recognition Letters (Elsevier), vol. 30, no. 2, pp. 98-102, 2009.

[41] H. Zhou, Y. Yuan, and C. Shi, "Object Tracking using SIFT Features and Mean Shift," Computer Vision and Image Understanding (Elsevier), vol. 113, no. 2, pp. 345-352, 2009.

[42] PETS 2001 Benchmark Data, http://www.cvg.rdg.ac.uk/PETS2001/.

[43] PETS 2006 Benchmark Data, http://www.cvg.rdg.ac.uk/PETS2006/.

[44] i-LIDS Dataset for AVSS 2007, ftp://motinas.elec.qmul.ac.uk/pub/iLids.

[45] Who’s Watching? Video Camera Surveillance in New York City and the Need for Public Oversight, A Special Report by the New York Civil Liberties Union, 2006.

[46] Big Apple is Watching You. How many surveillance cameras are there in Manhattan? 5/3/2010. http://www.slate.com/id/2252729.