# Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance

AHMED ELGAMMAL, RAMANI DURAISWAMI, MEMBER, IEEE, DAVID HARWOOD, AND LARRY S. DAVIS, FELLOW, IEEE

*Invited Paper*

*Automatic understanding of events happening at a site is the ultimate goal for many visual surveillance systems. Higher level understanding of events requires that certain lower level computer vision tasks be performed. These may include detection of unusual motion, tracking targets, labeling body parts, and understanding the interactions between people. To achieve many of these tasks, it is necessary to build representations of the appearance of objects in the scene. This paper focuses on two issues related to this problem. First, we construct a statistical representation of the scene background that supports sensitive detection of moving objects in the scene, but is robust to clutter arising out of natural scene variations. Second, we build statistical representations of the foreground regions (moving objects) that support their tracking and support occlusion reasoning. The probability density functions (pdfs) associated with the background and foreground are likely to vary from image to image and will not in general have a known parametric form. We accordingly utilize general nonparametric kernel density estimation techniques for building these statistical representations of the background and the foreground. These techniques estimate the pdf directly from the data without any assumptions about the underlying distributions. Example results from applications are presented.*

***Keywords*—***Background subtraction, color modeling, kernel density estimation, occlusion modeling, tracking, visual surveillance.*

## I. INTRODUCTION

In automated surveillance systems, cameras and other sensors are typically used to monitor activities at a site with the goal of automatically understanding events happening at the site. Automatic event understanding would enable functionalities such as detection of suspicious activities and site security. Current systems archive huge volumes of video for eventual off-line human inspection. The automatic detection of events in videos would facilitate efficient archiving and automatic annotation. It could be used to direct the attention of human operators to potential problems. The automatic detection of events would also dramatically reduce the bandwidth required for video transmission and storage as only interesting pieces would need to be transmitted or stored.

Higher level understanding of events requires certain lower level computer vision tasks to be performed such as detection of unusual motion, tracking targets, labeling body parts, and understanding the interactions between people. For many of these tasks, it is necessary to build representations of the appearance of objects in the scene. For example, the detection of unusual motions can be achieved by building a representation of the scene background and comparing new frames with this representation. This process is called *background subtraction*. Building representations for foreground objects (targets) is essential for tracking them and maintaining their identities. This paper focuses on two issues: how to construct a statistical representation of the scene background that supports sensitive detection of moving objects in the scene and how to build statistical representations of the foreground (moving objects) that support their tracking.

One useful tool for building such representations is statistical modeling, where a process is modeled as a random variable in a feature space with an associated probability density function (pdf). The density function could be represented parametrically using a specified statistical distribution, that

Manuscript received May 31, 2001; revised February 15, 2002. This work was supported in part by the ARDA Video Analysis and Content Exploitation project under Contract MDA 90 400C2110 and in part by Philips Research.

A. Elgammal is with the Computer Vision Laboratory, University of Maryland Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: elgammal@cs.umd.edu).

R. Duraiswami, D. Harwood, and L. S. Davis are with the Computer Vision Laboratory, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA (e-mail: ramani@umiacs.umd.edu; harwood@umiacs.umd.edu; lsd@cs.umd.edu).

Publisher Item Identifier 10.1109/JPROC.2002.801448.

is assumed to approximate the actual distribution, with the associated parameters estimated from training data. Alternatively, nonparametric approaches could be used. These estimate the density function directly from the data without any assumptions about the underlying distribution. This avoids having to choose a model and estimating its distribution parameters.

A particular nonparametric technique that estimates the underlying density, avoids having to store the complete data, and is quite general is the kernel density estimation technique. In this technique, the underlying pdf is estimated as

$$\hat{f}(x) = \sum_i \alpha_i K(x - x_i) \tag{1}$$

where $K$ is a "kernel function" (typically a Gaussian) centered at the data points in feature space, $x_i, i = 1 \dots n$, and $\alpha_i$ are weighting coefficients (typically uniform weights are used, i.e., $\alpha_i = 1/n$). Kernel density estimators asymptotically converge to any density function [1], [2]. This property makes these techniques quite general and applicable to many vision problems where the underlying density is not known.

In this paper, kernel density estimation techniques are utilized for building representations for both the background and the foreground. We present an adaptive background modeling and background subtraction technique that is able to detect moving targets in challenging outdoor environments with moving trees and changing illumination. We also present a technique for modeling foreground regions and show how it can be used for segmenting major body parts of a person and for segmenting groups of people.

## II. Kernel Density Estimation Techniques

Given a sample $S = \{x_i\}_{i=1\dots N}$ from a distribution with density function $p(x)$, an estimate $\hat{p}(x)$ of the density at $x$ can be calculated using

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x - x_i) \tag{2}$$

where $K_\sigma$ is a kernel function (sometimes called a "window" function) with a bandwidth (scale) $\sigma$ such that $K_\sigma(t) = (1/\sigma)K(t/\sigma)$. The kernel function $K$ should satisfy $K(t) \geq 0$ and $\int K(t)\,dt = 1$. We can think of (2) as estimating the pdf by averaging the effect of a set of kernel functions centered at each data point. Alternatively, since the kernel function is symmetric, we can also regard this computation as averaging the effect of a kernel function centered at the estimation point and evaluated at each data point. Kernel density estimators asymptotically converge to any density function with sufficient samples [1], [2]. This property makes the technique quite general for estimating the density of any distribution. In fact, all other nonparametric density estimation methods, e.g., histograms, can be shown to be asymptotically kernel methods [1].

For higher dimensions, products of one-dimensional (1-D) kernels [1] can be used as

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} K_{\sigma_j}\left(\frac{(x - x_i)_j}{\sigma_j}\right) \tag{3}$$

where the same kernel function is used in each dimension with a suitable bandwidth $\sigma_j$ for each dimension. We can avoid having to store the complete data set by weighting the samples as

$$\hat{p}(x) = \sum_{i=1} \alpha_i K_\sigma(x - x_i)$$

where the $\alpha_i$'s are weighting coefficients that sum up to one.

A variety of kernel functions with different properties have been used in the literature. Typically the Gaussian kernel is used for its continuity, differentiability, and locality properties. Note that choosing the Gaussian as a kernel function is different from fitting the distribution to a Gaussian model (normal distribution). Here, the Gaussian is only used as a function to weight the data points. Unlike parametric fitting of a mixture of Gaussians, kernel density estimation is a more general approach that does not assume any specific shape for the density function. A good discussion of kernel estimation techniques can be found in [1]. The major drawback of using the nonparametric kernel density estimator is its computational cost. This becomes less of a problem as the available computational power increases and as efficient computational methods have become available recently [3], [4].

## III. Modeling the Background

### A. Background Subtraction: A Review

*1) The Concept:* In video surveillance systems, stationary cameras are typically used to monitor activities at outdoor or indoor sites. Since the cameras are stationary, the detection of moving objects can be achieved by comparing each new frame with a representation of the scene background. This process is called background subtraction and the scene representation is called the background model. Typically, background subtraction forms the first stage in an automated visual surveillance system. Results from background subtraction are used for further processing, such as tracking targets and understanding events.

A central issue in building a representation for the scene background is what features to use for this representation or, in other words, what to model in the background. In the literature, a variety of features have been used for background modeling, including pixel-based features (pixel intensity, edges, disparity) and region-based features (e.g., block correlation). The choice of the features affects how the background model tolerates changes in the scene and the granularity of the detected foreground objects.

In any indoor or outdoor scene, there are changes that occur over time and may be classified as changes to the scene background. It is important that the background model tolerates these kind of changes, either by being invariant to them or by adapting to them. These changes can be local, affecting only part of the background, or global, affecting the entire background. The study of these changes is essential to understand the motivations behind different background subtraction techniques. We classify these changes according to their source.

**Illumination changes:**

- gradual change in illumination, as might occur in outdoor scenes due to the change in the location of the sun;

- sudden change in illumination as might occur in an indoor environment by switching the lights on or off, or in an outdoor environment by a change between cloudy and sunny conditions;
- shadows cast on the background by objects in the background itself (e.g., buildings and trees) or by moving foreground objects.

**Motion changes:**

- image changes due to small camera displacements (these are common in outdoor situations due to wind load or other sources of motion which causes global motion in the images);
- motion in parts of the background, for example, tree branches moving with the wind or rippling water.

**Changes introduced to the background:** These include any change in the geometry or the appearance of the background of the scene introduced by targets. Such changes typically occur when something relatively permanent is introduced into the scene background (for example, if somebody moves (introduces) something from (to) the background, or if a car is parked in the scene or moves out of the scene, or if a person stays stationary in the scene for an extended period).

*2) Practice:* Many researchers have proposed methods to address some of the issues regarding the background modeling, and we provide a brief review of the relevant work here.

Pixel intensity is the most commonly used feature in background modeling. If we monitor the intensity value of a pixel over time in a completely static scene, then the pixel intensity can be reasonably modeled with a Gaussian distribution $N(\mu, \sigma^2)$, given that the image noise over time can be modeled by a zero mean Gaussian distribution $N(0, \sigma^2)$. This Gaussian distribution model for the intensity value of a pixel is the underlying model for many background subtraction techniques. For example, one of the simplest background subtraction techniques is to calculate an average image of the scene, subtract each new frame from this image, and threshold the result. This basic Gaussian model can adapt to slow changes in the scene (for example, gradual illumination changes) by recursively updating the model using a simple adaptive filter. This basic adaptive model is used in [5]; also, Kalman filtering for adaptation is used in [6]–[8].

Typically, in outdoor environments with moving trees and bushes, the scene background is not completely static. For example, one pixel can be the image of the sky in one frame, a tree leaf in another frame, a tree branch in a third frame, and some mixture subsequently. In each situation, the pixel will have a different intensity (color), so a single Gaussian assumption for the pdf of the pixel intensity will not hold. Instead, a generalization based on a mixture of Gaussians has been used in [9]–[11] to model such variations. In [9] and [10], the pixel intensity was modeled by a mixture of $K$ Gaussian distributions ($K$ is a small number from 3 to 5). The mixture is weighted by the frequency with which each of the Gaussians explains the background. In [11], a mixture of three Gaussian distributions was used to model the pixel value for traffic surveillance applications. The pixel intensity was modeled as a weighted mixture of three Gaussian distributions corresponding to road, shadow, and vehicle distribution. Adaptation of the Gaussian mixture models can be achieved using an incremental version of the EM algorithm.

In [12], linear prediction using the Wiener filter is used to predict pixel intensity given a recent history of values. The prediction coefficients are recomputed each frame from the sample covariance to achieve adaptivity. Linear prediction using the Kalman filter was also used in [6]–[8].

All of the previously mentioned models are based on statistical modeling of pixel intensity with the ability to adapt the model. While pixel intensity is not invariant to illumination changes, model adaptation makes it possible for such techniques to adapt to gradual changes in illumination. On the other hand, a sudden change in illumination presents a challenge to such models.

Another approach to model a wide range of variations in the pixel intensity is to represent these variations as discrete states corresponding to modes of the environment, e.g., lights on/off or cloudy/sunny skies. Hidden Markov models (HMMs) have been used for this purpose in [13] and [14]. In [13], a three-state HMM has been used to model the intensity of a pixel for a traffic-monitoring application where the three states correspond to the background, shadow, and foreground. The use of HMMs imposes a temporal continuity constraint on the pixel intensity, i.e., if the pixel is detected as a part of the foreground, then it is expected to remain part of the foreground for a period of time before switching back to be part of the background. In [14], the topology of the HMM representing global image intensity is learned while learning the background. At each global intensity state, the pixel intensity is modeled using a single Gaussian. It was shown that the model is able to learn simple scenarios like switching the lights on and off.

Alternatively, edge features have also been used to model the background. The use of edge features to model the background is motivated by the desire to have a representation of the scene background that is invariant to illumination changes. In [15], foreground edges are detected by comparing the edges in each new frame with an edge map of the background which is called the background “primal sketch.” The major drawback of using edge features to model the background is that it would only be possible to detect edges of foreground objects instead of the dense connected regions that result from pixel-intensity-based approaches. A fusion of intensity and edge information was used in [16].

Block-based approaches have been also used for modeling the background. Block matching has been extensively used for change detection between consecutive frames. In [17], each image block is fit to a second-order bivariate polynomial and the remaining variations are assumed to be noise. A statistical likelihood test is then used to detect blocks with significant change. In [18], each block was represented with its median template over the background learning period and its block standard deviation. Subsequently, at each new frame, each block is correlated with its corresponding template, and blocks with too much deviation relative to the measured standard deviation are considered to be foreground. The major drawback with block-based approaches is that the detection unit is a whole image block and therefore they are only suitable for coarse detection.

In order to monitor wide areas with sufficient resolution, cameras with zoom lenses are often mounted on pan-tilt platforms. This enables high-resolution imagery to be obtained from any arbitrary viewing angle from the location where the camera is mounted. The use of background subtraction in such situations requires a representation of the scene background for any arbitrary pan-tilt-zoom combination, which is an extension to the original background subtraction concept with a stationary camera. In [19], image mosaicing techniques are used to build panoramic representations of the scene background. Alternatively, in [20], a representation of the scene background as a finite set of images on a virtual polyhedron is used to construct images of the scene background at any arbitrary pan-tilt-zoom setting. Both techniques assume that the camera rotation is around its optical axis and so that there is no significant motion parallax.

### B. Nonparametric Background Modeling

In this section, we describe a background model and a background subtraction process that we have developed, based on nonparametric kernel density estimation. The model uses pixel intensity (color) as the basic feature for modeling the background. The model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the density function of the pixel intensity distribution. Therefore, the model is able to estimate the probability of any newly observed intensity value. The model can handle situations where the background of the scene is cluttered and not completely static but contains small motions that are due to moving tree branches and bushes. The model is updated continuously and therefore adapts to changes in the scene background.

*1) Background Subtraction:* Let $x_1, x_2, \ldots, x_N$ be a sample of intensity values for a pixel. Given this sample, we can obtain an estimate of the pixel intensity pdf at any intensity value using kernel density estimation. Given the observed intensity $x_t$ at time $t$, we can estimate the probability of this observation as

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x_t - x_i) \tag{4}$$

where $K_\sigma$ is a kernel function with bandwidth $\sigma$. This estimate can be generalized to use color features by using kernel products as

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} K_{\sigma_j}\left(x_{t_j} - x_{i_j}\right) \tag{5}$$

where $x_t$ is a $d$-dimensional color feature and $K_{\sigma_j}$ is a kernel function with bandwidth $\sigma_j$ in the $j$th color space dimension. If we choose our kernel function $K$ to be Gaussian, then the density can be estimated as

$$\Pr(x_t) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{(x_{t_j} - x_{i_j})^2}{\sigma_j^2}}. \tag{6}$$

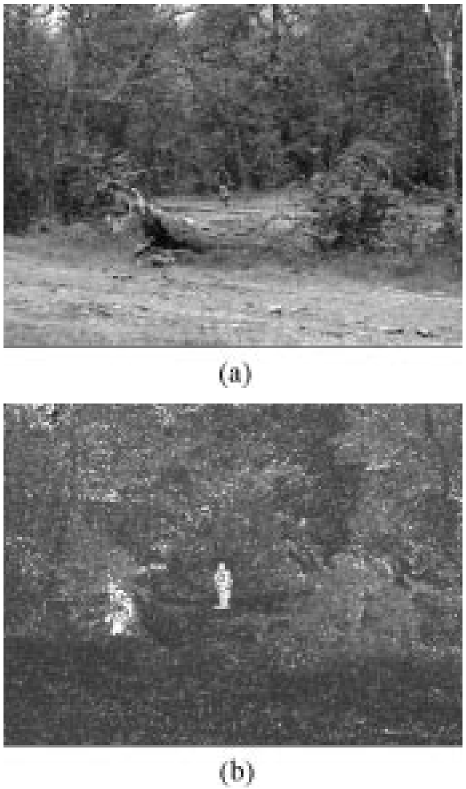

**Fig. 1.** Background Subtraction. (a) Original image. (b) Estimated probability image.

Using this probability estimate, the pixel is considered to be a foreground pixel if $\Pr(x_t) < \mathrm{th}$, where the threshold $\mathrm{th}$ is a global threshold over all the images that can be adjusted to achieve a desired percentage of false positives. Practically, the probability estimation in (6) can be calculated in a very fast way using precalculated lookup tables for the kernel function values given the intensity value difference $(x_t - x_i)$ and the kernel function bandwidth. Moreover, a partial evaluation of the sum in (6) is usually sufficient to surpass the threshold at most image pixels, since most of the image is typically from the background. This allows us to construct a very fast implementation.

Since kernel density estimation is a general approach, the estimate of (4) can converge to any pixel intensity density function. Here, the estimate is based on the most recent $N$ samples used in the computation. Therefore, adaptation of the model can be achieved simply by adding new samples and ignoring older samples [21]. Fig. 1(b) shows the estimated background probability where brighter pixels represent lower background probability pixels.

One major issue that needs to be addressed when using kernel density estimation technique is the choice of suitable kernel bandwidth (scale). Theoretically, as the number of samples reaches infinity, the choice of the bandwidth is insignificant and the estimate will approach the actual density. Practically, since only a finite number of samples are used and the computation must be performed in real time, the choice of suitable bandwidth is essential. Too small a bandwidth will lead to a ragged density estimate, while too wide a bandwidth will lead to an over-smoothed density estimate [2]. Since the expected variations in pixel intensity over time are different from one location to another in the image, a different kernel bandwidth is used for each pixel. Also, a different kernel bandwidth is used for each color channel.

To estimate the kernel bandwidth $\sigma_j^2$ for the $j$th color channel for a given pixel, we compute the median absolute deviation over the sample for consecutive intensity values of the pixel. That is, the median $m$ of $|x_i - x_{i+1}|$ for each consecutive pair $(x_i, x_{i+1})$ in the sample is calculated independently for each color channel. The motivation behind the use of median of absolute deviation is that pixel intensities over time are expected to have jumps because different objects (e.g., sky, branch, leaf, and mixtures when an edge passes through the pixel) are projected onto the same pixel at different times. Since we are measuring deviations between two consecutive intensity values, the pair $(x_i, x_{i+1})$ usually comes from the same local-in-time distribution, and only a few pairs are expected to come from cross distributions (intensity jumps). The median is a robust estimate and should not be affected by few jumps.

If we assume that this local-in-time distribution is Gaussian $N(\mu, \sigma^2)$, then the distribution for the deviation $(x_i - x_{i+1})$ is also Gaussian $N(0, 2\sigma^2)$. Since this distribution is symmetric, the median of the absolute deviations $m$ is equivalent to the quarter percentile of the deviation distribution. That is,

$$\Pr(N(0, 2\sigma^2) > m) = 0.25$$

and therefore the standard deviation of the first distribution can be estimated as

$$\sigma = \frac{m}{0.68\sqrt{2}}.$$

Since the deviations are integer gray scale (color) values, linear interpolation is used to obtain more accurate median values.

*2) Probabilistic Suppression of False Detection:* In outdoor environments with fluctuating backgrounds, there are two sources of false detections. First, there are false detections due to random noise which are expected to be homogeneous over the entire image. Second, there are false detections due to small movements in the scene background that are not represented by the background model. This can occur locally, for example, if a tree branch moves further than it did during model generation. This can also occur globally in the image as a result of small camera displacements caused by wind load, which is common in outdoor surveillance and causes many false detections. These kinds of false detections are usually spatially clustered in the image, and they are not easy to eliminate using morphological techniques or noise filtering because these operations might also affect detection of small and/or occluded targets.

If a part of the background (a tree branch, for example) moves to occupy a new pixel, but it was not part of the model for that pixel, then it will be detected as a foreground object. However, this object will have a high probability of being a part of the background distribution corresponding to its original pixel. Assuming that only a small displacement can occur between consecutive frames, we decide if a detected pixel is caused by a background object that has moved by considering the background distributions of a small neighborhood of the detection location.

Let $x_t$ be the observed value of a pixel $x$ detected as a foreground pixel at time $t$. We define the pixel displacement probability $P_{\mathcal{N}}(x_t)$ to be the maximum probability that the observed value, $x_t$, belongs to the background distribution of some point in the neighborhood $\mathcal{N}$ of $x$

$$P_{\mathcal{N}}(x_t) = \max_{y \in \mathcal{N}(x)} \Pr(x_t \mid B_y)$$

where $B_y$ is the background sample for pixel $y$, and the probability estimation $\Pr(x_t \mid B_y)$ is calculated using the kernel function estimation as in (6). By thresholding $P_{\mathcal{N}}$ for detected pixels, we can eliminate many false detections due to small motions in the background scene. To avoid losing true detections that might accidentally be similar to the background of some nearby pixel (e.g., camouflaged targets), a constraint is added that the whole detected foreground object must have moved from a nearby location, and not only some of its pixels. The component displacement probability $P_{\mathcal{C}}$ is defined to be the probability that a detected connected component $\mathcal{C}$ has been displaced from a nearby location. This probability is estimated by

$$P_{\mathcal{C}} = \prod_{x \in \mathcal{C}} P_{\mathcal{N}(x)}.$$

For a connected component corresponding to a real target, the probability that this component has displaced from the background will be very small. So, a detected pixel $x$ will be considered to be a part of the background only if $(P_{\mathcal{N}}(x) > \mathrm{th}_1) \wedge (P_{\mathcal{C}}(x) > \mathrm{th}_2)$.

Fig. 2 illustrates the effect of the second stage of detection. The result after the first stage is shown in Fig. 2(b). In this example, the background has not been updated for several seconds, and the camera has been slightly displaced during this time interval, so we see many false detections along high-contrast edges. Fig. 2(c) shows the result after suppressing the detected pixels with high displacement probability. Most false detections due to displacement were eliminated, and only random noise that is uncorrelated with the scene remains as false detections. However, some true detected pixels were also lost. The final result of the second stage of the detection is shown in Fig. 2(d), where the component displacement probability constraint was added. Fig. 3(b) shows results for a case where as a result of the wind load the camera is shaking slightly, resulting in a lot of clustered false detections, especially on the edges. After probabilistic suppression of false detection [Fig. 3(c)], most of these clustered false detection are suppressed, while the small target on the left side of the image remains.

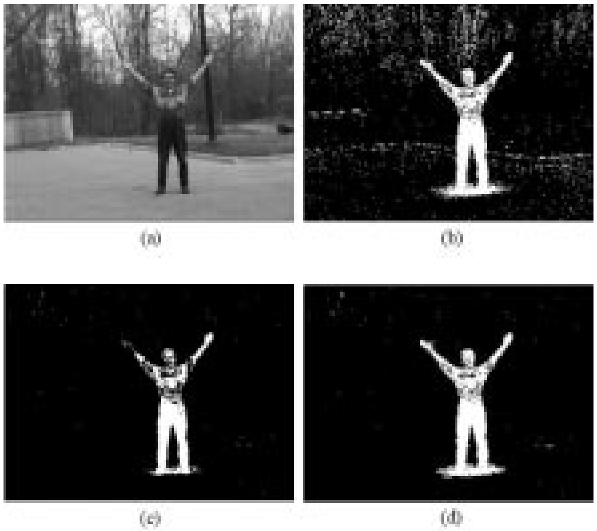

**Fig. 2.** Effect of the second stage of detection on suppressing false detections. (a) Original image. (b) First stage detection result. (c) Suppressing pixels with high displacement probabilities. (d) Result using component displacement probability constraint.

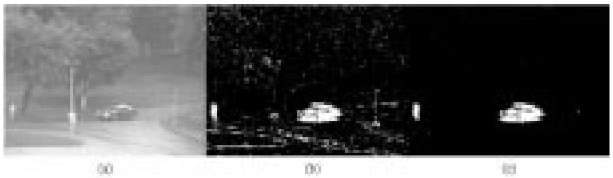

**Fig. 3.** (a) Original image. (b) Result after the first stage of detection. (c) Result after the second stage.

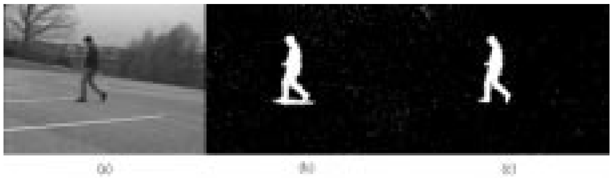

**Fig. 4.** (a) Original image. (b) Detection using $(R, G, B)$ color space. (c) Detection using chromaticity coordinates $(r, g)$.

*3) Working With Color:* The detection of shadows as part of the foreground regions is a source of confusion for subsequent phases of analysis. It is desirable to discriminate between targets and their shadows. Color information is useful for suppressing shadows from the detection by separating color information from lightness information. Given three color variables, $R$, $G$, and $B$, the chromaticity coordinates are $r = R/(R+G+B)$, $g = G/(R+G+B)$, and $b = B/(R+G+B)$, where $r+g+b = 1$ [22]. Using chromaticity coordinates for detection has the advantage of being more insensitive to small changes in illumination that arise due to shadows. Fig. 4 shows the results of detection using both $(R, G, B)$ space and $(r, g)$ space. The figure shows that using the chromaticity coordinates allows detection of the target without detecting its shadow. It must be noticed that the background subtraction technique we describe in Section III-B can be used with any color space (e.g., HSV, YUV, etc.).

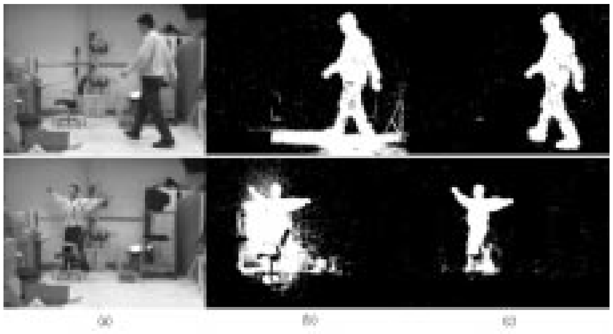

**Fig. 5.** (a) Original image. (b) Detection using $(R, G, B)$ color space. (c) detection using chromaticity coordinates $(r, g)$ and the lightness variable, $s$.

Although using chromaticity coordinates helps in the suppression of shadows, they have the disadvantage of losing lightness information. Lightness is related to the differences in whiteness, blackness, and grayness between different objects [23]. For example, consider the case where the target wears a white shirt and walks against a gray background. In this case, there is no color information. Since both white and gray have the same chromaticity coordinates, the target may not be detected.

To address this problem, we also need to use a measure of lightness at each pixel. We use $s = R + G + B$ as a lightness measure. Consider the case where the background is completely static, and let the expected value for a pixel be $\langle r, g, s \rangle$. Assume that this pixel is covered by shadow in frame $t$ and let $\langle r_t, g_t, s_t \rangle$ be the observed value for this pixel at this frame. Then, it is expected that $\alpha \leq (s_t/s) \leq 1$. That is, it is expected that the observed value $s_t$ will be darker than the normal value $s$ up to a certain limit, $\alpha s \leq s_t$, which corresponds to the intuition that at most a fraction $(1-\alpha)$ of the light coming to this pixel can be reduced by a target shadow. A similar effect is expected for highlighted background, where the observed value can be brighter than the expected value up to a certain limit. Similar reasoning was used by [24].

In our case, where the background is not static, there is no single expected value for each pixel. Let $A$ be the sample values representing the background for a certain pixel, each represented as $x_i = \langle r_i, g_i, s_i \rangle$, and let $x_t = \langle r_t, g_t, s_t \rangle$ be the observed value at frame $t$. Then, we can select a subset $B \subseteq A$ of sample values that are relevant to the observed lightness $s_t$. By relevant, we mean those values from the sample which, if affected by shadows, can produce the observed lightness of the pixel. That is,

$$B = \left\{ x_i \;\middle|\; x_i \in A \wedge \alpha \leq \frac{s_t}{s_i} \leq \beta \right\}.$$

Using this relevant sample subset, we carry out our kernel calculation, as described in Section III-B, based on the two-dimensional (2-D) $(r, g)$ color space. The parameters $\alpha$ and $\beta$ are fixed over all the image. Fig. 5 shows the detection results for an indoor scene using both the $(R, G, B)$ color space and the $(r, g)$ color space after using the lightness variable $s$ to

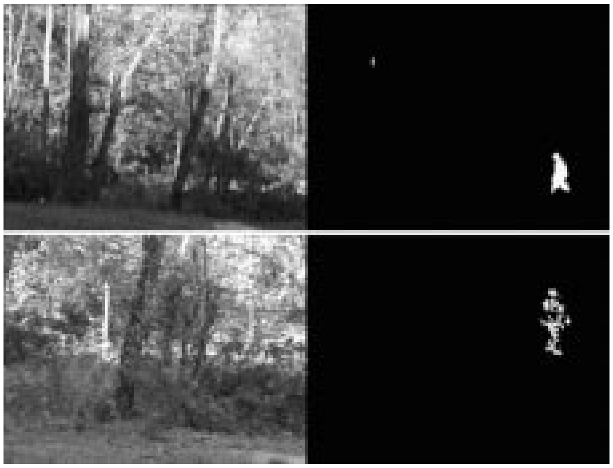

**Fig. 6.** Example of detection results.

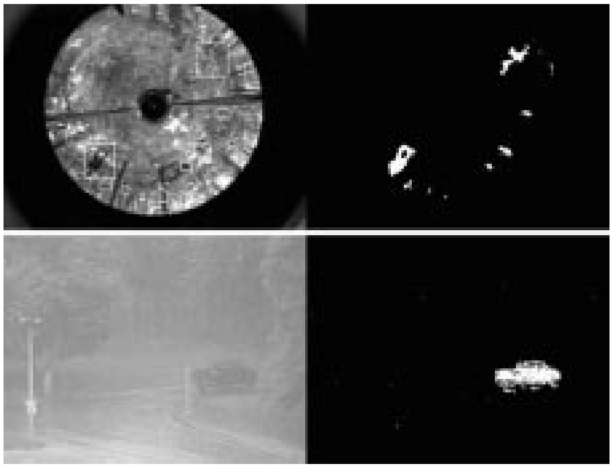

**Fig. 7.** Top: detection result from an omnidirectional camera. Bottom: detection result for a rainy day.

restrict the sample set to relevant values only. We illustrate the algorithm on an indoor sequence because the effect of shadows is more severe than in outdoor environments. The target in the figure wears black pants and the background is gray, so there is no color information. However, we still detect the target very well and suppress the shadows as seen in the rightmost parts of the figure.

*4) Example Detection Results:* The technique has been tested for a wide variety of challenging background subtraction problems in a variety of setups and was found to be robust and adaptive. In this section, we show some more example results. Fig. 6 shows two detection results for targets in a wooded area where the tree branches move heavily and the target is highly occluded. The technique is pixel-based and can work directly with raw images provided by omnidirection cameras [25]. Fig. 7 (top) shows the detection results using an omnidirectional camera. The targets are camouflaged and walking through the woods. Fig. 7 (bottom) shows the detection result for a rainy day where the background model adapts to account for different rain and lighting conditions.[1]

## IV. Modeling the Foreground

### A. Modeling Color Blobs

Modeling the color distribution of a homogeneous region has a variety of applications for object tracking and recognition. The color distribution of an object represents a feature that is robust to partial occlusion, scaling, and object deformation. It is also relatively stable under rotation in depth in certain applications. Therefore, color distributions have been used successfully to track nonrigid bodies [5], [26]–[28], e.g., for tracking heads [29], [28], [30], [27], hands [31], and other body parts against cluttered backgrounds from stationary or moving platforms. Color distributions have also been used for object recognition.

A variety of parametric and nonparametric statistical techniques have been used to model the color distribution of homogeneous colored regions. In [5], the color distribution of a region (blob) was modeled using a single Gaussian in the three-dimensional (3-D) *YUV* space. The use of a single Gaussian to model the color of a blob restricts it to be of a single color which is not a sufficiently general assumption to model regions with mixtures of colors. For example, people's clothing and surfaces with texture usually contain patterns and mixtures of colors. Fitting a mixture of Gaussians using the EM algorithm provides a way to model color blobs with a mixture of colors. This technique was used in [30] and [27] for color-based tracking of a single blob and was applied to tracking faces. The mixture of Gaussian techniques faces the problem of choosing the right number of Gaussians for the assumed model (model selection). Nonparametric techniques using histograms have been widely used for modeling the color of objects for different applications to overcome the previously mentioned problems with parametric models. Color histograms have been used in [32] for people tracking. Color histograms have also been used in [31] for tracking hands, in [26] for color region tracking and in [33] for skin detection. The major drawback with color histograms is the lack of convergence to the right density function if the data set is small. Another major drawback with histograms, in general, is that they are not suitable for higher dimensional features.

Given a sample $S = \{x_i\}$ taken from an image region, where $i = 1 \ldots N$ and $x_i$ is a $d$-dimensional vector representing the color, we can estimate the density function at any point $y$ of the color space directly from $S$ using the product of one-dimensional (1-D) kernels [1] as

$$\hat{P}(y) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} K_{\sigma_j}(y_j - x_{ij}) \qquad (7)$$

where the same kernel function is used in each dimension with a different bandwidth $\sigma_j$ for each dimension of the color space. Usually in color modeling 2-D or 3-D color spaces are used. Two-dimensional chromaticity spaces, e.g., $r = R/(R+G+B), g = G/(R+G+B)$ and $a, b$ from the *Lab* color space, are used when it is desired to make the model invariant to illumination geometry for reasons discussed in Section III-B3. Three-dimensional color spaces are widely

[1] Video clips showing these results and others can be downloaded from ftp://www.umiacs.umd.edu/pub/elgammal/video/index.htm

used because of their better discrimination since brightness information is preserved. The use of different bandwidths for kernels in different color dimensions is desirable since the variances in each color dimension are different. For example, the luminance variable usually has more variance than the chromaticity variables, and therefore wider kernels should be used in that dimension.

Using kernel density estimation for color modeling has many motivations. Unlike histograms, even with a small number of samples, kernel density estimation leads to a smooth, continuous, and differentiable density estimate. Since kernel density estimation does not assume any specific underlying distribution and the estimate can converge to any density shape with enough samples, this approach is suitable to model the color distribution of regions with patterns and mixture of colors. If the underlying distribution is a mixture of Gaussians, kernel density estimation converges to the right density with a small number of samples. Unlike parametric fitting of a mixture of Gaussians, kernel density estimation is a more general approach that does not require the selection of the number of Gaussians to be fitted. One other important advantage of using kernel density estimation is that the adaptation of the model is trivial and can be achieved by adding new samples. Since color spaces are low in dimensionality, efficient computation of kernel density estimation for color pdfs can be achieved using the Fast Gauss Transform algorithm [34], [35].

### B. Color-Based Body Part Segmentation

In this section, we use the color modeling approach described in Section IV-A to segment foreground regions, corresponding to tracked people in upright poses, into major body parts. The foreground regions are detected using the background subtraction technique described earlier. People can be dressed in many different ways but generally are dressed in a way that leads to a set of major color regions aligned vertically for people in upright poses (e.g., shirt, T-shirt, jacket on the top and pants, shorts, skirts on the bottom). We consider the case where people are dressed in a top–bottom manner which yields a segmentation of the person into a head, torso, and bottom. Generally, a person in an upright pose is modeled as a set of vertically aligned blobs $M = \{A_i\}$ where a blob $A_i$ models a major color region along the vertical axis of the person representing a major part of the body as the torso, bottom, or head. Each blob is represented by its color distribution as well as its spatial location with respect to the whole body. Since each blob has the same color distribution everywhere inside the blob, and since the vertical location of the blob is independent of the horizontal axis, the joint distribution of pixel $(x, y, c)$ (the probability of observing color $c$ at location $(x, y)$ given blob $A$) is a multiplication of three independent density functions

$$P_A(x, y, c) = f_A(x) g_A(y) h_A(c)$$

where $h_A(c)$ is the color density of blob $A$ and the densities $g_A(y), f_A(x)$ represent the vertical and horizontal location of the blob, respectively.

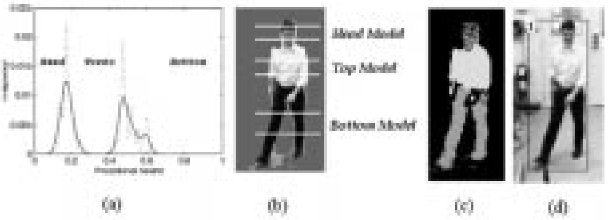

**Fig. 8.** (a) Blob separator histogram from training data. (b) Confidence bands. (c) Blob segmentation. (d) Detected blob separators.

Estimates for the color density $h_A(c)$ can be calculated using kernel density estimation. We represent the color of each pixel as a 3-D vector $X = (r, g, s)$ where $r = R/(R+G+B), g = G/(R+G+B)$ are two chromaticity variables and $s = (R+G+B)/3$ is a lightness variable. The three variables are scaled to be in the range 0 to 1. Given a sample of pixels $S_A = \{X_i = (r_i, g_i, s_i)\}$ from blob $A$, an estimate $\hat{h}_A(\cdot)$ for the color density $h_A(\cdot)$ can be calculated as

$$\hat{h}_A(r, g, s) = \frac{1}{N} \sum_{i=1}^{N} K_{\sigma_r}(r - r_i) K_{\sigma_g}(g - g_i) K_{\sigma_s}(s - s_i).$$

Given a set of samples $S = \{S_{A_i}\}$ corresponding to each blob, and initial estimates for the position of each blob $y_{A_i}$, each pixel is classified into one of the three blobs based on maximum-likelihood classification assuming that all blobs have the same prior probabilities

$$\begin{aligned} X \in A_k \text{ s.t. } k &= \arg_k \max P(X \mid A_k) \\ &= \arg_k \max g_{A_k}(y) h_{A_k}(c) \qquad (8) \end{aligned}$$

where the vertical density $g_{A_k}(y)$ is assumed to have a Gaussian distribution $g_{A_k}(y) = N(y_{A_k}, \sigma_{A_k})$. Since the blobs are assumed to be vertically above each other, the horizontal density $f_A(x)$ is irrelevant to the classification.

A horizontal blob separator is detected between each two consecutive blobs by finding the horizontal line that minimizes the classification error. Given the detected blob separators, the color model is recaptured by sampling pixels from each blob. Blob segmentation is performed, and blob separators are detected in each new frame as long as the target is isolated and tracked. Adaptation of the color model is achieved by updating the sample (adding new samples and ignoring old samples) for each blob model.

Model initialization is done automatically by taking three samples $S = \{S_H, S_T, S_B\}$ of pixels from three confidence bands corresponding to the head, torso, and bottom. The locations of these confidence bands are learned offline as follows. A set of training data with different people in upright pose (from both genders and in different orientations) is used to learn the location of blob separators (head-torso, torso-bottom) with respect to the body where these separators are manually marked. Fig. 8(a) shows a histogram of the locations of head-torso (left peak) and torso-bottom (right peak) in the training data. Based on these separator location estimates, we can determine the confidence bands proportional to the height where we are confident that they belong to the

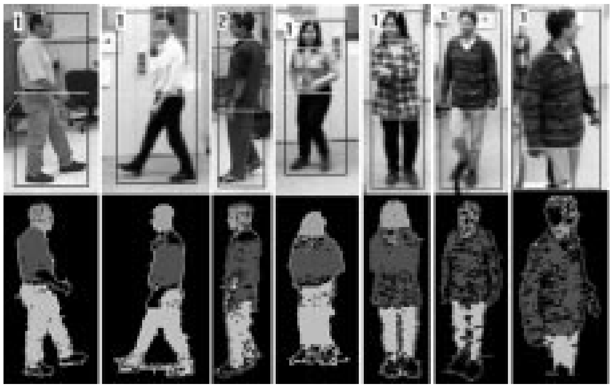

**Fig. 9.** Example results for blob segmentation.

head, torso, or bottom and use them to capture initial samples $S = \{S_H, S_T, S_B\}$. Fig. 8(b) shows initial bands used for initialization where the segmentation result is shown in 8(c), and the detected separators are shown in 8(d).

Fig. 9 illustrates some blob segmentation examples for various people. The segmentation and separator detection is robust even under partial occlusion of the target as in the rightmost result. Also, in some of these examples, the clothes are not of a uniform color.

### *C. Segmentation of Multiple People*

Visual surveillance systems are required to keep track of targets as they move through the scene even when they are occluded by or interacting with other people in the scene. It is highly undesirable to lose track of the targets when they are in a group. It is even more important to track the targets when they are interacting than when they are isolated. This problem is important not only for visual surveillance but also for other video analysis applications such as video indexing and video archival and retrieval.

In this section, we show how to segment foreground regions corresponding to a group of people into individuals given the representation for isolated people presented in Section IV-B. One drawback of this representation is its inability to model highly articulated parts such as hands. However, since our main objective is to segment people under occlusion, we are principally concerned with the mass of the body. Correctly locating the major blobs of the body will provide constraints on the location of the hands which could then be used to locate and segment them. The assumption we make about the scenario is that the targets are visually isolated before occlusion so that we can initialize their models.

Given a foreground region corresponding to a group of people, we search for the arrangement that maximizes the likelihood of the appearance of this region given the models that we have built for the individuals. As a result, we obtain a segmentation of the region. The segmentation result is then used to determine the relative depth of each individual by evaluating different hypothesis about the arrangement of the people. This allows us to construct a model for occlusion.

The problem of tracking groups of people has been addressed recently in the literature. The Hydra system [36] tracks people in groups by tracking their heads based on the silhouette of the foreground regions corresponding to the group. It is able to count the number of people in the groups as long as their heads appear as part of the outer silhouette of the group; it fails otherwise. The Hydra system was not intended to accurately segment the group into individuals nor does it recover depth information. In [32], groups of people were segmented based on the individuals' color distribution where the color distribution of the whole person was represented by a histogram. The color features are represented globally and are not spatially localized; therefore, this approach loses spatial information about the color distributions which is an essential discriminant.

*1) Segmentation Using Likelihood Maximization:* For simplicity and without loss of generality, we focus on the the two-person case. Given a person model $M = \{A_i\}$ where $i = 1 : n$, the probability of observing color $c$ at location $x, y$ given blob $A$ is

$$P_A(x, y, c) = f_A(x) g_A(y) h_A(c).$$

Since our blobs are aligned vertically, we can assume that all the blobs share the same horizontal density function $f(x)$. Therefore, given a person model $M = \{A_i\} i = 1 : n$, the probability of $(x, y, c)$ is

$$P(x, y, c \,|\, M) = \frac{f(x)}{C(y)} \sum_{i=1}^{n} g_{A_i}(y) h_{A_i}(c) \tag{9}$$

where $C$ is a normalization factor such that $C(y) = \sum_i g_{A_i}(y)$. The location $x, y$ and the spatial densities $g_{A_i}(y), f(x)$ are defined relative to an origin $o$. If the origin moves to $x_o, y_o$, we can shift the previous probability as

$$P(x, y, c \,|\, M(x_o, y_o)) = \frac{f(x - x_o)}{C(y - y_o)} \sum_{i=1}^{n} g_{A_i}(y - y_o) h_{A_i}(c).$$

This defines the conditional density as a function of the model origin $(x_o, y_o)$, i.e., $(x_o, y_o)$ is a parameter for the density, and it is the only degree of freedom allowed.

Given two people occluding each other with models $M_1(x_1, y_1)$ and $M_2(x_2, y_2)$, $h = (x_1, y_1, x_2, y_2)$ is a four-dimensional (4-D) hypothesis for their origins. We will call $h$ an arrangement hypothesis. For a foreground region $X = (X_1, \ldots, X_m)$ representing those two people, each foreground pixel $X_i = (x_i, y_i, c_i)$ can be classified into one of the two classes using maximum-likelihood classification (assuming the same prior probability for each person). This defines a segmentation $\omega_h(X) = (\omega_h(X_1), \ldots \omega_h(X_m))$ that minimizes Bayes error, where

$$\omega(X_i) = k \text{ s.t.} k = \arg_k \max P(X_i | M_k(x_k, y_k)), \quad k = 1, 2.$$

Notice that the segmentation $\omega_h(X)$ is a function of the origin hypothesis $h$ for the two models, i.e., each choice for the targets' origins defines a different segmentation of the foreground region. The best choice for the targets' origins is the one that maximizes the likelihood of the data over the entire foreground region. Therefore, the optimal choice for $h$ can be defined in terms of a log-likelihood function

$$h_{\rm opt} = \arg_h \max \sum_{i=1}^{m} \log P(X_i \,|\, M_k(h)).$$

For each new frame at time $t$, searching for the optimal $(x_1, y_1, x_2, y_2)_t$ solves both the foreground segmentation as well as person tracking problems simultaneously. This formalization extends in a straightforward way to the case of $N$ people in a group. In this case, we have $N$ differerent classes and an arrangement hypothesis is a $2N$-dimensional vector $h = (x_1, y_1, \ldots, x_N, y_N)$.

Finding the optimal hypothesis for $N$ people is a search problem in $2N$ dimension space, and an exhaustive search for this solution would require $O(w^{2N})$ tests, where $w$ is a 1-D window for each parameter (i.e., the diameter of the search region in pixels). Thus, finding the optimal solution in this way is exponential in the number of people in the group, which is impractical. Instead, since we are tracking the targets and the targets are not expected to move much between consecutive frames, we can develop a practical solution based on direct detection of an approximate solution $\hat{h}^t$ at frame $t$ given the solution $\hat{h}^{t-1}$ at frame $t-1$. Let us choose a model origin that is expected to be visible throughout the occlusion and can be detected in a robust way. For example, if we assume that the tops of the heads are visible throughout the occlusion, we can use these as origins for the spatial densities. Moreover, the top of the head is a shape feature that can be detected robustly given our segmentation. Given the model origin location $\hat{h}^{t-1} = (x_i, y_i)^{t-1}$ at frame $t-1$, we can use this origin to classify each foreground pixel $X$ at frame $t$ using the maximum likelihood of $P(X \,|\, M(x_i, y_i)_{t-1})$. Since the targets are not expected to have significant translations between frames, we expect that the segmentation based on $(x_i, y_i)_{t-1}$ would be good in frame $t$, except possibly at the boundaries. Using this segmentation, we can detect new origin locations (top of the head), i.e., $(x_i, y_i)_t$. We can summarize this in the following steps.

1) $h_o^t \leftarrow \hat{h}^{t-1} = (x_1, y_1, \ldots, x_N, y_N)^{t-1}$.
2) *Segmentation*: Classify each foreground pixel $X$ based on $P(X \,|\, M_k(x_k, y_k))$.
3) *Detection*: Detect new origins (top of heads) $\rightarrow \hat{h}^t$.

*2) Modeling Occlusion:* By occlusion modeling, we mean assigning a relative depth to each person in the group based on the segmentation result. Several approaches have been suggested in the literature to solve this problem. In [37], a ground plane constraint was used to reason about occlusion between cars. The assumption that object motion is constrained to the ground plane is valid for people and cars but would fail if the contact point on the ground plane is not visible because of partial occlusion by other objects or because contact points are out of the field of view (for example, see Fig. 10). In [32], the visibility index was defined to be the ratio between the number of pixels visible for each person during occlusion to the expected number of pixels for that person when isolated. This visibility index was used to measure the depth (higher visibility index indicates that the person is in front). While this can be used to identify the person in front, this approach does not generalize to more than two people. The solution we present here does not use the ground plane constraint and generalizes to the case of $N$ people in a group.

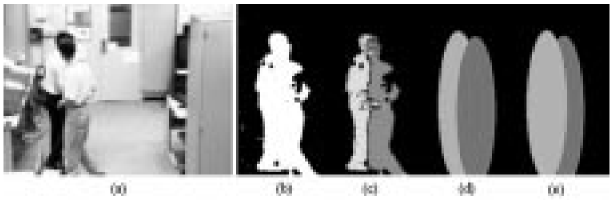

**Fig. 10.** (a) Original image. (b) Foreground region. (c) Segmentation result. (d), (e) Occlusion model hypotheses.

Given a hypothesis $h$ about the 3-D arrangement of people along with their projected locations in the image plane and a model of their shape, we can construct an occlusion model $O_h(x)$ that maps each pixel $x$ to one of the tracked targets or the scene background. Let us consider the case of two targets as shown in Fig. 10. The foreground region is segmented as in Section IV-C1, which yields a labeling $\omega(x)$ for each pixel [Fig. 10(c)] as well as the most probable location for the model origins. There are two possible hypotheses about the depth arrangement of these two people, and the corresponding occlusion models are shown in Fig. 10(d) and (e), assuming an ellipse as a shape model for the targets. We can evaluate these two hypotheses (or generally $N$ hypotheses) by minimizing the error in the labeling between $O_h(x)$ and $\omega(x)$ over the foreground pixels, i.e.,

$$\text{error}(h) = \sum_{x \in \text{FG}} (1 - \delta(O_h(x)\omega(x)))$$

for all foreground pixels.[2] We use an ellipse with major and minor axes set to the expected height and width of each person estimated before the occlusion. Figs. 11 and 12 show some examples of the constructed occlusion model for some occlusion situations.

Fig. 11 shows results for segmenting two people in different occlusion situations. The foreground segmentation between the two people is shown as well as part segmentation. Pixels with low likelihood probabilities are not labeled. In most of the cases, hands and feet are not labeled or are misclassified because they are not modeled by the part representation. The constructed occlusion model for each case is also shown. Notice that, in the third and fourth examples, the two people are dressed in similarly colored pants. Therefore, only the torso blobs are discriminating in color. This was sufficient to locate each person's spatial model parameters and therefore similarly colored blobs (head and bottom) were segmented correctly based mainly on their spatial densities. Still, some misclassification can be noticed around the boundaries between the two pants, which is very hard even for a human to segment accurately. Fig. 12 illustrates several frames from

[2] In the two-person case, an efficient implementation for this error formula can be achieved by considering only the intersection region and finding the target which appears most in this region as being the one in front.

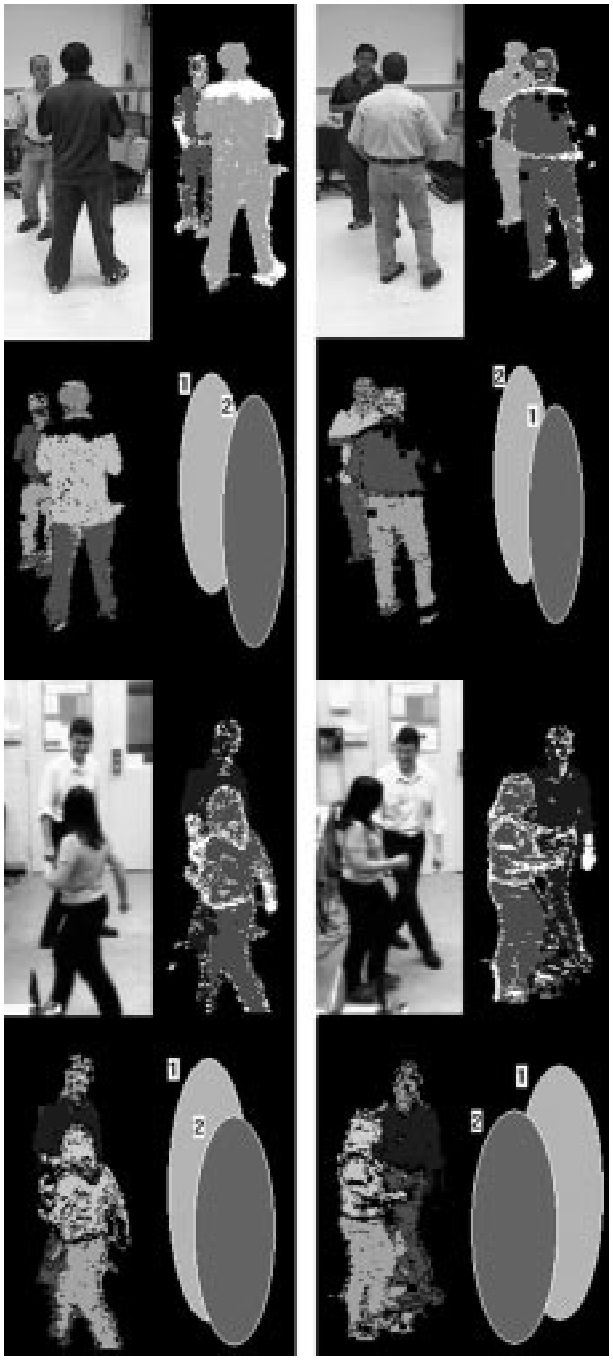

**Fig. 11.** Example results. Top left: original image. Top right: people segmentation. Bottom left: blob segmentation. Bottom right: constructed occlusion model.

a sequence for two targets being tracked throughout occlusion. The part segmentation results are shown as well as the constructed occlusion model. More details and more experimental results can be found in [38].

## V. Conclusion

In this paper, we presented nonparametric kernel density estimation techniques as a tool for constructing statistical representations for the scene background and foreground regions in video surveillance. Since the pdf associated with the background or the foreground does not necessarily follow a known parametric form, kernel estimation methods are a more suitable approach to use in these applications.

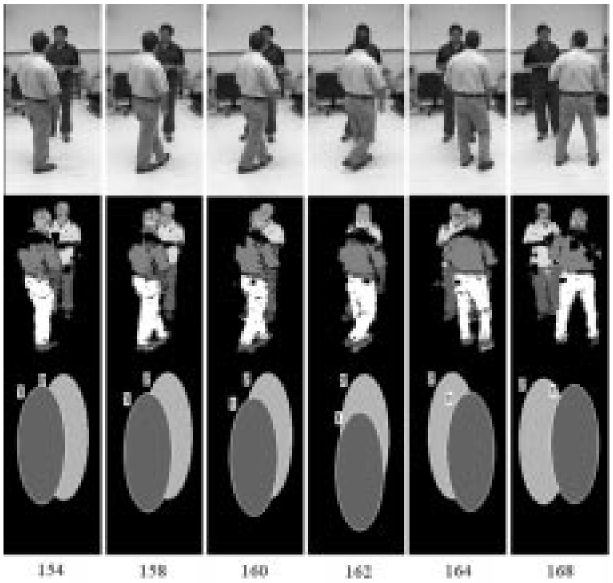

**Fig. 12.** Example results. Top: original image. Middle: blob segmentation. Bottom: occlusion model.

A background model and background subtraction technique was introduced. The model is based on estimating the pdf of pixel intensity directly from a set of recent intensity values. The model achieves sensitive detection of moving targets against cluttered backgrounds. The model can handle situations where the scene background is not completely static but contains small motions such as moving tree branches and bushes. The model is also adaptive to changes in the scene illumination. The model is able to suppress false detections that arise due to small camera displacements. We also showed how the model can use color information to suppress detection of the shadows of the targets.

We also used kernel estimation techniques for modeling the appearance of foreground regions. We showed that this technique is a general approach for modeling homogeneous color regions. We introduced a representation of people that spatially localizes color properties in a way that corresponds to their clothing. Based on this representation, we presented a general probabilistic framework that uses maximum-likelihood estimation to estimate the best arrangement for people in a group in order to segment the foreground regions corresponding to this group. A method to reason about occlusion was presented. The method constructs and maintains a model of the occlusion that is utilized in the same segmentation framework.

## References

[1] D. W. Scott, *Mulivariate Density Estimation*. New York: Wiley-Interscience, 1992.

[2] R. O. Duda, D. G. Stork, and P. E. Hart, *Pattern Classification*. New York: Wiley, 2000.

[3] C. Lambert, S. Harrington, C. Harvey, and A. Glodjo, “Efficient on-line nonparametric kernel density estimation,” *Algorithmica*, no. 25, pp. 37–57, 1999.

[4] A. Elgammal, R. Duraiswami, and L. S. Davis, “Efficient computation of kernel density estimation using fast gauss transform with applications for segmentation and tracking,” *Proc. IEEE 2nd Int. Workshop Statistical and Computational Theories of Vision*, July 2001.

[5] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of human body,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, July 1997.

[6] K.-P. Karmann and A. von Brandt, “Moving object recognition using and adaptive background memory,” in *Time-Varying Image Processing and Moving Object Recognition*. Amsterdam, The Netherlands: Elsevier, 1990.

[7] K.-P. Karmann, A. V. Brandt, and R. Gerl, “Moving object segmentation based on adabtive reference images,” in *Signal Processing V: Theories and Application*. Amsterdam, The Netherlands: Elsevier, 1990.

[8] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, “Toward robust automatic traffic scene analyis in real-time,” in *Proc. Int. Conf. Pattern Recognition*, 1994, pp. 126–131.

[9] W. E. L. Grimson, C. Stauffer, and R. Romano, “Using adaptive tracking to classify and monitor activities in a site,” in *IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 22–29.

[10] W. E. L. Grimson and C. Stauffer, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 1999, pp. 22–29.

[11] N. Friedman and S. Russell, “Image segmentation in video sequences: A probabilistic approach,” presented at the 13th Conf. Uncertainty in Artificial Intelligence, Providence, RI, 1997.

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

[13] J. Rittscher, J. Kato, S. Joga, and A. Blake, “A probabilistic background model for tracking,” in *Proc. 6th Eur. Conf. Computer Vision*, vol. 2, 2000, pp. 336–350.

[14] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Bouhman, “Topology free hidden markov models: Application to background modeling,” in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 294–301.

[15] Y.-H. Yang and M. D. Levine, “The background primal sketch: An approach for tracking moving objects,” *Machine Vision Applicat.*, vol. 5, pp. 17–34, 1992.

[16] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, “Detection and location of people in video images using adaptive fusion of color and edge information,” presented at the Int. Conf. Pattern Recognition, Barcelona, Spain, 2000.

[17] Y. Hsu, H. H. Nagel, and G. Rekers, “New likelihood test methods for change detection in image sequences,” *Comput. Vision Image Process.*, vol. 26, pp. 73–106, 1984.

[18] T. Matsuyama, T. Ohya, and H. Habe, “Background subtraction for nonstationary scenes,” in *Proc. 4th Asian Conf. Computer Vision*, 2000, pp. 662–667.

[19] A. Mittaland and D. Huttenlocher, “Scene modeling for wide area surveillance and image synthesis,” presented at the IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 160–167.

[20] T. Wada and T. Matsuyama, “Appearance sphere: Background model for pan-tilt-zoom camera,” presented at the 13th Int. Conf. Pattern Recognition, Vienna, Austria, 1996.

[21] A. Elgammal, D. Harwood, and L. S. Davis, “Nonparametric background model for background subtraction,” in *Proc. 6th Eur. Conf. Computer Vision*, vol. 2, 2000, pp. 751–767.

[22] M. D. Levine, *Vision in Man and Machine*. New York: McGraw-Hill, 1985.

[23] E. L. Hall, *Computer Image Processing and Recognition*. New York: Academic, 1979.

[24] T. Horprasert. D. Harwood and L. S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,” presented at the IEEE Frame-Rate Applications Workshop, Kerkyra, Greece, 1999.

[25] S. Nayar, “Omnidirectional video camera,” in *Proc. DARPA Image Understanding Workshop*, 1997, pp. 235–241.

[26] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, June 2000, pp. 142–149.

[27] Y. Raja, S. J. Mckenna, and S. Gong, “Tracking color objects using adaptive mixture models,” *Image Vision Comput.*, no. 17, pp. 225–231, 1999.

[28] P. Fieguth and D. Terzopoulos, “Color-based tracking of heads and other objects at video frame rates,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 21–27, June 1997.

[29] S. Birchfield, “Elliptical head tracking using intensity gradients and color histograms,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 232–237, June 1998.

[30] Y. Raja, S. J. Mckenna, and S. Gong, “Color model selection and adaptation in dynamic scenes,” in *Proc. 5th Eur. Conf. Computer Vision*, 1998, pp. 460–474.

[31] J. Martin, V. Devin, and J. Crowley, “Active hand tracking,” in *Proc. 3rd IEEE Int. Conf. Automatic Face and Gesture Recognition*, 1998, pp. 573–578.

[32] S. J. McKenna, S. Jabri, Z. Duric, and A. Rosenfeld, “Tracking groups of people,” *Comput. Vision Image Understanding*, no. 80, pp. 42–56, 2000.

[33] M. J. Jones and J. M. Rehg, “Statistical color models with application to skin detection,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 274–280, 1999.

[34] L. Greengard and J. Strain, “The fast gauss transform,” *SIAM J. Sci. Comput.*, vol. 2, pp. 79–94, 1991.

[35] A. Elgammal, R. Duraiswami, and L. S. Davis, “Efficient nonparametric adaptive color modeling using fast gauss transform,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 563–570, Dec. 2001.

[36] I. Haritaoglu, D. Harwood, and L. S. Davis, “Hydra: Multiple people detection and tracking using silhouettes,” in *Proc. IEEE Int. Workshop Visual Surveillance*, 1999, pp. 6–14.

[37] D. Koller, J. Weber, and J. Malik, “Robust multiple car tracking with occlusion reasoning,” in *Proc. Eur. Conf. Computer Vision*, vol. 1, 1994, pp. 189–196.

[38] A. Elgammal and L. S. Davis, “Probabilistic framework for segmenting people under occlusion,” *Proc. IEEE 8th Int. Conf. Computer Vision*, vol. 2, pp. 145–152, 2001.

**Ahmed Elgammal** received the B.Sc. degree and the M.Sc. degree in computer science and automatic control from the University of Alexandria, Alexandria Egypt, in 1993 and 1996, respectively, and the M.Sc. and Ph.D degrees in computer science from the University of Maryland, College Park, in 2000 and 2002, respectively.

He has been an Assistant Research Faculty Member at the Computer Vision Laboratory, University of Maryland Institute for Advanced Computer Studies, since 2001. His research interest includes computer vision, graphics, and multimedia computing.

**Ramani Duraiswami** (Member, IEEE) was born in Madras, India, in 1963. He received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1985 and the Ph.D. degree from The Johns Hopkins University, Laurel Park, in 1991.

He is a Research Scientist at the Institute for Advanced Computer Studies, University of Maryland, College Park. He recently helped establish the Perceptual Interfaces and Reality Laboratory at the Institute for Multidisciplinary Research in Perceptual Interfaces and Virtual Reality. He has broad research interests in the areas of virtual reality, computer vision, scientific computing, modeling human audition, computational acoustics, applied mathematics, and fluid dynamics.

**David Harwood** is a graduate of the University of Texas at Austin and the Massachusetts Institute of Technology, Cambridge.

He is a Member of the University of Maryland Institute for Advanced Computer Studies, College Park. He is the author of numerous publications on computer image analysis. His recent work has focused on real-time video analysis for surveillance.

**Larry S. Davis** (Fellow, IEEE) received the B.A. degree from Colgate University, Hamilton, NY, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of Maryland, College Park, in 1974 and 1976, respectively.

From 1977 to 1981, he was an Assistant Professor in the Department of Computer Science at the University of Texas, Austin. He returned to the University of Maryland as an Associate Professor in 1981. From 1985 to 1994, he was the Director of the University of Maryland Institute for Advanced Computer Studies. He is currently a Professor in the Institute and the Computer Science Department, as well as Chair of the Computer Science Department. He is known for his research in computer vision and high-performance computing. He has published over 75 papers in journals and has supervised over 12 Ph.D. students. He is an Associate Editor of the *International Journal of Computer Vision* and an Area Editor for *Computer Models for Image Processor: Image Understanding.*

Dr. Davis has served as program or general chair for most of the field's major conferences and workshops, including the Fifth International Conference on Computer Vision, the field's leading international conference.