Scene Modelling Using An Adaptive Mixture of Gaussians in Colour and Space

Patrick Dickinson   Andrew Hunter

Department of Computing and Informatics
University of Lincoln
Lincoln LN6 7TS, UK

Abstract

We present an integrated pixel segmentation and region tracking algorithm, designed for indoor environments. Visual monitoring systems often use frame differencing techniques to independently classify each image pixel as either foreground or background. Typically, this level of processing does not take account of the global image structure, resulting in frequent misclassification. We use an adaptive Gaussian mixture model in colour and space to represent background and foreground regions of the scene. This model is used to probabilistically classify observed pixel values, incorporating the global scene structure into pixel-level segmentation. We evaluate our system over 4 sequences and show that it successfully segments foreground pixels and tracks major foreground regions as they move through the scene.

1. Introduction

The challenge of intelligent video surveillance is to extract meaningful structure from a stream of raw image pixel data. A popular approach is to apply a "bottom-up" sequence of processing steps to each frame image. Each step represents an incremental abstraction of the image data, resulting in a high-level and context specific interpretation of the scene dynamics. Typically, the first level of processing employs a per-pixel background subtraction technique to classify each pixel as either foreground or background. Many methods have been proposed. Wren's Pfinder system [14], and Zhao [15] use an adaptive Gaussian to represent each pixel's background colour value. Subsequent values are classified by thresholding against this distribution, and adaption of the model copes with slow lighting changes. Elgammal [2] uses a non-parametric kernel density estimate instead of a Gaussian model. Stauffer [12] introduced a per-pixel mixture of Gaussians: the popularity of this model lies in its ability to represent multiple processes for each pixel. Many variations and modifications have been developed, such as [5, 6]. A common feature of these schemes is that pixels are treated as spatially independent processes.

Spatial correlation of foreground pixels is typically developed at the second processing stage, through the clustering of foreground pixels into homogeneous regions. Such systems have also been well studied and explored. The Pfinder system [14] uses Gaussian distributions in colour and space, and explicitly assigns each to a specific body part. Khan [7] uses a similar approach, and further extends the feature space to include optical flow values [8]. Non-probabilistic schemes have also been used successfully, such as connected components algorithms [9, 15].

The work presented in this paper is motivated by the fact that background subtraction techniques treat each pixel as an independent process: classification is based only on previous observation of that pixel's value. Failure to interpret observations within the context of the higher-level image structure leads to frequent and obvious misclassifications, for example when a background object, or the camera, move slightly. Some authors have addressed this issue. Stauffer's [12] model can eliminate misclassification due to cyclical motion in the background (such as moving foliage), but not sporadic movements. Elgammal [2] thresholds against adjacent pixel distributions, to eliminate small movements. However, a more structural approach is warranted. Harville [4] proposes a general framework in which high-level decisions feedback to background subtraction. Cristani [1] successfully integrates a pixel model with a particle filter tracking system to implement high-level modulation of low-level processing.

Our approach is to dispense with a per-pixel background model. Instead we model homogenous regions of scene pixels using a 5-dimensional mixture of Gaussians in colour and space. Each regional distribution is classified as either foreground or background, and each pixel is assigned to one region. Thus each pixel is classified as foreground or background according to the classification of the distribution to which it is assigned.

We process each new frame by probabilistically assigning each pixel to a distribution. We then re-estimate each distribution's parameters from the statistics of its assigned pixels. The result is that if a distribution represents a region of pixels which moves or changes over time, then the parameters of the distribution are updated to reflect this, and the distribution automatically tracks the region through the scene. If changes in the scene are not well represented by a the model then we add new distributions, which we classify as foreground. The background and foreground distributions are initialised and developed automatically, and require no off-line or supervised pre-processing.

Using this scheme we are able to directly integrate pixel-level segmentation and classification with a dynamic structural representation of the scene. We are currently using this as the basis for developing a visual surveillance system for monitoring individuals in interior environments.

2. Related Work

Statistical region modelling without background subtraction has been used by some researchers in computer vision and video indexing. McKenna [10] developed a tracker in which the colour distribution of a known object is represented as a mixture of 2-dimensional Gaussians. This system does not explicitly model spatial features of objects or regions, and requires the off-line learning of the object model.

Gaussian mixture models have been used in the field of image indexing as a method of extracting salient features from a scene. Greenspan [3] indexes video sequences by adding time to the pixel feature vector. This requires that the video sequence is split into sub-sequences, and a model built for each. A second processing pass uses the model to segment objects in each frame. Though effective, this approach is unsuitable for real-time surveillance applications.

Work by Pece [11] in tracking objects in monochromatic video bears conceptual similarity to ours. Pece models the scene image as a mixture of components in space and greyscale intensity. The likelihood of each pixel is calculated for each cluster, and used to weight the pixel's contribution to the parameter re-estimation. However, the cluster densities are only modelled as Gaussian for the spatial distribution of foreground objects.

3. Our Approach

We represent regions of the scene using a mixture of Gaussian components in 5-dimensional space. The first two dimensions are x and y image coordinates, the remaining three are YUV colour values. A pixel value is represented in this feature space by a vector $\mathbf{X} = [x, y, Y, U, V]^T$. Each component of the mixture is represented by a distribution function of the form:

$$p(\mathbf{X}|\theta_i) = \omega_i \frac{1}{\sqrt{(2\pi)^d |\mathbf{\Sigma}_i|}} \, e^{-\frac{1}{2}(\mathbf{X}-\mu_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{X}-\mu_i)} \quad (1)$$

Where the distribution parameters $\theta_i = \{\omega_i, \mu_i, \mathbf{\Sigma}_i\}$ are the weight ω_i, mean μ_i, and covariance matrix $\mathbf{\Sigma}_i$ of the i^{th} component. The dimensionality, d, is 5. For a mixture of k components, the following condition holds:

$$\sum_{i=1}^{k} \omega_i = 1 \quad (2)$$

Given a set of model parameters of this form, an observed pixel value may be classified by assigning it to the component with the maximum posterior probability, C_{map}. Using the log likelihood of the pixel value:

$$C_{map} = argmax_i \, \{log(p(\mathbf{X}|\theta_i))\} \quad (3)$$

We relax the model slightly by assuming that the spatial and colour distributions of each component are independent and uncorrelated. We may therefore re-express the distribution function in equation (1) as the product of a 2-dimensional spatial Gaussian and a 3-dimensional colour Gaussian, with parameter sets $\theta_i^s = \{\omega_i^s, \mu_i^s, \mathbf{\Sigma}_i^s\}$ and $\theta_i^c = \{\omega_i^c, \mu_i^c, \mathbf{\Sigma}_i^c\}$. Correspondingly, each pixel value is expressed by the spatial vectors $\mathbf{X}^s = [x, y]^T$, and colour vector $\mathbf{X}^c = [Y, U, V]^T$. Hence, equation (3) becomes:

$$C_{map} = argmax_i \, \{log(p(\mathbf{X}^s|\theta_i^s) + log(p(\mathbf{X}^c|\theta_i^c)\} \quad (4)$$

We use each component of our model to represent a homogenous region of the scene: that is, a set of pixels with similar spatial and colour characteristics. Our premise is that such a region is generated by a single corresponding process, such as part of an object. Components are used to represent both foreground and background regions. We define a background region as one which corresponds to a static or marginally varying process. This could be an object or part of an object which is perceived as static or displaying inconsequential movement. A foreground region is one in which the process is not static: the corresponding object exhibits significant movement or change over time. Each component is explicitly classified as either foreground or background. Background components are constructed in an initialisation phase, and foreground components are detected during frame processing. All components are updated during frame processing, to reflect changes in the scene.

The remaining significant data structure in our system is the "support map", which stores the current component assignment for each pixel. During frame processing we use equation (4) to assign each new pixel to one of the current components. If the colour value of the pixel changes significantly it may be assigned to a new component. When this happens, the support map entry for the pixel is changed correspondingly.

3.1. Building the Background

The background components are constructed from the first frame of the sequence. Expectation Maximisation (EM) is an established algorithm for estimating a maximum likelihood set of parameters for a Gaussian mixture, given a model order and data set. Greenspan [3] uses this technique to build parameters for short video sections, and utilises the Minimum Description Length criteria to estimate an appropriate model order. Although this technique is effective and well principled, we face some problems in using it for a real-time surveillance system. Firstly, EM is computationally expensive. We wish to employ a method which initialises quickly to build the background model, and which is fast enough to develop the model on a per-frame basis. EM methods are unsuitable in this respect. Secondly, we need to be able to adapt the model order dynamically. If we consider that the model order is related to the number of underlying processes, then, as objects enter and leave the scene, the model order needs to be re-estimated.

The technique of splitting and merging components has been shown to be effective for controlling convergence of the EM algorithm [13]. It has also been used by McKenna [10] and by Pece [11] as a technique for dynamically adapting model order. We use the iterative splitting and merging of components as the basis for building an initial background model, and subsequently for developing foreground components during frame processing. We find that this method is computationally manageable, and, by minimising the variance of the model components, generates a suitable representation of the scene regions. We build the background model using the following steps:

1 We initialise the mixture with a single component estimated from the statistics of the entire image, and set each support map entry to this component.

2 We iteratively select the components with the highest spatial and colour variances, and split each into two new components.

3 Pairs of similar components are merged.

4 Components which are significantly spatially disconnected (possibly representing different objects or processes) are split.

We now describe these steps in more detail. Starting with an initial component generated by step 1, we split it into a set of components by iteratively applying step 2, as follows. We calculate the principle eigenvalue, λ_i^s and corresponding eigenvector, $\mathbf{\Lambda}_i^s$ for each component's spatial covariance matrix. We select the component $C_{sp} = argmax_i \{\lambda_i^s\}$. If its eigenvalue $\lambda_{sp}^s > T_{sp}^s$, where T_{sp}^s is a predefined threshold, then we split component C_{sp}. We create a new component and re-assign to it those pixels which satisfy:

$$(\mathbf{X}^s - \mu_{sp}^s) \cdot \mathbf{\Lambda}_{sp}^s > 0 \tag{5}$$

This amounts to placing a separating plane through the spatial mean, perpendicular to $\mathbf{\Lambda}_{sp}^s$. The parameters of both components are then re-estimated from the statistics of their respective assigned pixel values as follows:

$$\omega_i^s = \frac{n_i}{N} \tag{6}$$

$$\mu_i^s = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{X}_{ij}^s \tag{7}$$

$$\mathbf{Z}_i^s = \mu_i^{sT} \; \mu_i^s \tag{8}$$

$$\mathbf{\Sigma}_i^s = \frac{\sum_{j=1}^{n_i} \mathbf{X}_{ij}^{s\ T} \mathbf{X}_{ij}^s}{n_i} - \mathbf{Z}_i^s \tag{9}$$

Where $\mathbf{X}_{ij}^s$ is the spatial component of the j^{th} pixel assigned to the i^{th} component; n_i is the number of pixels assigned to the component; and N is the total number of pixels in the image. The value of μ_i^s used in equation (8) is the new value calculated using equation (7). We then apply the same selection and splitting procedure in colour space, using a corresponding threshold T_{sp}^c, to split the component with the highest colour variance. We repeat this process, alternating between spatial and colour distributions, until reaching a maximum number of components, or until the largest eigenvalues fall below their thresholds.

We now merge similar components. If $\mathcal{M}_i^s(\mathbf{X}^s)$ is the spatial Mahalanobis distance of $\mathbf{X}^s$ from μ_i^s, and $\mathcal{M}_i^c(\mathbf{X}^c)$ is the colour Mahalanobis distance of $\mathbf{X}^c$ from μ_i^c, then a pair of components is considered suitable for merging if the following holds:

$$\begin{gathered} \mathcal{M}_1^s(\mu_2^s) < T_{mg}^s \ \wedge \ \mathcal{M}_2^s(\mu_1^s) < T_{mg}^s \ \wedge \\ \mathcal{M}_1^c(\mu_2^c) < T_{mg}^c \ \wedge \ \mathcal{M}_2^c(\mu_1^c) < T_{mg}^c \end{gathered} \tag{10}$$

Where T_{mg}^s and T_{mg}^c are predefined thresholds. We consider each pair of components, and merge the qualifying pair with the lowest value of $max(\mathcal{M}_1^c(\mu_2^c),\ \mathcal{M}_2^c(\mu_1^c))$. This procedure is repeated until no qualifying pairs remain.

We next seek to identify components which represent spatially disconnected regions, and split them to represent those regions separately. We order the components in descending value of λ_i^s, and step through the list. For each, we use a connected components algorithm to determine if it represents two or more disconnected regions of the support map: if so, we split the largest region away from the rest as a new component. For reasons of efficiency we implement this at a reduced resolution. We repeat this until no disconnected components are found, or for a maximum number of iterations. Finally, when this process is complete, components which have a zero or very small weight are culled from the model.

3.2. Frame Processing

Once the background model has been built we are ready to start processing new frames. For each new frame the following operations are applied:

1 The current model is used assign the pixels in the new frame, and rebuild the support map.

2 Existing background and foreground components are updated from the support map.

3 New foreground components are detected and created.

4 Foreground components are merged and culled.

The current mixture of components is used to assign each pixel in the new frame using equation (4). Many systems, such as [12], use a predictive filter to estimate the change in position of objects between frames. Instead, we employ Elgammal's observation [2] that foreground objects, though moving, will have moved only a relatively short distance between one frame and the next. Thus we use only the previous frame's observed model to assign pixels: each pixel is assigned to C_{map}, and its corresponding support map entry set to reflect this. During frame processing a minimum likelihood threshold T_{map} is applied: if $log(p(\mathbf{X}|\theta_{C map})) < T_{map}$, the pixel is set as unassigned.

The parameters of the existing background and foreground components are now re-estimated. For foreground components, equations of the form (6) to (9) are used to rebuild its spatial and colour distributions from its assigned pixels. For background components we adapt the parameters more slowly. For each component i we start by calculating a set of parameter values $\theta_{(i,sm)}$ from the support map, in the same way as for foreground components. However, we do not substitute these new values directly. Given the existing parameters $\theta_{(i,t-1)}$, we calculate the new set $\theta_{(i,t)}$ using an adaptive learning rate:

$$\theta_{(i,t)} = \alpha_i \theta_{(i,sm)} + (1 - \alpha_i)\theta_{(i,t-1)} \tag{11}$$

The learning rate α_i is calculated for each component for each frame as:

$$\alpha_i = \frac{\omega_{(i,sm)}}{\omega_{(i,t-1)}} \quad , \quad \alpha_i \in [0, 1] \tag{12}$$

Where $\omega_{(i,sm)}$ and $\omega_{(i,t-1)}$ are the weights from $\theta_{(i,sm)}$ and $\theta_{(i,t-1)}$ respectively. Constraining the adaptation in this way ensures that if a background component is occluded it does not adapt too quickly to represent only the visible part. It also helps to prevent the background from over adapting to misclassified foreground pixels. It is necessary to renormalise the component weights at this point, to enforce the condition in equation (2).

Next we use the support map to detect new foreground regions. The map is divided into a grid of resolution 16×16 pixels, and the number of unassigned pixels counted for each location. Locations exceeding a threshold density are considered to correspond to new foreground regions. A single foreground component is built from the unassigned pixels in all such grid locations, using equations of the form (6) to (9) to build the spatial and colour distributions. This new component is then split using the same splitting method used to build the background.

Regardless of whether any new components have been added this frame, we test all foreground components for possible merging. First, we restrict the spatial and colour variances of each component to pre-defined maximum values. This helps prevent over adaptation to misclassified background pixels. We then merge similar components using the same pair-wise method as was used for the background model. Finally, we conclude frame processing by culling any foreground components which have a zero or very low weight.

3.3. Modifications to Frame Processing

We have made some modifications to the algorithm in order to improve performance. Firstly we encounter a problem using equation (4) during frame updates: the spatial variance $\mathbf{\Sigma}_i^s$ for large background components is typically very high. This frequently results in pixels being assigned to regions from which they are significantly disconnected. In particular, this hampers detection of new foreground regions. To resolve this we apply the additional restriction that a pixel may only be assigned to a background component if its spatial likelihood exceeds a predefined threshold T_{lik}^s:

$$log(p(\mathbf{X^s}|\theta_i^s)) > T_{lik}^s \tag{13}$$

We also make a performance optimisation to the assignment of pixels during frame processing. If a pixel is currently assigned to an existing component, its colour value remains relatively unchanged, and its likelihood given the same assignment is greater than T_{map} then we leave its assignment unchanged. This significantly reduces processing time. A pixel value is defined as unchanged if each element of its YUV colour value is within a threshold deviation from the value first used to assign it.

4. Experiments

We have tested our system on 4 short sequences, each comprising 150-200 frames, and recorded using a DV camcorder in 720×576 PAL format. Each sequence shows a single human figure engaged in some routine activity, such as walking to a chair and sitting down. One sequence was filmed in an exterior environment, and the others are interior. Figure 1 shows an example frame from one of the sequences, together with representations of the background and foreground components, and a binary mask of the foreground pixels. The component representations were constructed by setting each pixel to μ^c of the component $C^s_{map} = argmax_i \{log(p(\mathbf{X^s}|\theta^s_i))\}$. Figure 2 shows some similar example frames from two of the other sequences.

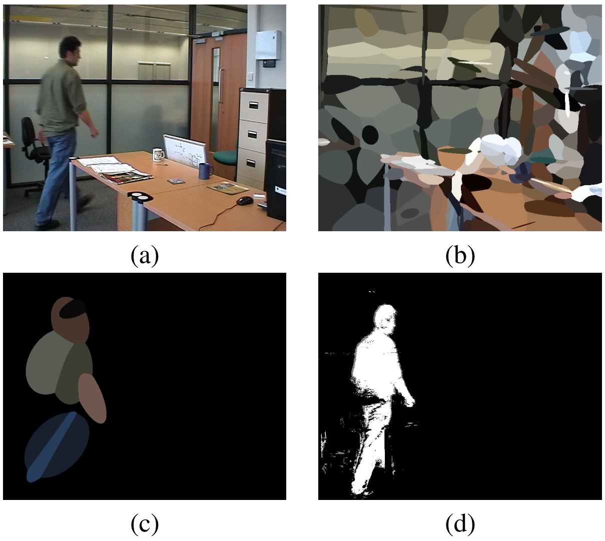

Figure 1: *A frame from an image Sequence. (a) Captured image. (b) Background components. (c) Foreground components. (d) Foreground pixels in the support map.*

The representation shown in figure 1 is typical in that we can infer a correspondence between foreground components and body parts of the subject. We intend to develop our system to use the foreground regions as the basis for a human body tracking system. We therefore evaluated how effectively the foreground components in our system represent and track major body parts.

In all sequences, the major body parts were represented and tracked successfully. In some cases the segmentation failed where part of the foreground coincided with, or was immediately adjacent to, a background region of similar colour. Small body parts, such as hands or feet, were frequently not represented by a corresponding component, or were not tracked between frames due to sudden large displacements.

The system proved robust to camera movement. During one of the sequences the camera was unintentionally moved, resulting in a global image translation. The system continued to operate without noticeable corresponding misclassifications. However, we also noted that the slowed adaptation of the background sometimes results in the introduction of spurious foreground components where there is a significant lighting change.

Each sequence was processed using an Intel Pentium 4 (2.8GHz) PC system, and the executable code was developed in C++. Each sequence took approximately 10 seconds to initialise. Frame processing time ranged between 0.94 and 2.70 seconds.

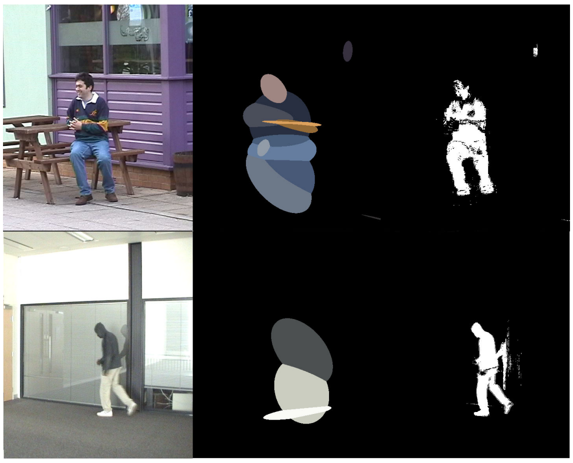

Figure 2: *Example frames from two other sequences.*

We quantify our results in the following tables, which were constructed as follows. For each sequence we identified the major body parts. We inspected the representation in each frame, in an arrangement similar to figure 1. For each body part we categorised the estimated proportion (of its pixels) assigned to a corresponding foreground component as either more than 0.75, 0.75-0.5, less than 0.5, or 0. A classification of 0 corresponds to a failure to represent the body part in that frame. For the whole sequence we then calculated the proportion of frames in which each body part was represented by each category. The sequence tables are ordered from easiest to hardest in terms of perceived difficulty of segmentation. The lower arms are classified separately in the last two sequences, as the subject was wearing short sleeves.

Results for Sequence 1				
	>0.75	>0.50	<0.50	0.00
Head & Face	**0.95**	0.05	0	0
Torso & Arms	**0.92**	0.08	0	0
Legs & Feet	**0.73**	0.19	0.08	0

Results for Sequence 2				
	>0.75	>0.50	<0.50	0.00
Head & Face	0.07	0.24	**0.57**	0.12
Torso & Arms	**0.89**	0.11	0	0
Legs & Feet	**1.00**	0	0	0

Results for Sequence 3				
	>0.75	>0.50	<0.50	0.00
Head & Face	0.27	0.04	**0.60**	0.09
Torso & Arms	**0.52**	0.43	0.05	0
Lower Arms	0.24	0.07	**0.45**	0.24
Legs & Feet	**0.50**	0.46	0.02	0.02

Results for Sequence 4				
	>0.75	>0.50	<0.50	0.00
Head & Face	0.28	**0.37**	0.13	0.22
Torso & Arms	**0.52**	0.43	0.05	0
Lower Arms	**0.69**	0.21	0.10	0
Legs & Feet	0.06	0.02	0.24	**0.68**

5. Conclusions and Further Work

We have presented a novel algorithm which tracks objects by modelling global scene structure, and shown that it is effective. The execution speed of our algorithm is not quite suitable for real-time processing, but we consider that since we have not yet attempted low-level performance optimisation, and that our platform is relatively modest, it is feasible to develop a version which runs in real-time. This will be one objective for further development work.

We are currently developing an indoor surveillance system based on our work. To achieve this we need to improve the on-line adaptation of the model, so that we can identify spurious foreground components and re-classifying them as background. We intend to do this by clustering associated foreground regions to build models of complete objects. We also wish to investigate using the features of foreground components to implement a human body tracking system.

Acknowledgements

This work was supported by an EPSRC CASE studentship in conjunction with Nectar Electronics Ltd, Durham, UK.

References

[1] M. Cristani, M. Bicego, and V. Murino, "Integrated Region- and Pixel-Based Approach to Background Modelling," *Proc. of IEEE Workshop on Motion and Video Computing*, pp. 3-8, 2002.

[2] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and Foreground Modeling Using Non-Parametric Kernel Density Estimation for Visual Surveillance," *Proc. of the IEEE*, Vol. 90, No. 7, pp.1151-1163, July 2002.

[3] H.Greenspan, J. Goldberger and A.Mayer, "Probabilistic Space-Time Video Modelling via Piecewise GMM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 3, pp.384-396, March 2004.

[4] M. Harville. "A framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models," *Proc. of 6th European Conference on Computer Vision*, Vol. 3, pp. 543-560, 2002.

[5] O. Javed, K. Shafique, and M. Shah, "A Hierarchical Approach to Robust Background Subtraction Using Color and Gradient Information", *Proc. of IEEE Workshop on Motion and Video Computing*, pp. 2227, 2002.

[6] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-Time Tracking with Shadow Detection", *2nd European Workshop on Advanced Video-based Surveillance Systems*, 2001.

[7] S. Khan and M. Shah, "Tracking People in the Presence of Occlusion," *Asian Conference on Computer Vision*, 2000.

[8] S. Khan and M. Shah, "Object Based Segmentation of Video Using Color, Motion and Spatial Information," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 746751, 2001.

[9] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking Groups of People," *Computer Vision and Image Understanding*, Vol. 80, No. 1, pp. 4256, October 2000.

[10] Y. Raja, S. J. Mckenna, and S. Gong, "Color Model Selection and Adaptation in Dynamic Scenes," *Proc. 5th European Conference on Computer Vision*, pp. 460474, 1998.

[11] A. Pece, "Generative-Model-Based Tracking by Cluster Analysis of Image Differences," *Robotics and Autonomous Systems*, Vol. 39, Nos. 3-4, pp. 181-194, 2002.

[12] C. Stauffer and W.Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246252, 1999.

[13] N. Ueda, R. Nakano, Z. Ghahramani, and G.E. Hinton, "SMEM algorithm for mixture models," *Neural Computation*, Vol. 12, No. 9, pp. 2109-2128, September 2000.

[14] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 780785, July 1997.

[15] T.Zhao and R.Nevatia, "Tracking Multiple Humans in Complex Situations," *IEEE Trans. on Pattern Analysis and Machine Intelligence* Vol. 26, No. 9, pp. 1208-1221, September 2004.