Tracking People in Presence of Occlusion

Sohaib Khan and Mubarak Shah
Computer Vision Lab
School of Computer Science
University of Central Florida
Orlando, FL 32816, USA
{khan, shah}@cs.ucf.edu

ABSTRACT

Tracking humans is a difficult problem because of the non-rigid nature of the human body, and the frequent occlusion encountered among people. We present a framework to track multiple people in a fixed camera situation. Our framework deals implicitly with occlusion, and is able to correctly label people *during* occlusion. We first segment a person into classes of similar color using the Expectation Maximization algorithm. Then we use a maximum *a posteriori* probability approach to track these classes from frame to frame. The system deals well with partial and complete occlusion. Results are presented for an indoor sequence in an office environment.

Keywords: human motion analysis, tracking in occlusion, maximum likelihood, expectation maximization, Bayesian probability, activity recognition.

1. INTRODUCTION

Tracking moving objects is a key problem in computer vision. Recently there has been a lot of interest in analysis of videos involving humans. Human motion analysis is essential in a wide variety of applications, such as activity recognition, surveillance, man-machine interfaces, content-based retrieval, model-based compression and athletic performance analysis. Tracking humans is different from motion estimation of a rigid body, because the human body is a non-rigid form. Various approaches to human motion analysis can be categorized depending on whether information about human body structure is used, or the tracking is done without using body parts[1]. [1, 10, 7] present excellent reviews.

A popular approach to tracking using a stationary camera is to look for regions of change in the scene. This can be done using consecutive frame differencing [14, 8], or more popularly, by comparing the current frame against a background model [11, 12, 13, 6]. The background difference methods differ from each other in the way the background model is built. In [4] the model is built assuming a normal distribution for the color at each pixel. In [11] it is argued that the intensity at each pixel can be a result of multiple processes, therefore, multiple Gaussian distributions should be fitted to the color values of each pixel to obtain the background model. In [13] a model similar to [4] is built, but gray values are used instead of color. The model in [12] is a simplification of the Gaussian models, where the absolute maximum, minimum and largest consecutive difference values are used. This speeds up computation but might be more sensitive to outliers.

Once change detection is done, either by comparing with a background model or by subtracting consecutive frames, most approaches, e.g. [11, 12, 13] perform a correspondence step to label each of the regions as an object. Other approaches perform explicit tracking, in which each frame is not compared with the background frame. Rather, this differencing may be used only for initialization, and after that, the foreground object is followed from frame to frame [9, 4]. Correspondence based approaches work well for cases with no occlusion, but are unable to decide about object identities during occlusion. Indeed excessive occlusion is observed when objects being tracked are close to the camera, as for example in indoor sequences.

Occlusion is a significant problem in human motion analysis. People tend to walk and interact in groups with other people, thereby increasing the chances that persons will occlude each other completely or partially in images. The probability of observing occlusion can be decreased in general by placing the cameras at a higher angle of elevation from the plane of movement of people. That is, by placing the cameras looking vertically downwards, the chance of one person occluding the other is minimized. Indeed most of the previous work in human tracking either uses this constraint on camera positioning, e.g. [11, 5], or does not deal with occluding cases at all [4, 15].

Limited solutions to the occlusion problem are presented by [9] and [12]. In [9] occlusion from static objects is dealt with, using an occlusion reasoning framework, which maintains multiple hypothesis for occluded regions and keeps eliminating wrong ones as time progresses. However, this approach is demonstrated to be useful in simplistic cases and needs to be explored further in the case of more complicated scenarios. Moreover, it is limited to occlusion by static objects, and may not generalize to the more compli-

cated case of occlusion from non-rigid objects, such as other persons. In [12] statistical features of the two persons before occlusion begins are used to resolve the labels after occlusion has ended, but the system cannot decide about which pixels belong to which person *during* the occlusion event.

For most human activity recognition applications, some sort of solution for occlusion is a must. If the tracking system cannot provide correct labels to persons during occlusion, then the performance of the activity recognition system will be degraded if the average time spent during occlusion is significant compared to the total time. This is indeed the case in a number of practical situations, like for example in office environments [3], where occlusion is frequent and it might not be feasible to put a large number of cameras at vertical angles of elevation. If, however, the tracker is providing correct labels even during occlusion, then the task of the activity recognition module is simplified, since it receives more complete information, in this case.

In this paper, we present a statistical framework for tracking multiple people. We impose no constraints on camera positioning, and most of our sample sequences are taken with cameras roughly at the head and body level, looking parallel to the floor plane. This case may result in maximum occlusion. Our approach is based on the previous work by [4], but differs from it in some important respects. The approach in [4] deals with only single person scenes and does not have the ability to work on sequences containing multiple people. Therefore, the issue of person-to-person occlusion does not arise at all. This limits the scope of application of that method. In our framework, a person is detected and segmented into coherent regions upon entering the scene. Persons are tracked separately and not confused which each other even in the presence of complete occlusion. Thus the advantage of this method is not only that it can handle multiple people, but it also handles person-to-person occlusion well within the same framework.

Our system works in two distinct stages. The first stage is when the scene is empty and there is no person visible. We assume that this is the case in the beginning. The initial set of frames is used to build a background model and each frame is analyzed to detect the first person by looking for significant changes from the background. Once there is at least one person in the scene, we switch to the next stage, where we track people and detect new persons entering the scene. If all persons exit, we will switch back to Stage 1.

2. OVERALL APPROACH

Our approach is based on representing people as a mixture of Gaussians in spatial and color space. Each person is modeled as a set of classes, where each class has a spatial component (x, y) and a color component (Y, U, V). A class is thus represented by a 5-dimentional Gaussian distribution. These classes are tracked from one frame to another using a maximum *a posteriori* probability approach.

We start by building a background model, using a set of frames containing no people. The background model is simply the mean and the covariance of the color values observed at each pixel during training. After the background model is completed, each frame is compared to the model, by computing the Mahalanobis distance of the current color value at each pixel from the background model for that pixel. If significant change is detected, it is concluded that a person has entered the scene. This person is then segmented into a set of classes by fitting a multi-variate Gaussian mixture model, using the EM algorithm. The background model is treated as a separate class.

Once the first person is segmented, we shift to our tracking algorithm. We assign a class label to each pixel in subsequent frames by computing the probability of that pixel belonging to each of the existing classes (and the background class), and picking the maximum probability. After assigning all the pixels their new labels, the class statistics are updated by computing the new means and covariances for each class. This update is done in a 'slow' manner, by using a low-pass filter, so that noisy changes in class statistics can be eliminated.

Detecting the entrance of additional persons is not trivial, because the pixels of the new person are still assigned a label from one of the existing classes. This is so because at each new frame, we just pick the maximum likelihood value at each pixel. To counter this problem, we perform a pseudo-connected component algorithm by fitting 1-D Gaussians to the vertical projection of the image. If the number of Gaussians with high weight is more than the existing number of persons, then this indicates that at least one more person has entered the scene. This new person is similarly segmented into a set of classes. The total set of classes is grown to include the classes of the new person.

This framework handles occlusion implicitly and no additional computation is required. The classes of a person are not deleted in case of occlusion and are therefore still used in the maximum likelihood computation. Thus, when the person partially reappears, the pixels of that person immediately return a higher likelihood value for their own class, compared to any other class. The reason is simply that the class which modeled that part of the person before occlusion still models these pixels better than any other class. The assumption is that the color of the person's body would not have changed significantly during the occlusion process. This assumption seems to work well in most practical cases, but might break down in cases of rotation, significant lighting changes or shadows while the person is occluded, or actual physical color changes, like a change of clothing while occluded! It is worthwhile to point out that these changes do not have that

significant an effect on non-occluding persons, because their class information is being continuously updated. In the case of occlusion, however, this is not the case, and so we may observe a big jump between the existing class means and the reappeared person.

3. INITIALIZATION

Background Model

We use a simple background model consisting of the mean color and covariance of the color values observed during training for each pixel [4]. Thus, for every pixel (x, y) in a set of k frames of the background training sequence, we compute the mean μ and the covariance $\mathbf{\Sigma}$ as

$$\mu = \frac{1}{k}\sum_{k}\mathbf{x} \quad (1)$$

$$\mathbf{\Sigma} = \frac{1}{k-1}\sum_{k}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T \quad (2)$$

where $\mathbf{x}$ represents the pixel color $[Y, U, V]^T$ at location (x, y).

Even though we are using a simple background model, a multiple-distribution model like [11] can be used without any additional complication within our framework. The only difference is that we will have more than one background class, rather than just a single background class for each pixel, as is currently the case.

Scene Change Detection

Once the background model is completed, we look for large changes from the background model in subsequent frames. A large change from the background will indicate that a person has entered the scene. Change at pixel (x, y) is given by the Mahalanobis distance, given by

$$d = (\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) \quad (3)$$

If $d > T$, where T is some appropriate threshold, then that pixel is declared foreground, otherwise it is labeled as background. Finally, if there are a significant number of foreground pixels in an image, then we know that some foreground object, in this case a person, has entered the scene.

Initial Segmentation

Once the first person is detected, we segment the person into regions of similar color. We use the EM algorithm [2] to fit a mixture of 3-dimentional Gaussian distributions to the color distribution of the person, given by:

$$M(\mathbf{x}) = \sum_{i=1}^{k}\frac{\omega_i}{(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}}e^{-\frac{(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)}{2}} \quad (4)$$

where ω_i is the weight assigned to the i^{th} Gaussian and d is the dimension of our space (3 in this case). Each Gaussian fitted by the EM algorithm now represents one of the classes of the full person. After the best fit Gaussians are computed, any two Gaussians with very similar means are merged together to form a single Gaussian. This reduces the number of classes representing the person. Finally, a maximum likelihood computation is done to assign each pixel to its correct class.

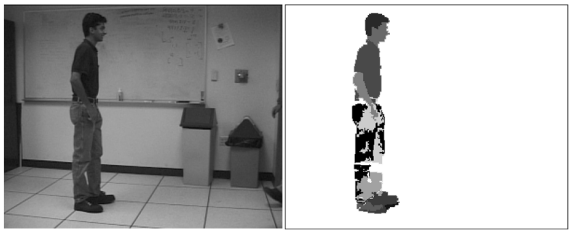

Figure 1. Initial segmentation results: Original frame (left), segmented frame (right)

Given a color image and a mask representing the foreground region (computed using the Mahalanobis distance as above), the mixture of Gaussians is computed by finding the correct parameters ω_i, μ_i, Σ_i for each of the k Gaussians in the mixture model. The number k is a parameter that is an input to the algorithm. The initialization of parameters is done by using equal weights, each as $1/k$. The covariance matrices are all initialized to identity. The means are initialized as small random steps from one of the data points.

For the expectation step, we compute the likelihood of every pixel for each of the Gaussian distributions. For each pixel $\mathbf{x} \in \mathbf{X}$ (where $\mathbf{X}$ is the total set of changed pixels) this likelihood is given by

$$L_i(\mathbf{x}) = \omega_i g[\mu_i, \mathbf{\Sigma}_i](\mathbf{x}) \quad (5)$$

where

$$g[\mu_i, \mathbf{\Sigma}_i](\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}}e^{-\frac{(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)}{2}} \quad (6)$$

and i is the i^{th} Gaussian distribution $1 \leq i \leq k$.

We define S_x as the sum of all k likelihood values at a pixel, and normalize $L_i(\mathbf{x})$ by S_x to get the probabilities $P_i(\mathbf{x})$. This is the probability of the current pixel belonging to a particular Gaussian.

$$P_i(\mathbf{x}) = \frac{L_i(\mathbf{x})}{S_{\mathbf{x}}} = \frac{L_i(\mathbf{x})}{\sum_{j=1}^{k}L_j(\mathbf{x})} \quad (7)$$

As a consequence, we have $\sum_{\mathbf{x}\in\mathbf{X}}\sum_{i=1}^{k}P_i(\mathbf{x}) = |\mathbf{X}|$, i.e. the total number of changed pixels.

For the maximization step, we update the mixture model parameters according to the following set of equations:

$$\omega'_i = \frac{\sum_{\mathbf{x}\in X}P_i(\mathbf{x})}{|\mathbf{X}|} \quad (8)$$

$$\mu'_i = \frac{\sum_{\mathbf{x}\in X}P_i(\mathbf{x})\mathbf{x}}{\sum_{\mathbf{x}\in X}P_i(\mathbf{x})} \quad (9)$$

$$\mathbf{\Sigma}'_i = \frac{1}{\sum_{\mathbf{x}\in X}P_i(\mathbf{x}) - 1}\sum_{\mathbf{x}\in X}P_i(\mathbf{x})\left[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\right] \quad (10)$$

Figure 2. Tracking two persons during occlusion

The process is repeated till the change in means μ and covariances Σ is very small [2]. The convergence time of this algorithm depends on the threshold set for minimum acceptable error. For our purposes, typically, the algorithm converges in two to three iterations, though the convergence time is dependent on initial values, and may take longer for bad initialization.

The next step is to merge regions with very similar means. This step prevents the problem of oversegmentation, while giving us the freedom to choose a high value of k in the previous step to ensure that all colors are represented completely. If two Gaussians have means very close to each other, then they are merged together into one distribution.

The classes created through the above process might be spatially disconnected, because we just fitted a mixture model to the color space. In cases like checkered shirts, we do expect to get spatially disconnected components belonging to the same class. However, if a person's black shoes are put in the same class as black hair, they should be split into separate classes. If we fail to do so, then the spatial covariance of this class would become very high, which in turn will adversely affect the probability computation. To counter this problem we perform connected component labeling on each class and recursively merge components that are closer to each other than a certain threshold. The remaining clusters left in the end are made into separate classes.

Once the mixture model has been computed, the likelihood that each pixel belongs to one of the Gaussians is computed by simply computing the Mahalanobis distance and picking Gaussian which returns the least distance value. This gives us the actual pixels for the region. Figure 1 shows some results of the initial segmentation algorithm.

4. TRACKING

Stage two of our system becomes active when the segmentation the first person is completed. In this stage, we track the persons in the scene and keep looking for new persons entering the scene.

Frame-to-Frame Correspondence

Once we know that at least one person is visible in the camera FOV, we stop looking at the Mahalanobis distance to differentiate foreground from the background. Instead, we compute the likelihood of each new pixel belonging to one of the existing classes, and label it with the class that returns the maximum likelihood value. The motivation for this comes from the Bayesian probability theory. Assume there are $n+1$ classes given by $c_0 ... c_n$, where class c_0 represents the background and classes $c_1 ... c_n$ represent n foreground regions. Each foreground region class is a 5-tuple, containing spatial information (x, y), which is the centroid of the region, and mean color information $[Y, U, V]$. and a 5-by-5 covariance matrix. The color mean and covariance are obtained from the initial segmentation step described in the previous section. The spatial mean and covariance are obtained by just finding the first and second moments according to equations (1) and (2). Notice that the covariance matrix is a block diagonal, with the cross-covariance terms between space and color set to zero. Thus both sets of covariances can be computed separately and combined to form a single 5-by-5 covariance matrix.

For each pixel in the new frame, we compute the log likelihood of it being a member of every class, and assign it to the class that returns the maximum log likelihood value. Thus for every pixel, if we define $\mathbf{x}$ to be the vector $[x, y, Y, U, V]^T$, then the probability that $\mathbf{x}$ belongs to class c_k is given by Bayes theorem:

$$P(c_k|\mathbf{x}) = \frac{P(\mathbf{x}|c_k)P(c_k)}{P(\mathbf{x})} \qquad (11)$$

We label this pixel the class that returns the highest probability value, i.e. the label is given by $\arg\max_k(P(\mathbf{x}|c_k))$. Since we are only interested in comparison, $P(\mathbf{x})$ is just a scale factor and can be ignored. Furthermore, the numerator of equation (11) can be multiplied with any monotonically increasing function without affecting our decision rule. Therefore, the log of the numerator is used as the decision rule because it simplifies computations by converting multiplication to addition.

The term $P(c_k)$ is the *a priori* probability of observing a particular class. One possible way to compute this probability is to observe a long enough data set and count the fraction of time a particular class appears. If we choose to ignore this term, we will imply that all classes are equally likely to occur and no particular class is favored over another. In our work, we observed that giving background class a higher weight than the foreground classes eliminated the problem of shadows significantly. However, it sometimes causes misclassification of foreground as background, if the color of the foreground region is similar to the background model at that location. Thus there is a tradeoff between the extra weight assigned to background and the quality of foreground segmentation.

The problem of finding the correct class for each pixel is now simplified to computing the log likelihood

of a pixel being in all classes, and finding the maximum value:

$$l(x,y) = \arg\max_i (\log P(\mathbf{x}|c_i)) \qquad 0 \le i \le k \qquad (12)$$

where $l(x,y)$ denotes the class to which pixel (x,y) is assigned and k is the number of foreground regions in the previous frame. Note that $P(\mathbf{x}|c_0)$ is computed in a slightly different fashion than the other probabilities, since the spatial component of the background model is fixed and thus has zero variance.

The likelihood terms $P(\mathbf{x}|c_k)$ are computed by assuming that the probability function is given by the multivariate Gaussian distribution given by equation (4). Taking log of (4) and eliminating constant factors which will not affect our decision problem gives:

$$l(x,y) = \arg\max_i \{-(\mathbf{x}-\mu_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x}-\mu_i) - \ln|\mathbf{\Sigma}_i| - d\ln(2\pi)\} \quad 0 \le i \le k \qquad (13)$$

where d is 3 for the c_0 and 5 for all other classes.

Once each pixel has been assigned a label, the mean and the covariance matrix for each of the foreground classes are updated using the new regions computed by Eq.(13). We update the means and the covariances in a 'slow' fashion by using a causal low-pass filter.

$$\mu_{t+1} = \alpha \mathbf{x}_{t+1} + (1-\alpha)\mu_t \qquad (14)$$

where α is a small constant corresponding to the time constant of the filter. Covariance can be recursively updated for computational efficiency by rewriting it as $\mathbf{\Sigma} = E(\mathbf{x}\mathbf{x}^T) - E(\mu\mu^T)$.

Reducing Misclassifications

Misclassifications occur when classes of one person get assigned to pixels of another person. It is important to correct misclassification as soon as possible because their effect keeps accumulating over time. When a few pixels are misclassified, they increase the spatial covariance of their class, thus generating more misclassifications in the next frame. We counter this run-away effect by employing area based filters. If a class acquires some very small disconnected regions far away from its mean, then they are assigned to another class that they are adjacent to. If a class has thin edge-like regions disconnected from its main cluster, then they are removed, by applying an eccentricity test. Finally, if a class has grown such that it forms two or more distinct components that are well separated spatially, then they are split to form more classes. Classes which run out of support, i.e. no pixels are assigned to them, are deleted. However, this deletion is done only in cases where occlusion is not happening. If the system identifies that a person is being occluded, it will retain its classes even though they have no support, because there is a chance they will reappear again later in the sequence. The overall aim is to keep classes spatially compact, with low spatial variance values. This reduces misclassifications, especially in cases where persons are wearing similar colored clothes.

As a final step, we detect when occlusion is not occurring at all by checking if the persons are well separated from each other. In this case, we reclassify any misclassifications by recomputing their likelihood using only the classes of that person. This technique allows us to recover completely from any misclassifications that would have persisted through the occlusion event.

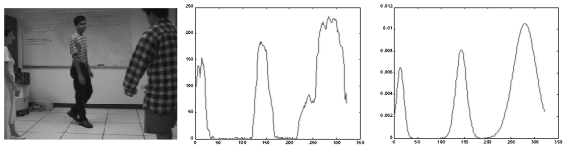

Figure 3. Detecting New Person: One more Gaussian than the number of known persons is fitted to the projection. The 3rd person is correctly detected.

Occlusion

This framework implicitly handles occlusion well. When there is more than one person, we just have to keep track of which subset of classes belongs to each person. When complete or partial occlusion occurs, we do not delete the classes that are no longer visible. The occluded classes retain their statistics, while statistics of other classes are updated slowly. Upon re-emergence of an occluded region, the classification is automatically correct, provided the color and location of the region has not changed significantly during the time when it was occluded. This is so because the re-emerging pixels will still have the highest likelihood of being assigned to their correct class rather than any of the other classes. Figure 2 shows some results of this step. The two persons in this figure entered the scene at different times and were segmented into classes by the initial segmentation algorithm. Then frame-to-frame correspondence was done. The results are very good even in the case of complete occlusion and re-emergence of the occluded person. It can be seen that some pixels are misclassified in the case of occlusion. However, most of the pixels are classified correctly. Once the occlusion event is complete and the persons separate out again, the system is able to recover from these misclassifications successfully.

Finding New Persons

Detecting new persons entering the scene is difficult because of the way we perform frame-to-frame correspondence. For every pixel in a new frame, the most likely label is assigned from the set of existing classes. Since Eq.(13) is just a max operation, pixels belonging to the new person will incorrectly get assigned to one of the existing classes, even though their likelihood value might be lower than usual. In the case of a new person, however, we need to make new classes to represent that person. This makes it necessary to identify when a new person has entered and also to determine the bounds of the new person. The initial segmentation algorithm is applied within this region

only to segment the new person into regions. A separate data structure keeps a record of which classes belong to which person.

To detect a new person, we use a pseudo-connected component approach. We look at the vertical projection of the entire foreground image, and fit this with a 1D Gaussian mixture of $N+1$ distributions, where N is the number of persons currently known. To fit this mixture model, we again employ the EM algorithm, described by equations Eq.(4) to Eq.(10), but use a 1D version of it. If the fit is correct, N Gaussians will fit to the existing persons and the $(N+1)^{th}$ Gaussian will fit to the rest of the data. Since we already know the spatial means of all the existing persons, we find the Gaussian which does not fit any of the existing people by finding the mean which is farthest away from the existing person means $\mu_1 ... \mu_n$. This gives us the mean of the Gaussian fitting the residual data.

If this 'residual' Gaussian distribution is 'well-formed' i.e. it is representing significant portion of the data, then that is an indication of a new person. For this purpose, we do two tests. First we check if the residual Gaussian is far from all Gaussians representing the existing persons. This is necessary because in the absence of an actual new person, the residual distribution could easily fit a small hump on the projection of an existing person. Secondly we check for the 'peakiness' of the residual Gaussian. If this Gaussian is very low and wide, then it does not represent a new person. Thus the ratio ω_r/σ_r should be high for a good fit to a new person. Figure 3 shows sample working of this algorithm.

5. CONCLUSION

We have presented a new framework for dealing with the occlusion problem in human tracking. We observe that in indoor sequences and unconstrained camera positioning, occlusion is very likely to occur. This framework identifies persons *during* occlusion, unlike some of the existing methods which can resolve the occlusion problem only after the occlusion event is completed. We show results that deal well with 'hard' problem of multiple people and complete occlusion. We have observed that currently in this approach, dealing with shadows is a problem. Shadows too differ from the background model and are segmented out with the person. However they need to be identified as not being a part of the person if the tracker information is being sent to an activity recognition module.

References

[1] Aggarwal, J. K. and Q. Cai, "Human Motion Analysis: A Review" *CVIU*, Vol.73, No.3, March, pp. 428-440, 1999.

[2] Alder, M. D., "An Introduction to Pattern Recognition: Statistical, Neural Net and Syntactic Methods of Getting Robots to See", http://ciips.ee.uwa.edu.au/~mike/PatRec /node95.html, Sept, 1997

[3] Ayers, D. and M. Shah, "Monitoring Human Behavior in an Office Environment", *Interpretation of Visual Motion Workshop, CVPR-98*, June 1998.

[4] Azarbayejani, A. *et. al.*, "Real-Time 3D Tracking of the Human Body", *MIT Media Lab, Perceptual Computing Setion, TR No. 374*, May 1996.

[5] Bobick, A. F. *et. al.*, "The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment", *Teleoperators and Virtual Environments*, Vol.8, No.4, pp. 367-391, 1999.

[6] Boult, T. *et. al.*, "Frame-Rate Multibody Tracking for Surveillance", *IUW*, Nov 20-23, pp.305-308, 1998

[7] Cédras, C. and M. Shah, "Motion-Based Recognition: A Survey", *Image and Vision Computing* Vol.13, No.2, March 1995, pp. 129-155.

[8] Dettmer, S., A. Seetharamaiah, L. Wang and M. Shah, " Model-Based Approach for Recognizing Human Activities From Video Sequences", *Workshop on Recent Advances in Computer Vision*, Karachi, Pakistan, January 1-2, 1998.

[9] Fieguth, P. and D. Terzopoulos, "Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates", *CVPR-97*, pp. 21-27, 1997.

[10] Gavrila, D. M., "The Visual Analysis of Human Movement: A Survey" *CVIU* Vol.73, No.1, January, pp. 82-98, 1999.

[11] Grimson, W. E. L. *et. al.*, "Using Adaptive Tracking to Classify and Monitor Activities in a Site" *CVPR-98*, pp. 22-29, June 23-25, 1998

[12] Haritaoglu, I., D. Harwood and L. Davis, "W^4 - Who, Where, When, What: A Real-Time System for Detecting and Tracking People", *International Face and Gesture Recognition Conf*, 1998

[13] Kanade, T. *et. al.*, "Advances in Cooperative Multi-Sensor Video Surveillance", *IUW* November, 1998, pp. 3-24

[14] Lipton, A. J., H. Fujiyoshi, R. S. Patil, "Moving Target Classification and Tracking from Real-time Video", *IUW*, pp.129-136, 1998.

[15] Olson, T. J. and F. Z. Brill, "Moving Object Detection and Event Recognition Algorithm for Smart Cameras", *IUW*, pp. 159-175, May 1997.