

Real Time Robust Human Detection and Tracking System

Jianpeng Zhou and Jack Hoang
I3DVR International Inc

780 Birchmount Road, Unit 16, Scarborough, Ontario, Canada M1K 5H4
Email: Jack@i3dvr.ca

Abstract

In this paper, we present a real time robust human detection and tracking system for video surveillance which can be used in varying environments. This system consists of human detection, human tracking and false object detection. The human detection utilizes the background subtraction to segment the blob and use codebook to classify human being from other objects. The optimal design algorithm of the codebook is proposed. The tracking is performed at two levels: human classification and individual tracking. The color histogram of human body is used as the appearance model to track individuals. In order to reduce the false alarm, the algorithms of the false object detection are also provided.

1. Introduction

With the increasing threat of terrorist, the advanced video surveillance system has to be put into use. The advanced video surveillance system needs to analyze the behaviors of people in order to prevent the occurrence of the potential dangerous case. The analysis of behaviors of people requires the human detection and tracking system. In recent years, the development of human detection and tracking system has been going forwards for several years, many real time systems have been developed [1][2][3][4][5]. However, there are still some challenging technologies need more researches: foreground segmentation and false alarm elimination. From the view of the state of arts, there is no perfect algorithm for foreground segmentation to be adaptive to tough situations such as heavy shadow, sudden light change, tree shaking so on. Most systems of human detection and tracking work fine in the environment with gradual light change, however, they fail to deal with the situation with sudden light change, tree shaking and moving background.

In this paper, we present a real time robust human detection and tracking system with the ability to deal with tough situations. This system consists of foreground segmentation, human recognition, human tracking and false object detection. Through the step of the false object detection, the system can learn the false alarm which make the system more stable and robust. This system has been integrated with DVR system. From the test result in varying environments, this system can handle most of tough situations such as sudden light change, heavy shadow, the objects in background remove etc.

2. Previous Works

There are two kinds of techniques for foreground segmentation: optical flow computation [6] and background subtraction. Although the optical flow computation can provide better performance, it is computationally expensive method and unsuitable for real time system. To solve the problem of computation consuming, Y.L.Tian and A.Hampapur combine these two techniques together [7]. They firstly use the background subtraction to locate the motion area, and then perform the optical flow computation only on the motion area to filter out false foreground pixels. The background subtraction is popularly used in foreground segmentation. The motion information is extracted by thresholding the difference between the current image and background image. The background can be modeled as Gaussian distribution $N(\mu; \sigma^2)$, this basic Gaussian model can adapt to gradual light change by recursively updating the model using an adaptive filter [3]. However, this basic model will fail to handle multiple backgrounds, such as water wave and tree shaking. To solve the problem of multiple backgrounds, the models such as the mixture of Gaussian [8], Nonparametric Kernel [9], and codebook [10] are provided recently. Although these algorithms are effective for modeling multiple backgrounds, they require more memories and more

computation. In this paper, we proposed an improved algorithm of foreground segmentation based the basic Gaussian model considering the environments with the sudden light change, shadow and tree shaking.

There are two kinds of ways for human recognition: neural network based [11] and model based [2][12]. The neural network is powerful tool for pattern recognition. In [11], the BP network was used to recognize the pedestrian. The model based human recognition analyzes the shape of object and classify the people from other objects. In order to recognize the people, we introduce the codebook to model the shape of human, and propose the distortion sensitive competitive learning algorithm to design the codebook.

The goal of tracking algorithm is to establish the correspondence between the people in current frame and the people in previous frame and find out what every individual is doing. In order to track the people, the human model has to be created. The human model includes the human features, such as the color, aspect ratio, edge, velocity etc. Occlusion is a significant problem in human tracking. Some previous work does not deal with occlusion at all. In order to solve the problem of occlusion, Kalman filter based method [13] and appearance-based tracking method [14][15] were proposed. To evaluate the performance of system, C.E.Erden, B.Sankur and A.M.Tekalp provides a method [16]. In our system, the color histogram is used to model the human feature.

Most of algorithms developed in previous works were based on RGB color space. In real time system, we have to convert the image from the YUV color space to RGB space. This kind of work increases the burden of CPU. Our system developed is based on YUV color space to save the CPU usage.

3. Architecture

Human detection and track system consists of the following parts: foreground detection, blob segmentation, human recognition, human tracking and false object detection. In our system, background subtraction approach is used for foreground detection. After the background subtraction, shadow detection is applied. In order to filter out the camera noise and irregular object motion, we use the morphological operations following the shadow detection. Then the foreground mask image is formed. After that the blobs are segmented from the foreground mask image. Because of noises, one object maybe includes several blobs. So, Blob merge is used to form the whole object following the blob segmentation. There are two ways for human classification: one way is to use the codebook to recognize whether the blob is human or

not; another way is to track the blob, if the blob is tracked successfully, this object is human. The appearance-based tracking approach is used for blob tracking and human tracking. The two kinds of false object detection are used to reduce the false alarm and adjust the background model. The architecture of system is described in fig.1.

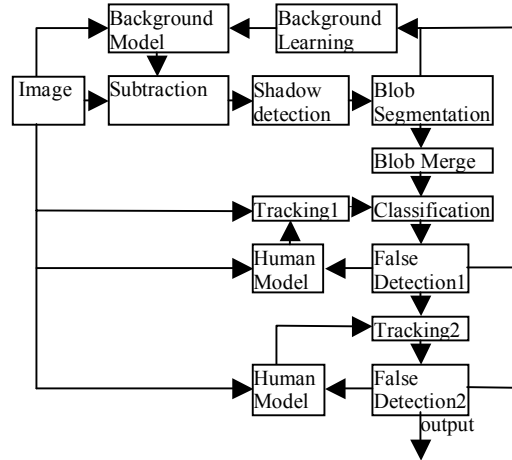


Fig.1 Architecture of system

4. Background Subtraction

The background subtraction system is used to provide foreground image through the threshold of difference image between the current image and reference image. If the reference image is the previous frame, this method is called temporal differencing. The temporal differencing is very adaptive to dynamic environment, but generally does a poor job of extracting all relevant feature pixels. The mixture of Gaussian, Nonparametric Kernel, and codebook can behave better performance, but they need extra expensive computation and more memories. For real time system, especially integrated with DVR system, we don't think their cost is worthy, because the CPU usage and memories usage are very significant for system stability. In our system, we use the running average as background model, and use the equation (1) and (2) to update in order to adapt to gradual light change.

$$\mu_{n+1} = \alpha\mu_n + (1 - \alpha)P_{n+1} \quad (1)$$

$$\sigma_{n+1} = \alpha\sigma_n + (1 - \alpha)|\mu_{n+1} - P_{n+1}| \quad (2)$$

Where μ_n is a running average, σ_n is a standard deviation, P_n is pixel value, α is updating rate in the n^{th} frame. In order to filter out some noises caused by camera movement, water wave and tree leaf shaking, we have to figure out the new way to create the

difference image between the current image and the background image because the traditional way is not successful for dealing with such situations. Following the following steps, we will get the robust difference image:

We assume B_n is an pixel in background image, $B_n^1, B_n^2, B_n^3, B_n^4$ are its neighbors on the vertical and horizontal directions. P_n is the responding pixel of B_n in current image, and P_n^1, P_n^2 are its neighbors on the vertical direction. Firstly, we compute the intensity histogram of pixels in the window $r \times r$ centered by B_n , and then find the intensity value M_n with the maximum possibility in the window $r \times r$. In our system, $r = 7$. Secondly, we calculate the median value \hat{P}_n of intensity values of P_n, P_n^1, P_n^2 ; and calculate the mean value \bar{B}_n of intensity values of $B_n^1, B_n^2, B_n^3, B_n^4$. Finally, the difference value D_n can be computed according to the equation (3) based on assumption that water wave and tree shaking are the movement of the part of background.

$$D_n = \min(|\hat{P}_n - M_n|, |\hat{P}_n - \bar{B}_n|, |\hat{P}_n - B_n^Y|) \quad (3)$$

Where $|a|$ is the function of computing the absolute value of a , B_n^Y is the intensity value of B_n .

In order to create the foreground mask image, we assume MSK_n is the value of corresponding pixel of P_n in the foreground mask image. TH_1 and TH_2 are threshold values we used. We define TH_2 is greater than TH_1 . In order to filter out the shadow that is cast by people moving in the scene, we use the following rule to threshold the difference image and get the foreground mask image: If $D_n < TH_1$, then $MSK_n = 0$; if $D_n \geq TH_2$, then $MSK_n = 1$; if D_n is between TH_1 and TH_2 , we will check whether P_n is shadow or not. If P_n is shadow, $MSK_n = 0$, else $MSK_n = 1$.

The selection of TH_1 is the key for successful threshold of the difference image. If TH_1 is too low, some background are labeled into foreground. If TH_1 is too high, some foreground are detected as

background. The traditional way is to select 3σ as TH_1 [3] based on the assumption that illumination gradually changes. However when light suddenly changes, this assumption will be violated. Instead of the traditional way, we have to automatically select TH_1 according to the level of light change. Firstly, we search the median value MID of difference image. After that, the equation (4) is used to compute TH_1 for each pixel.

$$TH_1 = MID + 2\sigma + TD \quad (4)$$

Where TD is initial threshold, normally $TD < 10$. In our system, we select $TD = 5$.

The TH_2 can be selected as $TH_1 + Gat$ where Gat is a gate. In our system, Gat is 50.

In order to adapt the sudden light change, we set different α to update the background for different level of light change. The rule for the selection of updating rate α is:

$$\alpha = \begin{cases} \alpha_1 & \text{if } MID < T_1 \\ \alpha_2 & \text{if } T_1 \leq MID < T_2 \\ \alpha_3 & \text{others} \end{cases} \quad (5)$$

Where $T_1 < T_2$. In our system, we use $\alpha_1 = 0.9$, $T_1 = 4$; $\alpha_2 = 0.85$, $T_2 = 7$; $\alpha_3 = 0.8$.

5. Shadow Detection

The shadow affects the performance of foreground detection. The regions of shadow will be detected as foreground. Recognizing shadow is hard work, and now this subject is still under research. There is no perfect algorithm for shadow detection. Some of algorithms were developed in previous work [3][17] with respect to the specific application. The assumption for the shadow detection is that the regions of shadow are semi-transparent. An area cast into shadow often results in a significant change in intensity without much change in chromaticity. S.J. McKenna proposed a method to detect shadow based on RGB color space [3]. In order to utilize the color invariant feature of shadow, we use the normalized color components in YUV color space, which are defined as $U^* = U/Y, V^* = V/Y$. Our shadow detection algorithm is described as following.
Step1: Compute the color difference.

We assume bU_n^*, bV_n^* are the normalized color components of B_n, cU_n^*, cV_n^* are the normalized

color components of P_n . The color difference is defined as equation (6).

$$diff_c = |cU_n^* - bU_n^*| + |cV_n^* - bV_n^*| \quad (6)$$

Step2: Compute the texture difference.

We assume B_n^Y is the intensity value of B_n in background image, $B_n^{Y1}, B_n^{Y2}, B_n^{Y3}, B_n^{Y4}$ are the intensity values of pixels of its neighbors $B_n^1, B_n^2, B_n^3, B_n^4$ on the vertical and horizontal direction. P_n^Y is the intensity value of P_n pixel in current image, and $P_n^{Y1}, P_n^{Y2}, P_n^{Y3}, P_n^{Y4}$ are the intensity values of pixels of its neighbors $P_n^1, P_n^2, P_n^3, P_n^4$ on the vertical and horizontal direction. The texture difference is defined as equation (7).

$$diff_t = \sum_{i=1}^4 |Th(|P_n^Y - P_n^{Yi}|) - Th(|B_n^Y - B_n^{Yi}|)| \quad (7)$$

Where $Th(Val)$ is a function defined as equation (8).

$$Th(Val) = \begin{cases} 1 & \text{if } Val > Th \\ 0 & \text{others} \end{cases} \quad (8)$$

Step 3: Shadow decision

If $diff_t = 0$ and $diff_c < cTh$ and $P_n < B_n$, then P_n is shadow, otherwise P_n is not shadow.

Where cTh is the color threshold. The assumption for $P_n < B_n$ is that the region of shadow is always dark than background.

6. Classification

The ultimate goal of system is to be able to identify people and track individuals to find out what they are doing. For human recognition, we introduce the codebook to classify the human from other objects. First step, we normalize the size of object to 20 by 40, and then extract the shape of object as the features. Second step, we match the feature vector with the code vectors of codebook. The match process is to find a code vector in codebook with the minimum distortion to the feature vector of object. If the minimum distortion is less than a threshold, this object is human.

In order to describe the procedure of classification based on codebook. We assume W_i is the i^{th} code vector in code book. X is feature vector of object. N

is the number of code vectors in code book. The dimension of code vector is M . So the distortion between W_i and X is computed as equation (9).

$$dist_i = ||W_i - X|| = \sum_{j=0}^M |W_i^j - X^j| \quad (9)$$

The minimum distortion between X and the code vectors in the code book is defined as equation (10).

$$diss = \min(dist_i) \quad i = 0, \dots, N-1 \quad (10)$$

If $diss$ is less than a threshold, the object with the feature vector X is human, otherwise, it is not human.

In order to create the shape vector of object, the mask image and boundary of human body are created by the previous steps (Fig 2). We use the distance from the boundary of human body to the left side of bounding box as feature vector. Fig2(a) is the mask image of human body and Fig2(b) is the boundary of human body. From the boundary of human body, firstly, we select 10 points in left side of the boundary, and compute their distances to left side of bounding box. Then we select 10 points in right side of boundary, and compute their distance to left side of bounding box. The shape vector of object is created by the set of these distances. So, in our system the dimension of the shape vector is 20.

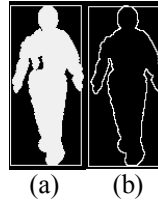


Fig 2 (a) Mask image of human (b) Boundary of human body.

The design of the codebook is critical for the classification. The well-known partial distortion theorem for design codebook is that each partition region makes an equal contribution to the distortion for an optimal quantizer with sufficiently large N [18]. Based on this theorem, we proposed distortion sensitive competitive learning (DSCL) algorithm to design the codebook. In order to describe this algorithm, we define $W = \{W_i; i = 1, 2, \dots, N\}$ as the codebook and W_i is the i^{th} code vector. X_i is the t^{th} train vector and M is the number of train vector. D_i is the partial distortion of region R_i , and D is the average distortion of codebook. The DSCL algorithm is described as follows.

Step1: Initialization 1

Set $W(0) = \{W_i(0); i = 1, 2, \dots, N\}$ and $D(0) = \infty$,
 $D_j(0) = 1, j = 0$.

Step 2: Initialization 2

Set $t = 0$

Step3: Compute the distortion for each code vector

$$dis_i = \|X_t - W_i(t)\|$$

Step4: Select the winner: the k^{th} code vector.

$$dis_k^* = \min(D_i(t)dis_i) \quad i = 1, 2, \dots, N$$

Step5: Adjust the code vector for winner

$$W_k(t+1) = W_k(t) + \varepsilon_k(t)(X_t - W_k(t))$$

Step6: Adjust D_k for winner

$$\Delta D_k = \frac{N_k}{t+1} \|W_k(t) - W_h(t+1)\| + \frac{1}{t} dis_k$$

$$D_k(t+1) = D_k(t) + \Delta D_k$$

Where N_k is the number of train vectors belonging to region R_k .

Step7: Check whether $t < M$

If $t < M$ then $t = t + 1$, and go to step3. Others go to step 8.

Step8: Compute $D(j+1)$

$$D(j+1) = \frac{1}{M} \sum \|X_i - W\|$$

If $\frac{|D(j+1) - D(j)|}{D(j)} < \varepsilon$ stop, else $j = j + 1$,

then go to step2.

In our system, the blob tracking is also used for human classification. When the blobs in current frame have been segmented, we track these blobs using the blobs in previous frame. If the blob is successfully tracked, then we think this blob is human. Otherwise, we use the code book to recognize it.

7. Appearance-based Tracking

In order to track the individuals, the human model has to be created for each individual. The good human model should be invariant to rotation, translation and changes in scale, and should be robust to partial occlusion, deformation and light change. We use the color histogram, direction, velocity, the number of pixels and size as the human model to describe the humans. In order to decrease the computation cost, we define the color of pixel as equation (11).

$$I_n = 0.3P_n + 0.35U_n + 0.35V_n \quad (11)$$

Where P_n, U_n, V_n are the Y,U,V value of pixel in current image, and I_n is the color value we use to compute the histogram. We assume H_t and H_{ref} are the current histogram and reference histogram, and then the comparison rule of histogram is defined as equation (12).

$$H_s = \frac{\sum_{i=0}^{255} \min(H_t(i), H_{ref}(i))}{\min(N_H^t, N_H^{ref})} \quad (12)$$

Where N_H^t and N_H^{ref} are defined as follows:

$$N_H^t = \sum_{i=0}^{255} H_t(i), N_H^{ref} = \sum_{i=0}^{255} H_t^{ref}(i) \quad (13)$$

For tracking, we assume the human always moves in similar direction and similar velocity. During the process of tracking, we will check whether the people stop or change the direction. If the person doesn't move for period of time, we will check whether this person is false. Once the false person is found, system will learn this false alarm and adjust the background.

There are two level of tracking: blob level tracking and human level tracking. The goal of blob level tracking is to help classify the human from other objects. The goal of human level tracking is for analysis of human activity and false human detection. The main condition of blob level tracking is stricter than that of human level tracking.

8. False Object Detection

In our system, we detect the false objects caused by the following cases: sudden light change, humans in background removes and tree shaking. The purpose of false human detection is to heal the background image by selectively adding uninteresting, stationary foreground object to it. The false detection is done during the process of tracking. There are two level of false object detection: blob tracking level and human tracking level. In blob tracking level, we mainly detect the false blobs caused by removed, dropped object or light change. The common feature of these kinds of false blobs is stationary, so when a stationary blob is tracked for period of time, we begin to check whether this blob is false or not. In human tracking level, we mainly detect the false blobs caused by tree shaking. The feature of this kind of false objects is that the human can't be successfully tracked in similar direction and the move scope is limited for period of time. When a person moves in some limited area for some period of time and is unsuccessfully tracked

most time, we begin check whether this person is false.

In blob tracking level, the based assumption of detection method is that object boundaries coincide with color boundaries. The following steps are used to detect the false blob.

Step1: use the foreground mask image to create the boundary of blob. For every pixel in boundary, find two points P_o and P_i outside and inside boundary respectively. P_o and P_i have the same distance to the boundary. This is illustrated in Fig3.

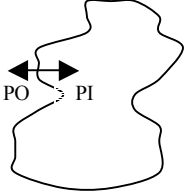


Fig 3 Find points P_o and P_i outside and inside boundary

Step 2: We assume N_b is the number of pixels on boundary of blob. Compute the gradient feature G_c of the boundary in current image and the gradient feature G_b in background image. The gradient feature G of the boundary is calculated using the equation (14).

$$G = \sum_{j=1}^{N_b} Grad(|Po^j - Pi^j|) \quad (14)$$

Where Po^j, Pi^j are the outside and inside point with respect to the j th point of boundary of blob. The Function $Grad(Val)$ is defined as follows:

$$Grad(Val) = \begin{cases} 1 & \text{if } Val > GTh \\ 0 & \text{others} \end{cases} \quad (15)$$

Where GTh is the gradient threshold.

Step3: Make the decision. If $G_c > 1.2 G_b$ or $G_c < 0.3 N_b$, then this blob is false.

In human tracking level, the assumption for false object detection is that the false objects are caused by the movement of a part of background, like the tree branch shaking. The detection algorithm is described as follows.

Step1: Utilize the color histogram of object to check whether all the pixels of the object are of one similar color. Because the false object is a part of tree or other background, their pixels have the similar color. If the object is false, the pixel values should be centered by one pixel with maximum probability in color histogram.

Step2: Use the technique of the region growing to check whether the object is one part of background.

We use the average color as a seed to grow the pixels on vertical direction to form a larger region. If the number of pixels covered by extended region is more than the number of original object, then this object is false.

9. Experimental Results

In order to prove the performance of the proposed algorithm, we test our system with the different videos in varying environments. Firstly, we test the performance of background subtraction algorithm using the video with the moving background. Fig 4 illustrates the test result using a video with tree shaking and heavy shadow. Inside this video, the shadow of tree changes frequently because of the branch shaking. From Fig4(b), two people are detected and the branch shaking of tree on the upright corner in the image is also detected, however, human recognition and false object detection will filter out the branch shaking of tree.



(a) Original (b) Mask Image

Fig.4 Background Subtraction algorithm in moving background

Second, we test the performance of human classification based on codebook. We chose 10 people from our R&D and lent them do different activities such as walking, jumping, running, and sitting. So we can collect the training videos. And then we train the code book using the feature vectors created by the training videos. The number of the code vectors in code book is 256 in our system. Fig5(a) and Fig5(b) show 2 frames of training videos.



(a) Train Image1 (b) Train Image2

Fig5 Sample image of training videos

We use the videos collected from outdoor and indoor to test the performance of human classification. The test result is that more than 99% human are correctly classified if the human is not far from the camera. And the vehicles on the street are not classified as human. However, some chairs are sometimes classified as human. Some test image samples are given in fig 6.



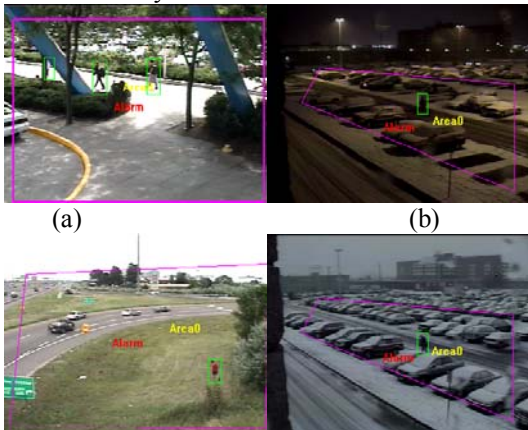
(a) (b)
Fig6 Sample image of test videos

In our system, there are 4 functions based on the human detection and human tracking: area alarm, crosswire alarm, idle alarm and pass through counter. The area alarm will trigger alarm when the people enter the predefined area. The crosswire alarm will detect and track people, when the people cross a predefined line in a specific direction, it will trigger the alarm. When some people stay inside the specific area more than predefined period of time, the idle alarm will trigger the alarm. The pass through counter will counter the number of people who pass through a predefined gate. The test results for these four functions are described in table 1.

Table1 The test results of functions

Camera	Area alarm	Crosswire Alarm	Idle Alarm	Counter
Angle	98%	98%	98%	98%
Above	93%	90%	92%	85%
Far away	95%	92%	95%	93%

From the table1, the test results showed that our system is effective for human detection and tracking. And now, we check the performance of human detection in varying environment. Fig7(a) shows the test result in the environment with light sudden change and tree shaking. Fig7(b) shows the test result in the environment with low light. The background is pretty dark; however, the person walking on the road is still detected. Fig7(c) shows the test result in the environment beside the highway. The vehicles move on the highway, a person walks on the lawn with the grass waving and tree shaking. Fig7(d) shows the result in snowy environment.



(c) (d)
Fig 7 (a) Detection in changing environment (b) Detection at night with low light (c) Detection beside highway (d) Detection in snowy environment

However, in a situation when the person enters the house through an automatic door, our system fails to detect the human at the moment the person is entering at the door. Because the automatic door is also detected as foreground when the door moves, the person and door are detected as one object. Our classification algorithm recognizes this object as no human. In another situation, when two people walk together, if they are very close, they will be classified as one person. Because these two people are detected as one object, our classification is based on the shape of human body.

From the above experiments, the test results proved that the system based on the proposed algorithm is robust. We use the computer with P4 3.0 G Hz and 512 MB memory to test the CPU usage for 4 channels, use the image size of 320×240 and set the 15 fps for video capture with the DVR system, the CPU usage is below 50%.

10. Conclusion

We presented a real time robust human detection and tracking system which can perform in varying environment. In this system, we used the background subtraction technique considering the camera moving, shadow and tree shaking to segment the foreground. This algorithm has been proved to be robust to varying environment. During the process of human recognition, we introduce the codebook to recognize the human. In order to reduce the false alarm, we proposed the algorithms of false detection. The experiments also proved that the tracking algorithm based on color histogram is robust to partial occlusion of people.

11. References

- [1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfindex: Real-time tracking of the human body," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.19, 1997, pp. 780-785
- [2] I.Haritaoglu, D.Harwood and L.S.Davis, "W4: Who? When? Where? What? A Real Time System for Detecting and Tracking People", In Proc. Of the International Conference on Face and Gesture Recognition, April, 1998
- [3] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. "Tracking Groups of People". Computer Vision and Image Understanding. 80:42-56, 2000

- [4] J. Connell, A.W. Senior, A. Hampapur, Y-L Tian, L. Brown, and S. Pankanti, "Detection and Tracking in the IBM PeopleVision System", IEEE ICME, June 2004
- [5] L.M.Fuentes and S.A.Velastin, "People Tracking in surveillance application", in Proc. 2nd IEEE International Workshop on PETS, Dec. 2001
- [6] S.S. Beauchemin and J.L.Barron, "The Computation of Optical flow", ACM Computing Surveys, Vol.27, 1995, pp 433 - 466
- [7] Y.L.Tian and A.Hampapur, "Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance", in Proc. Of IEEE Computer Society Workshop on Motion and Video Computing, January, 2005
- [8] C.Stauffer and W.E.L.Grimson, "Adaptive background mixture models for real tracking". Int. Conf. Computer Vision and Pattern Recognition, Vol.2, 1999,pp246-252
- [9] A.Elgammal, R.Duraiswami, D.Harwood and L.Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance", Proceeding of the IEEE, Vol.90, No.7, July, 2002
- [10] K. Kim, T. H. Chalidabhongse, D. Harwood and L. Davis, "Background Modeling and Subtraction by Codebook Construction", IEEE International Conference on Image Processing (ICIP), 2004
- [11] L.Zhao and C.E.Torpe, "Stereo- and Neural Network-Based Pedestrian Detection", IEEE Trans. Intelligent Transportation System, Vol.1,No.3, Sept., 2000
- [12] C.BenAbdelkader and L.Davis," Detection of People Carrying Objects : a Motion-based Recognition Approach", 5th IEEE International Conference on Automatic Face and Gesture Recognition ,May, 2002
- [13] J.W.Lee, M.S.Kim, I.S.Kweon, "A Kalman filter based visual tracking algorithm for an object moving in 3D ", IEEE International Conference on Intelligent Robots and System, Vol.1, Aug., 1995
- [14] A.Senior, A.Hampapur, Y.L.Tian, L.Brown, S.Pankanti and R.Bolle,"Appearance Models for Occlusion Handling", in proceedings of Second International workshop on Performance Evaluation of Tracking and Surveillance systems in conjunction with CVPR'01, Dec. 2001
- [15] M. Balcells-Capellades, D. DeMenthon and D. Doermann. "An Appearance-based Approach for Consistent Labeling of Humans and Objects in Video". Pattern Analysis and Applications, November 2004.
- [16] C.E.Erden, B.Sankur and A.M.Tekalp, "Performance Measure for Video Object Segmentation and Tracking", IEEE Trans. Image Processing, Vol.13,No.7,July,2004, pp937-951
- [17] P.L.Rosin and T.Ellis, "Image Difference Threshold Strategies and Shadow Detection", In Proc. Of 6th BMVC Conference, 1995, pp347-356
- [18] A. Gersho, "Asymptotically optimal block quantization," IEEE Trans. Inform. Theory, vol. 25, July 1979, pp. 373–380.