Object Tracking for Real Time Video using MATLAB

Madhvi Bagga Panwar [*],Priyanka Goyal [#]

[*] 1st ECE Deptt., Madhav University, Abu Road
madhvi.bagga@gmail.com

[#] 2nd ECE Deptt., Madhav University, Abu Road
Priyankagoyal2102@gmail.com

Abstract— **In this paper object tracking for real time video is developed which, demonstrates the motion compensated video processing by using sum of absolute differences. First an object has taken as reference object or image then the next successive object is compared with the reference object or image. Each time the successive object is compared with the reference object and produces an absolute difference, then the summation of all these differences shows its sum of absolute difference. This difference shows the change in the two images. Finally by using negative threshold, the change in the motion of sum of absolute differences in the object image is shown. A simulink model is also developed for object tracking for real time video.**

Keywords- Absolute, Threshold, Tracking, Real time.

I. INTRODUCTION

Image tracking and activity recognition are receiving increasing attention among computer scientists due to the wide spectrum of applications where they can be used, ranging from athletic performance analysis to video surveillance. By image tracking we refer to the ability of a computer to recover the position and orientation of the object from a sequence of images[4]. There have been several different approaches to allow computers to derive automatically the kinematics pose and activity from image sequences.

In digital video communication systems it is important that a video to be compressed, because of storing capacities as well as bit-rate constraints. The video processing is done using Sum of Absolute Differences and with the image processing block set. First motion vectors between successive frames are calculated and use them to reduce redundant information[8]. Then each frame is divided into sub matrices and apply the discrete cosine transform to each sub matrix. Finally, apply a quantization technique to achieve further compression. The Decoder subsystem performs the inverse process to recover the original video.

II. TRACKING: POSSIBLE ISSUES

A. Introduction:

Video tracking is the process of locating a moving object in time using a camera. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. The main difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate[10]. Here, video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track. The role of the tracking algorithm is to analyze the video frames in order to estimate the motion parameters. These parameters characterize the location of the target.

B. Component of visual Tracking system:

Target Representation and Localization is mostly a bottom-up process. Typically the computational complexity for these algorithms is low. The following are some common Target Representation and Localization algorithms:

- Blob tracking: Segmentation of object interior (for example blob detection, block-based correlation or optical flow).
- Kernel-based tracking (Mean-shift tracking): An iterative localization procedure based on the maximization of a similarity measure.
- Contour tracking: Detection of object boundary (e.g. active contours or Condensation algorithm).
- Visual feature matching: Registration

One approach to reduce the problem space and to make the problem computationally tractable is to provide constraints on the positions of the object. Constraints can be based on temporal information, camera configuration, or any combination of these. Camera configuration constraints are usually expressed by

making assumptions on the relative positioning of the subject with respect to the camera.

III. OPTIMIZATION METHOD OF TRACKING

Most human motion and pose estimation approaches propose some sort of optimization method, direct or probabilistic, to optimize the pose (and/or body model) subject to the image features observed.

A. Direct Optimization

Direct optimization methods often formulate a continuous objective function F(Xt, It), where Xt is the pose of the body at time t and It is the corresponding observed image, and then optimize it using some standard optimization technique. Since F(Xt, It) is highly non-linear and non-convex there is almost never a guarantee that a global optimum can be reached. However, by iteratively linearizing F(Xt, It) and following the gradient with respect to the parameters a local optimum can be reached[11]. If a good estimate from the previous time step is available, and the pose changes slowly over time, then initializing the search with the previous pose often leads to a reasonable solution.

B. Probablistic Inference

It is often convenient and natural to formulate tracking and pose estimation as probabilistic inference. A probabilistic framework has two advantages over the direct optimization methods:

- It can encode the confidence of any given articulated interpretation of the image.
- It allows one to maintain multi-modal predictions both spatially and over time. Multi-modality arises naturally in human motion estimation, since the body in different postures can look very similar (if not identical) in the image.

The number of valid interpretations of the images depend significantly on the features used, imaging conditions and the temporal history. By maintaining a multi-modal pose hypothesis over time, approaches can often benefit by resolving the ambiguities as more information becomes available.)

C. Video generation and Processing

As said that an image is a set of pixels so for the generation of a video the scanning of the each and every pixel is necessary .So video is generated by scanning the pixels and each pixel represented by a value or set of values. The pixels are scanned as shown in the above figure. The scanning starts from the right most pixel to the left most pixel in the first row and then comes back to the next row and then start from the right most pixel. towards the end of the row and so on. Once after the scanning entire image then it again returns back to the starting point as shown.

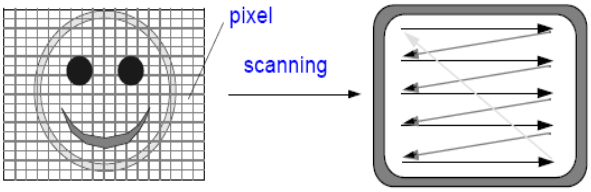

Figure 1 : Video Generation

For the best results interlaced scanning is employed in which the image is divided in to two fields, even field and odd field.

Video processing is a very important phenomenon now a days. Many processing methods are widely used either in television systems[6], video post production or even in common life. Despite the fact that professional hardware video processing solutions exist, software video processing is very popular mainly because of the great flexibility it offers.

By transforming a signal the energy is separated into sub bands, by describing each sub band with different precisions, higher precision within high energy sub bands and less precision in low energy sub bands, the signal can be compressed. The most common transform used is the DCT (Discrete Cosine Transform) which has excellent in energy compaction which means that the energy of the matrix is concentrated to a small region of the transformed matrix[2].

IV. MOTION COMPENSATED VIDEO PROCESSING

A. Overview

Block based motion compensation uses blocks from a past frame to construct a replica of the current frame. The past frame is a frame that has already been transmitted to the receiver. For each block in the current frame a matching block is found in the past frame and if suitable, its motion vector is substituted for the block during transmission. Depending on the search threshold some blocks will be transmitted in their entirety rather than substituted by motion vectors. The problem of finding the most suitable block in the past frame is known as the block matching problem. . Block based motion compensated video compression takes place in a number of distinct stages. The flow chart above illustrates how the output from the earlier processes form the input to later processes. Consequently choices made at early stages can have an impact of the effectiveness of later stages. To fully understand the issues involved with this type of video compression it is necessary to examine each of the stages in detail.

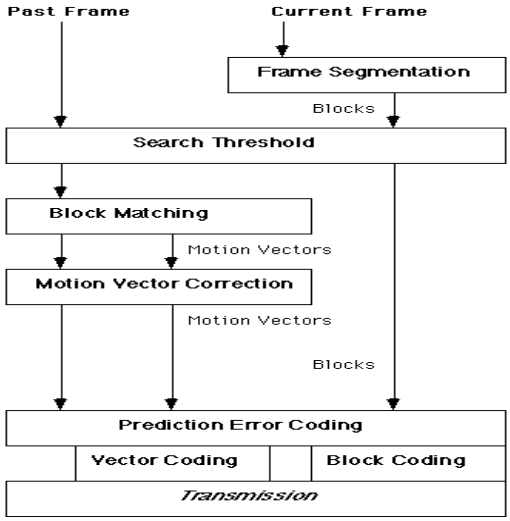

Figure 2 : Block Diagram of Motion Compensated Video Processing

These stages can be described as:

- Frame Segmentation
- Search Threshold
- Block Matching
- Motion Vector Correction
- Vector Coding
- Prediction Error Coding

B. Block Matching

Block matching is the most time consuming part of the encoding process. During block matching each target block of the current frame is compared with a past frame in order to find a matching block.

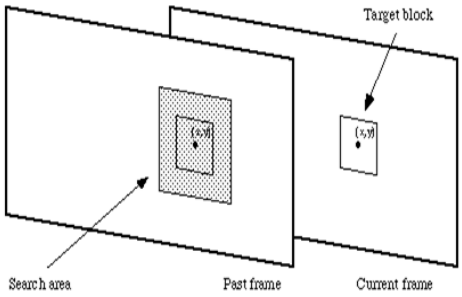

Figure 3: Corresponding blocks from a current and past frame, and the search area in the past frame.

When the receiver reconstructs the current frame this matching block is used as a substitute for the block from the current frame. Block matching takes place only on the luminance component of frames. The colour components of the blocks are included when coding the frame but they are not usually used when evaluating the appropriateness of potential substitutes or candidate blocks. The search can be carried out on the entire past frame, but is usually restricted to a smaller search area centred on the position of the target block in the current frame (see above figure). This practice e places an upper limit, known as the maximum displacement, on how far objects can move between frames, if they are to be coded effectively[10]. The maximum displacement is specified as the maximum number of pixels in the horizontal and vertical directions that a candidate block can be from the position of the target block in the original frame.

The quality of the match can often be improved by interpolating pixels in the search area, effectively increasing the resolution within the search area by allowing hypothetical candidate blocks with fractional displacements.

The search area need not be square. Because motion is more likely in the horizontal direction than vertical, rectangular search areas are popular. The CLM460x MPEG video encoder, for example, uses displacements of -106 to +99.5 pixels in the horizontal direction, and -58 to +51.5 pixels in the vertical. The half pixel accuracy is the result of the matching including interpolated pixels. The cheaper CLM4500, on the other hand, uses ±48 pixels in the horizontal direction, and ±24 in the vertical, again with half pixel accuracy. If the block size is b and the maximum displacements in the horizontal and vertical directions are dx and dy respectively, then the search area will be of size $(2dx + b)(2dy + b)$. Excluding sub-pixel accuracy it will contain $(2dx + 1)(2dy+1)$ distinct, but overlapping, candidate blocks.

C. Block Based Motion Compensation

Block based motion compensation, like other interframe compression techniques, produces an approximation of a frame by reusing data contained in the frame's predecessor. This is completed in three stages

First, the frame to be approximated, the *current frame*, is divided into uniform non overlapping blocks, as illustrated below (left)[12]. Then each block in the current frame is compared to areas of similar size from the preceding or past frame in order to find an area that is similar. A block from the current frame for which a similar area is sought is known as a target block. The location of the similar or matching block in the past frame might be different from the location of the target block in the current frame. The relative difference in locations is known as the Motion vector.

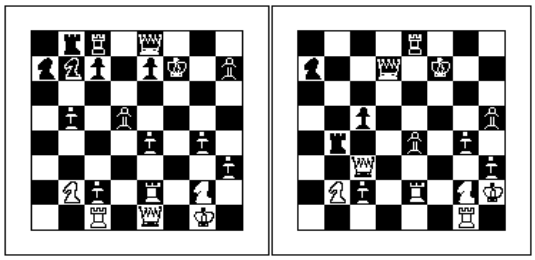

Figure 4 : Past Frame - Current frame to be coded

If the target block and matching block are found at the same location in their respective frames then the motion vector that describes their difference is known as a Zero vector. The illustration below shows the motion vectors that describe where blocks in the current frame (below left) can be found in past frame (above left).

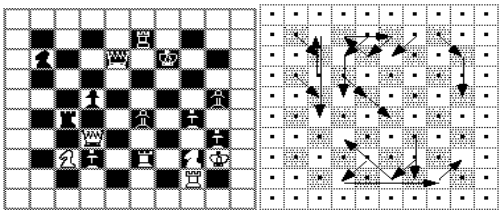

Figure 5 : Motion Vectors Indicating Changed Blocks

Current frame to be coded divided into blocks. Motion vectors indicating where changed blocks in the current frame have come from. Unchanged blocks are marked by dots.

Finally, when coding each block of the predicted frame, the motion vector detailing the position (in the past frame) of the target block's match is encoded in place of the target block itself. Because fewer bits are required to code a motion vector than to code actual blocks, compression is achieved.

V. SAD(SUM OF ABSOLUTE DIFFERENCE)

Sum Absolute Difference (SAD) is an operation frequently used by a number of algorithms for digital motion estimation. a single vector instruction is proposed that can be performed (in hardware) on an entire block of data in parallel . Assuming a machine cycle comparable to the cycle of a two cycle multiply, it has been shown that for a block of 16x1 or 16x16, the SAD operation can be performed in 3 or 4 machine cycles respectively. The proposed implementation operates as follows: first determination in parallel which of the operands is the smallest in a pair of operands .Second the absolute value of the difference of each pairs are computed by subtracting the smallest value from the largest and finally the accumulation is computed. The operations associated with the second and the third step are performed in parallel resulting in a multiply (accumulate) type of operation.

SAD operation is usually considered for 16x16 pixels (pels) blocks and because the search area could involve a high number of blocks, performing the SAD operation could be time-consuming if traditional methods are used for its computation. Here we implement a new instruction that is capable of producing the direct SAD operation. Furthermore we also show that the implemented instruction is scalable, depending on the constraints of the technology considered for the design. This is shown by considering a 16x1 sub-block element and an entire 16x16 element and showing that the implementation will require 3 machine cycles for a 16x1 sub-block and 4 cycles for a 16x16 block. The 16x16 block performance is achieved by using hardware proportional in size to a 16x1 sub-block unit, that is we achieve a 4 cycle 16x16 block SAD using approximately 16 times the area of the 16x1 SAD.

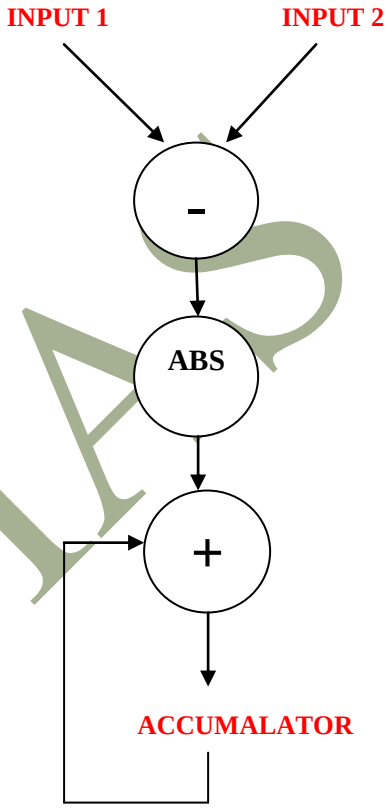

Figure 6 : Main Computation in the Sum of Absolute Differences Kernel

As shown in this figure, the main set of computations in the SAD kernel includes subtraction, followed by computing the absolute, and, finally, accumulating with previous results.

A. Graphical Representation

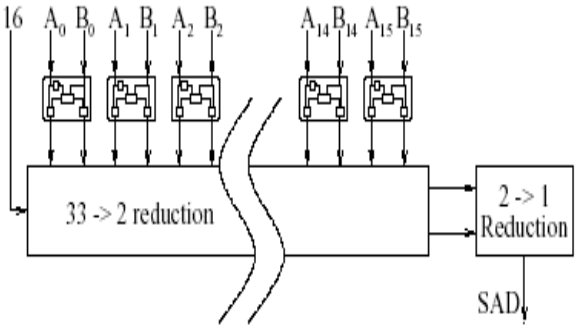

Figure 7 : Graphical Representation of a 16 x 1 Unit

Above figure shows a graphical representation of a 16x1 unit, that is a unit operation on 16 couples of elements producing a single output value. The top half shows 16 times steps 1 and 2 in parallel, and steps 4 and 5 are depicted in the bottom half. Step 3 is represented by the addition term at the left (16). The concept can be expanded to an array capable of computing the SAD of 16x16 pel blocks. In this case, the 2 rows going into the 2-to-1 reduction should go into another 32-to-2 reduction unit, together with the 30 rows of the 15 other units. The result of this 32-to-2 reduction is then reduced by a 2-to-1 final adder. This saves both the execution time and the area of 15 2-to-1 reduction units.

VI. SIMULINK MODELS FOR VIDEO PROCESSING

Motion detection is a key feature for a video surveillance system and can be used to alarm video/audio recording and transmission. However, reliable motion detection techniques should avoid the false alarms. A realistic motion detection technique should tolerate the optical noise reproduced by camera and only respond to the movement in the region of interest (ROI). To measure movement in video scenes, motion detection can use the sum of absolute difference (SAD) and correlation.

A. Simulink model formotion detection using SAD

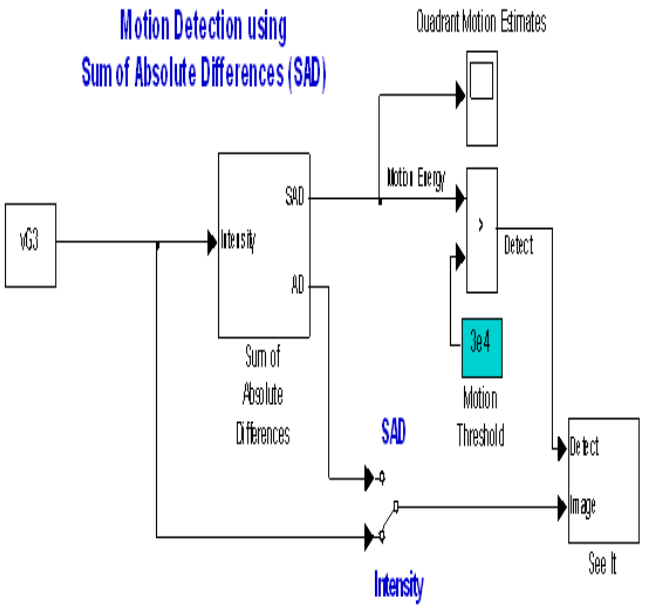

Figure 8 : Model for Motion Detection

Sometimes, the color information can also enhance the performance of motion detection. Many smart video surveillance systems currently in market support this feature.

B. Simulink Model for Surveillance Recording Based on Motion Detection

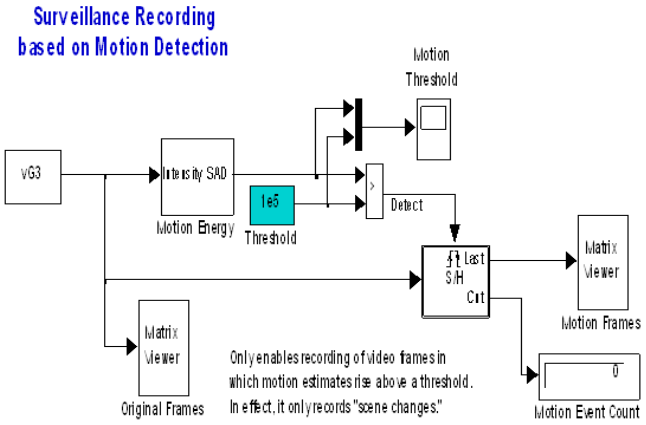

Figure 9 : Model for Surveillance Recording Based on Motion Detection.

VII. RESULTS AND CONCLUSION

A. Result of Object Tracking for Real Time Video

In Surveillance Systems :

The object images from Figure 10 are captured when there is a motion. These images will be shown in the form of Video Queue.

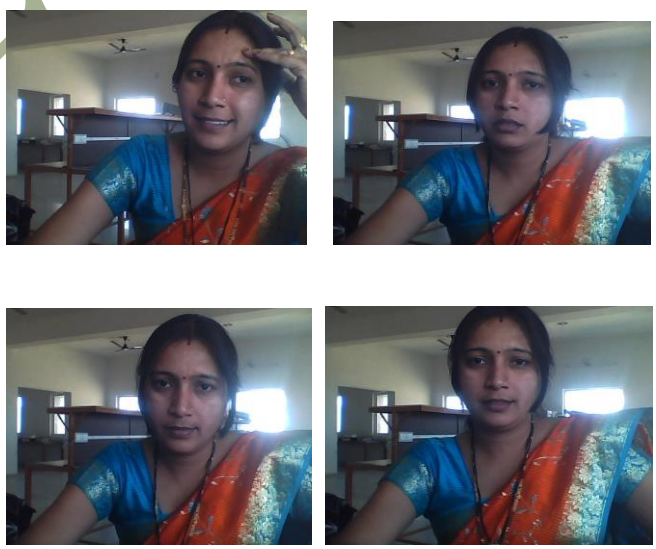

Figure 10. : Images captured when there is a motion

In Motion Tracking :

All these figures are captured when there is a change in motion and results were shown to the changes that occurred in object motion. Initially frame 1 explains the captured image , frame 2 explains the black and white characteristics of frame 1, frame 3 explains the greyscale information of frame 1, frame 4 shows the negative of the

frame 1 and frame 5 shows the objects motion which is tracked.

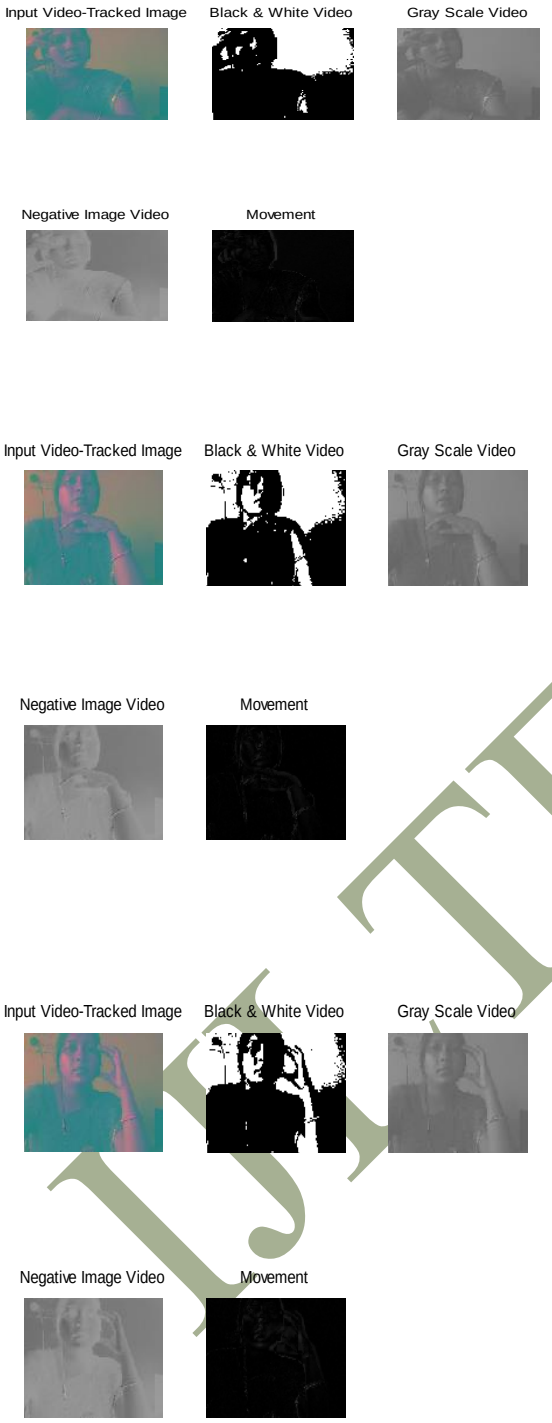

Figure 10. :Result of Object Tracking for Real Time Video

CONCLUSION

In this paper the main attention is on the object tracking for real time video Sum of Absolute Differences is used and designed it for object tracking for real time video to detect the motion of object in different views. the basic concepts of object tracking, properties and performance of object tracking , in various fields of its applications i.e. image tracking by keeping camera constant or camera in motion and object constant or object in motion. Some factors are identified which are not performing to its potential. These factors includes faster movements , single object among multiple object etc., and the noise effect and issues of implementing them is crucial for proper functionality. Here after discussion and the result it can be concluded that the Sum of Absolute Differences technique is easier and can be implemented easily and economical compare to the standard algorithms which are used for object tracking.

REFRENCES

[1] Digital Image Processing By Gonzalez 4th Edition
[2] Digital Signal Processing By John Proakis
[3] Web site : www.mathworks.com
[4] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," in Proc. Int. Conf. Acoustics, Speech, Signal Processing, vol. 2, Apr. 1988,
[5] B. J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT) in Proc. IEEE Data Compression Conf., Mar. 1997, pp. 251–260.
[6] Adams, R.; Bischof, L.; "Seeded region growing", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 16, Issue 6, June 1994 Page(s):641 - 647
[7] Chen, B.; Lei, Y.; "Indoor and outdoor people detection and shadow suppression by exploiting HSV color information", 4th International Conference on Computer and Information Technology, 14-16 Sept 2004, Page(s):137 - 142
[8] Collins, R. T.;Lipton, A. J.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D.; Enomoto, N. and Hasegawa, O.; "A system for video surveillance and monitoring", Technical Report CMU-RI-TR-00-12, CMU, 2000
[9] Comaniciu, D.; Ramesh, V.; Meer, P.; "Real-time tracking of non-rigid objects using mean shift", Computer Vision and Pattern Recognition, 2000
[10] Dirks, W.; Yona, G.; "A comprehensive study of the notion of functional link between genes based on microarray data, promoter signals, protein-protein interactions and pathway analysis", Technical Report, 2003
[11] Elgamal A.; Duraiswami R.; Harwood D. and Davis L.; "Background and foreground modelling using nonparametric kernel density estimation for visual surveillance", Proc of the IEEE, 90, No 7 (July 2002).
[12] Haritaoglu, I.; Harwood, D.; Davis, L. S.; "Hydra: multiple people detection and tracking using silhouettes", International Conference on Image Analysis and Processing, 27-29 Sept. Page(s):280 – 285