Chapter 5

Experiments and Results

5.1 Experimental Setup

We now present some experimental results obtained on real sequences. These results illustrate the tracking algorithm based on motion vectors and DCT Coefficients of color components. The tracking algorithm was used in video sequences shot in our institute premises using a Sony handy cam held in static condition. All sequences are captured at a constant frame rate of 25 frames per sec. The videos were converted to standard MPEG video with 352 X 288 pixel size frame. The sequence of frames in GOP has also kept as standard, i.e. **I-B-B-P-B-B-P-B-B-P-B-B-I.** We performed all the tests in a PC powered with Intel Pentium IV 1.8 GHz processor and 256 MB RAM.

The video uploading is web based through an applet shown in Figure 5.1. The video is then decoded and played in slow motion for the ease of the marker who in turn can freeze it at any desired frame. The user then can mark the object to be tracked in the pixel domain with the help of mouse and given this option of "**TRACK**" as shown in the figure 5.2, where the user has marked the moving car after pausing the playing video at a desired frame.

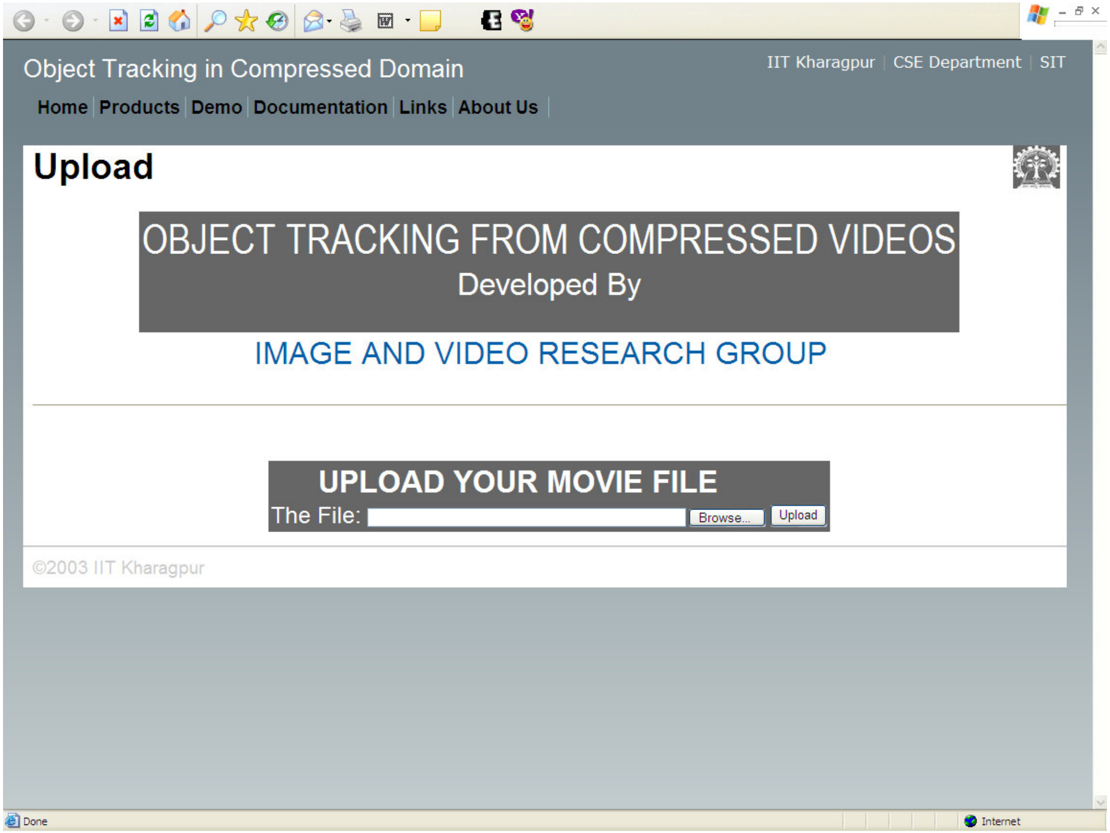


Figure 5.1 Web Based Video Uploading System for Tracking of Objects

The type of accomplished evaluations is always "frame wise" in this thesis. This means, tracking output and ground truth are compared on a frame-by-frame basis. The rectangle coordinates of the tracked objects in the sequence are marked on the original frames using ***Dali 1.0*** package [28], as predicted by the algorithm. The ground truth hence can be seen in the marked sequence of frames.

We have performed large-scale experiments with our object tracking system. The video sequences for experiments were shot in outdoor as well as indoor environments to cater to different situations like change in background, object size, illumination, etc. The sequences were streamed to a Personal Computer. We converted these video clippings into a standard MPEG video. Various kinds of moving objects were used for testing, including cars, slow and fast moving humans as well as multiple objects.

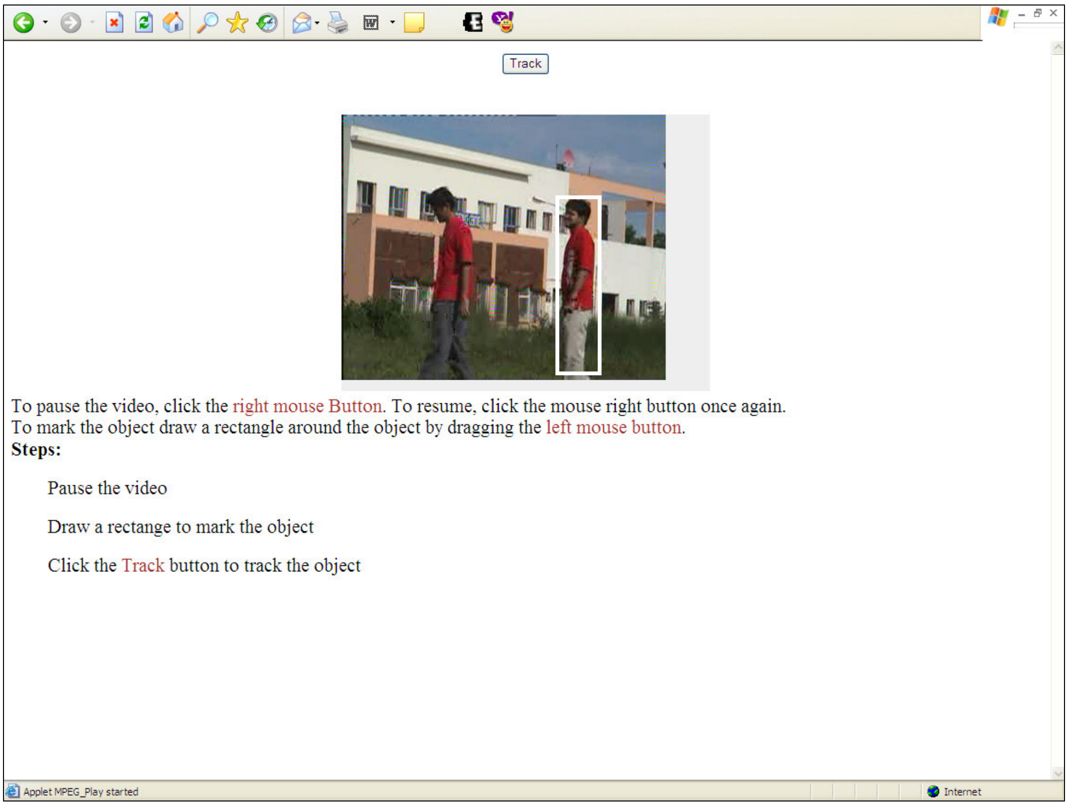
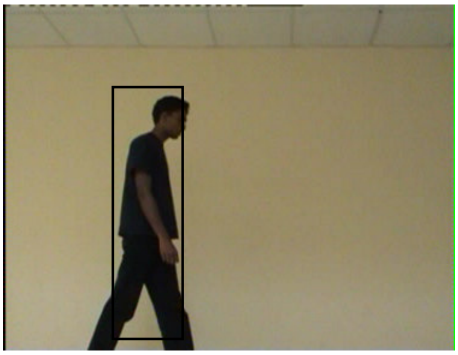


Figure 5.2 Marking of The Object To Be Tracked By The User For Experiment

5.2 Results of Object Tracking using Our Approach

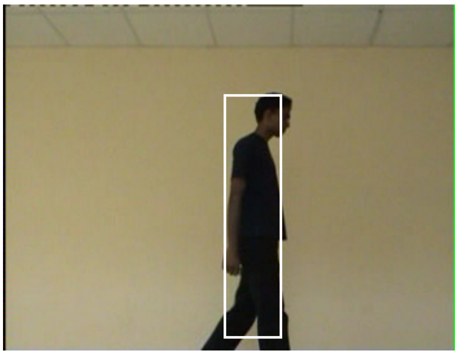
Results of the tracking are shown by, drawing rectangles at the places where it has predicted location of the target. These rectangles have been marked during the decoding process. The decoding process continues after parsing the compressed domain features solely for the purpose of showing the results. The tracked frame sequences of above captured video of a moving person as shown in Figure 5.3, Figure 5.4 and Figure 5.5.

(a) Frame - 2

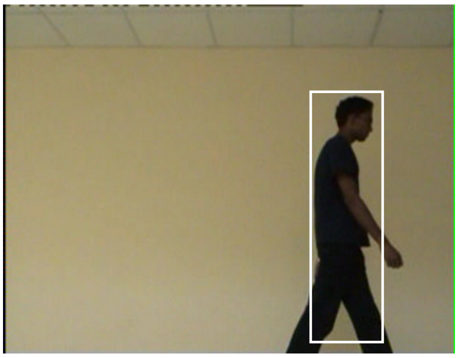


(b) Frame – 51

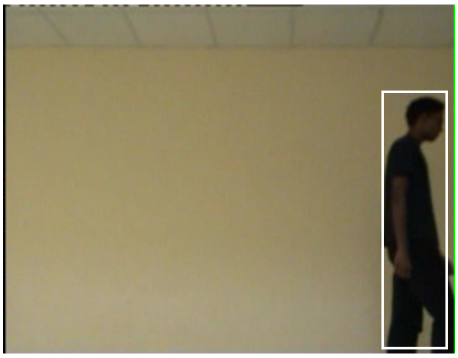
(c) Frame - 60



(d) Frame – 63



e) Frame - 75



(f) Frame – 89

Figure5.3 Tracked Sequence of Moving Person (a) Background frame, (b) User marks the object, (c)-(f) Object is tracked by the system

(a) Frame - 2

(b) Frame – 51

(c) Frame - 59

(d) Frame – 65

e) Frame - 74

(f) Frame – 81

Figure5.4 Tracked Sequence of Moving Person (a) Background frame,

(b) User marks the object, (c)-(f) Object is tracked by the system

(a) Frame - 2

(b) Frame – 62

(c) Frame - 76

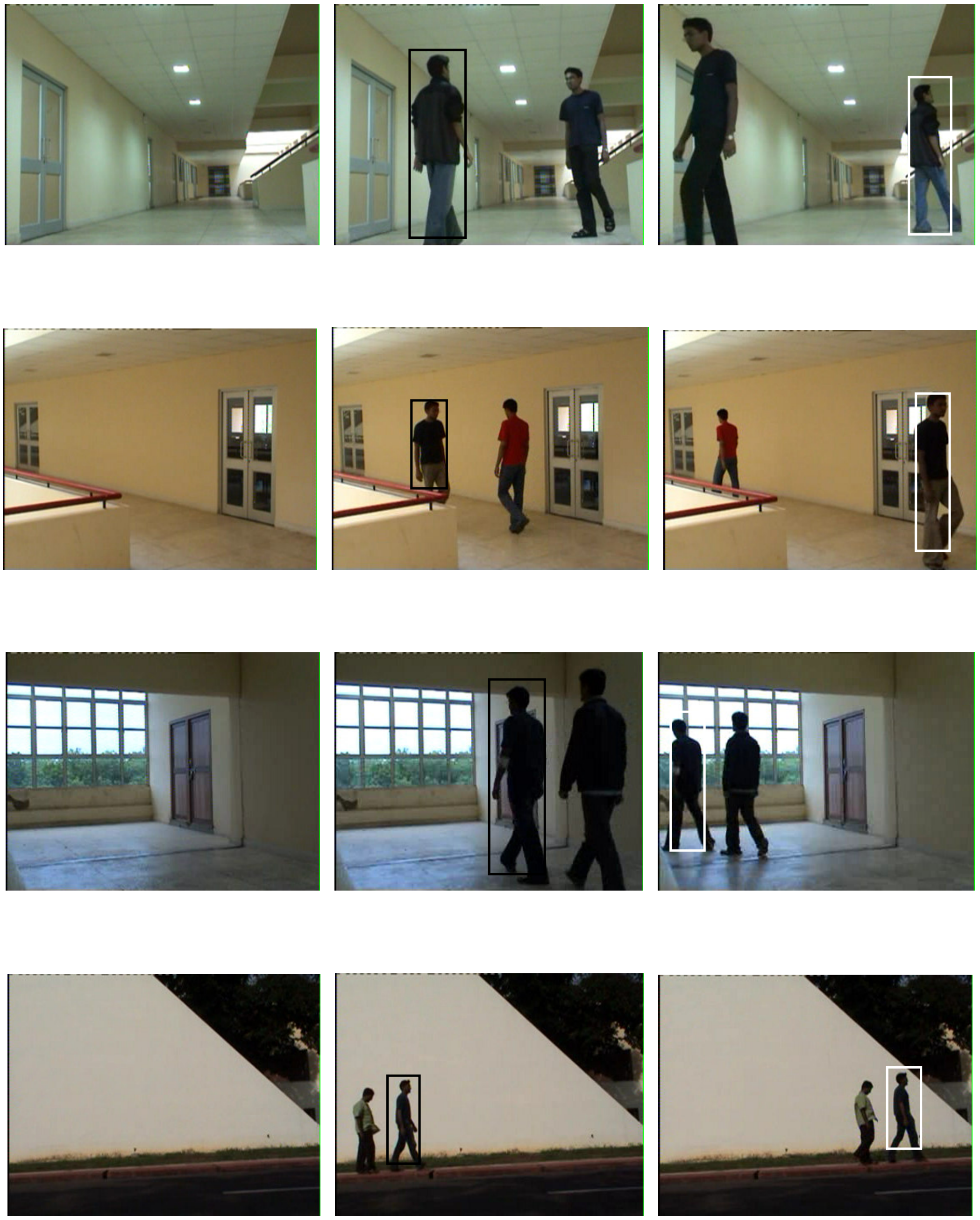
(d) Frame – 81

e) Frame - 84

(f) Frame – 92

Figure5.5 Tracked Sequence of Moving Person (a) Background frame, (b) User marks the object, (c)-(f) Object is tracked by the system

We show results of object tracking in a number of complex situations.



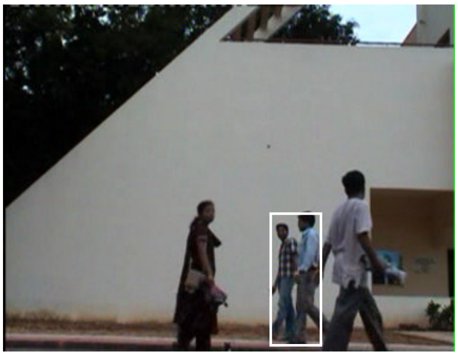
(a) (b) (c)

Figure5.6 Tracked Sequence of Moving Person (a) Background frame, (b) User marks the object, (c)-(f) Object is tracked by the system

The algorithm *did not work* well in few of the cases, an example of such a case is shown in Figure 5.7.

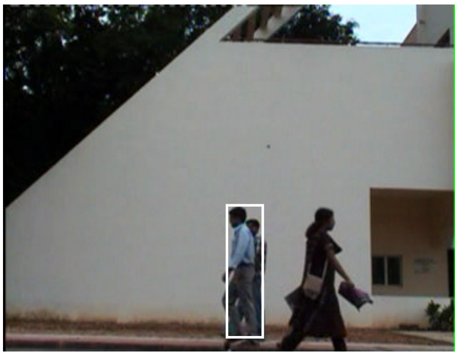
(a) Frame - 2

(b) Frame – 73



(c) Frame - 79

(d) Frame – 87



e) Frame - 92

(f) Frame – 96

Figure5.7 Tracked Sequence of Moving Person (a) Background frame, (b) User marks the object, (c)-(f) Object is tracked by the system

To find the performance metric in terms of the % of frames in which the algorithm tracks correctly, we need to do a frame-by-frame evaluation. This means, tracking output and ground truth are compared on a frame-by-frame basis. The rectangle coordinates of the tracked objects in the sequence are marked on the frames while playing the video by the front-end applet. These window locations were obtained from a file, which is the output of the back-end.

The algorithm was tested with several videos and in most of the cases we could obtain correct result. Though the result was wrong at some frames in between, the algorithm still corrected this error in **I** frames.

Minimum Track Length	**20 Frames**	**40 Frames**	**80 Frames**	**90 Frames**	**200 Frames**
Correct Tracking	17(85%)	35(87.5%)	69(86.25%)	72(80%)	186(93%)
False Tracking	3(15%)	5(12.5%)	11(13.75%)	18(20%)	14(7%)

Table 2 Tracking performance

Quantitative result on accuracy is shown in Table 3. PA denotes the proposed approach of incorporating velocity and acceleration sensitiveness in background subtraction and motion estimation based method and MVBS denotes a background subtraction and motion estimation based method alone. Performance evaluation was done “frame wise” in this study. This means, tracking output and ground truth were compared on a frame-by-frame basis. An object was considered “missed” in a frame by our system if less than 50% of it was tracked correctly. Results have been grouped together under various shooting conditions, like indoor, sunny, cloudy and sunset conditions.

Algorithm	*Environment*			
	Outdoor			*Indoor*
	Sunny	*Cloudy*	*Sunset*	
MVBS	83.5%	78.7%	77.2%	91.2%
PA	93.25%	82.3%	84.6%	94.8%

Table 3 Comparison of accuracy

We also compared the time efficiency of our algorithm with the background subtraction and motion estimation combined approach. The algorithm is found to be pretty fast and it is derived that object tracking using background subtraction and motion estimation combined approach and our proposed approach took almost equal time. Figure 5.8 shows time to perform tracking, in the unit of *time/frame*. We tested with MPEG video files with a varied range of frame numbers. The resolution, in all of these video files, was *352 X 288*.

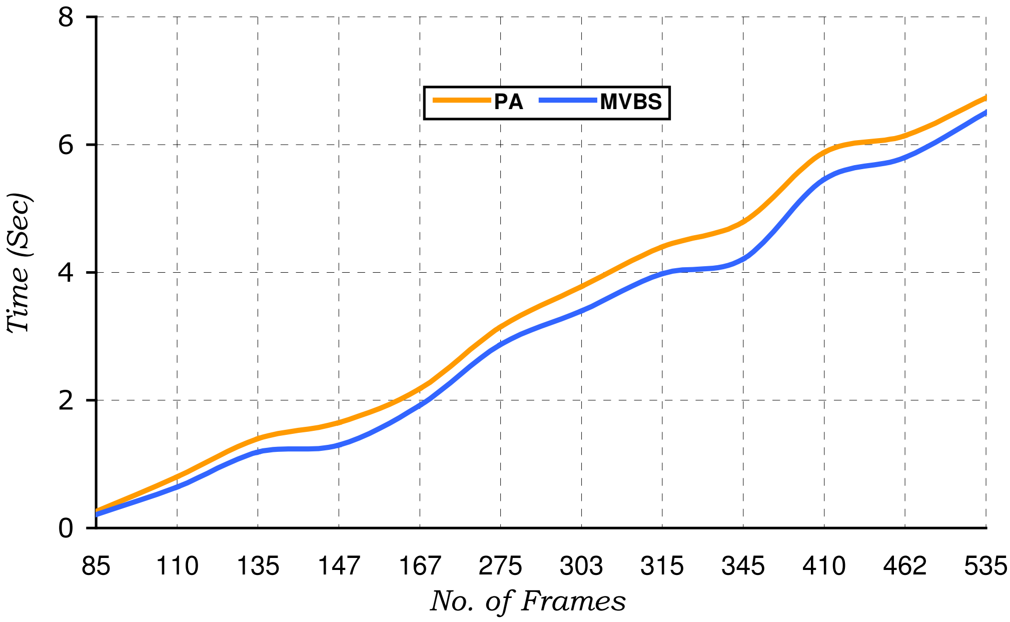


Figure 5.8 Speed comparison