# Chapter 4

# 4 Experimental Results

All the algorithms for visual tracking are implemented in MATLAB 7.11 Release 2010b and executed on the Pentium–IV, 3.00GHz CPU with 1 GB RAM. Image Processing toolbox, Wavelet Toolbox, Neural Network toolbox, Contourlet Toolbox and Curvelet Toolbox available with MATLAB are used.

## 4.1 Datasets

To validate the accuracy and efficiency of the proposed algorithm for object identification, Face dataset and Vehicle dataset are considered. For the visual tracking algorithm different types of sequences available from standard dataset and the different web sources are used. Some of the real time pre-recorded sequences are also implemented for testing the accuracy of the proposed method.

### 4.1.1 Face Dataset and Vehicle Dataset

For face identification two different databases have been used:

Face94 and IIT_Kanpur Dataset. The results for recognition using discrete Curvelet transforms are compared with the discrete Contourlet Transform [23], [28].

### A. IIT_Kanpur Dataset[77]

IIT_Kanpur dataset consists of total 660 male and female images. Total database consists of 22 images of female faces and 38 images of male faces having 40 distinct subjects in up right, frontal position with tilting and rotation. Therefore this is a more difficult database to work with. The size of each image is 640x480 pixels, with 256 grey levels per pixels. For each individual, 3 images have been selected randomly for training and 10 images for testing. Figure 4.1 (a) shows the original image of one female face having different position and tilting. Figure 4.1 (b) shows gray scale images of IIT_Kanpur dataset before filtering.

### B. Face94 Dataset[78]

Face94 dataset consists of total 2660 images. The dataset consists of 20 female and 113 male face images having 20 distinct subject containing variations in illumination and facial expression. For each individual again 3 images have been selected randomly for training and 10 images for testing out of 20 different types of face images. Figure 4.2(a) shows female face image from Face 94 Dataset having different pose and (b) shows some of the images of Face94 Dataset used for training. Figure 4.3 shows the gray scale face images used for testing purpose from faces94 dataset.

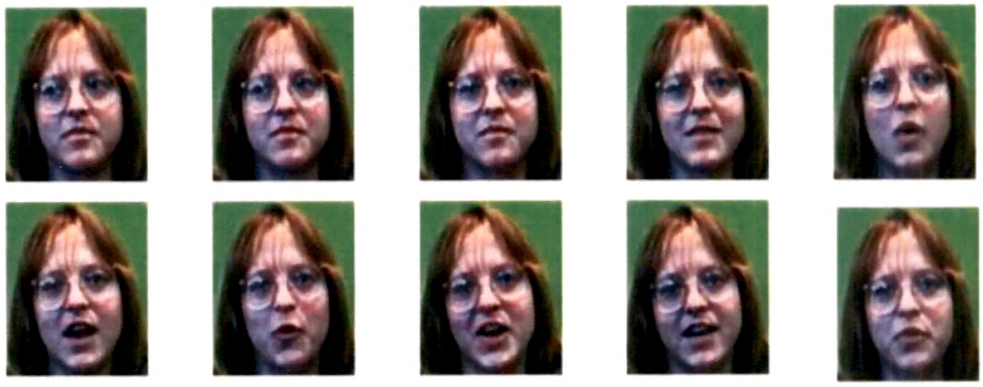
### C. PASCAL VOC 2006 Dataset

To validate the accuracy of the Vehicle Classifier system, different images of the vehicles from Pascal VOC 2006 dataset have been used [81]. Training dataset consists of 300 images of different subjects. VOC dataset contains 10 different classes of dataset they are bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person. Only vehicle dataset from the VOC dataset is used. Some of the vehicle images are downloaded from different commercial websites. Testing dataset consists of 100 real world images. Figure 4.4 shows the images used for testing purpose. The testing dataset is implemented with unsupervised data not used for training.

(a)

(b)

Figure 4.1 : (a) Face Images with Different Position and Tilting (b) Gray Scale Images of IIT Kanpur Dataset



(a)

(b)

Figure 4.2 : (a) Sample Images from Face 94 Dataset having Different Pose (b) Some of the Images of Face94 Dataset used for Training

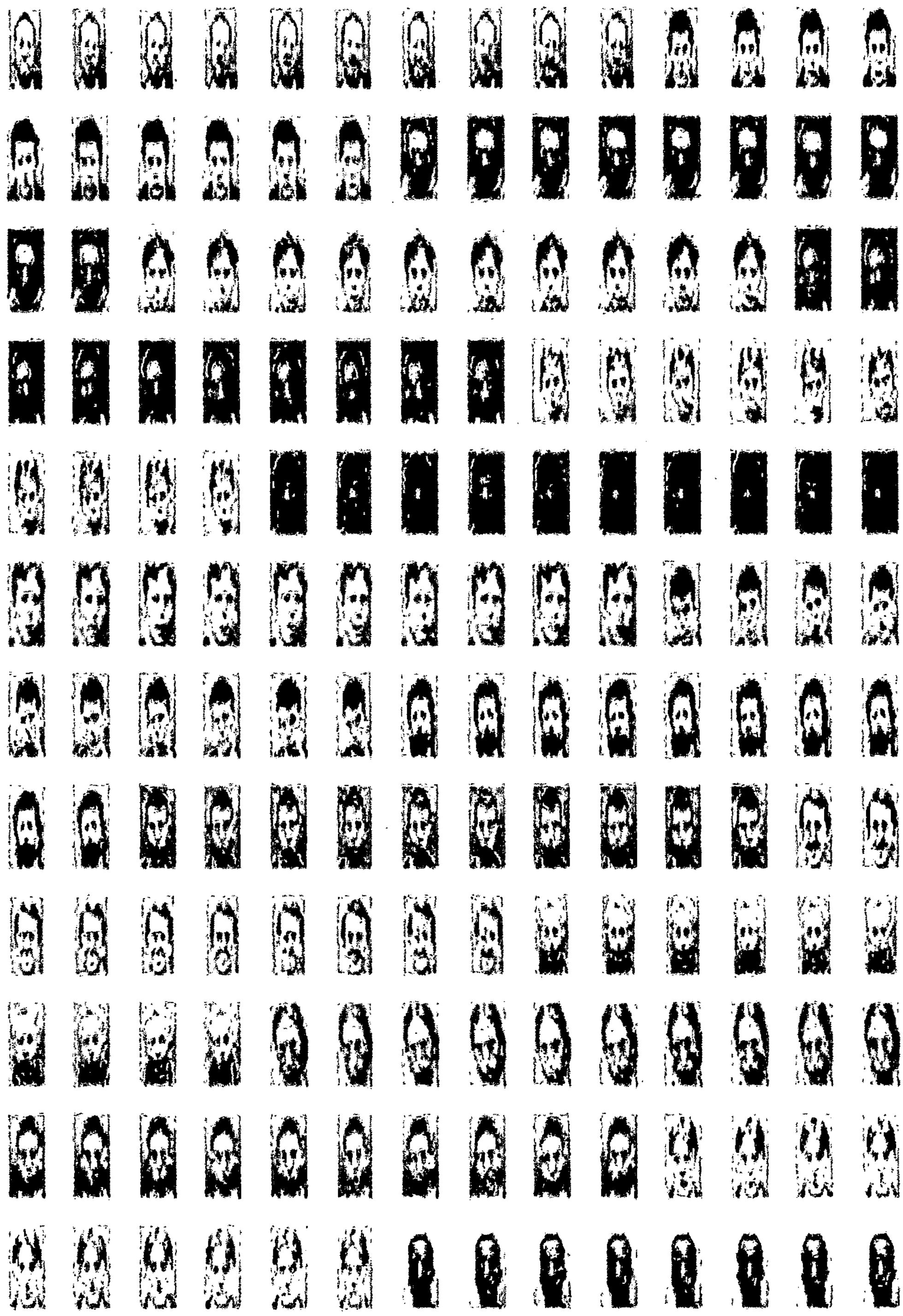


Figure 4.3 : Some of the Gray Scale Images of Face94 Dataset used for Testing

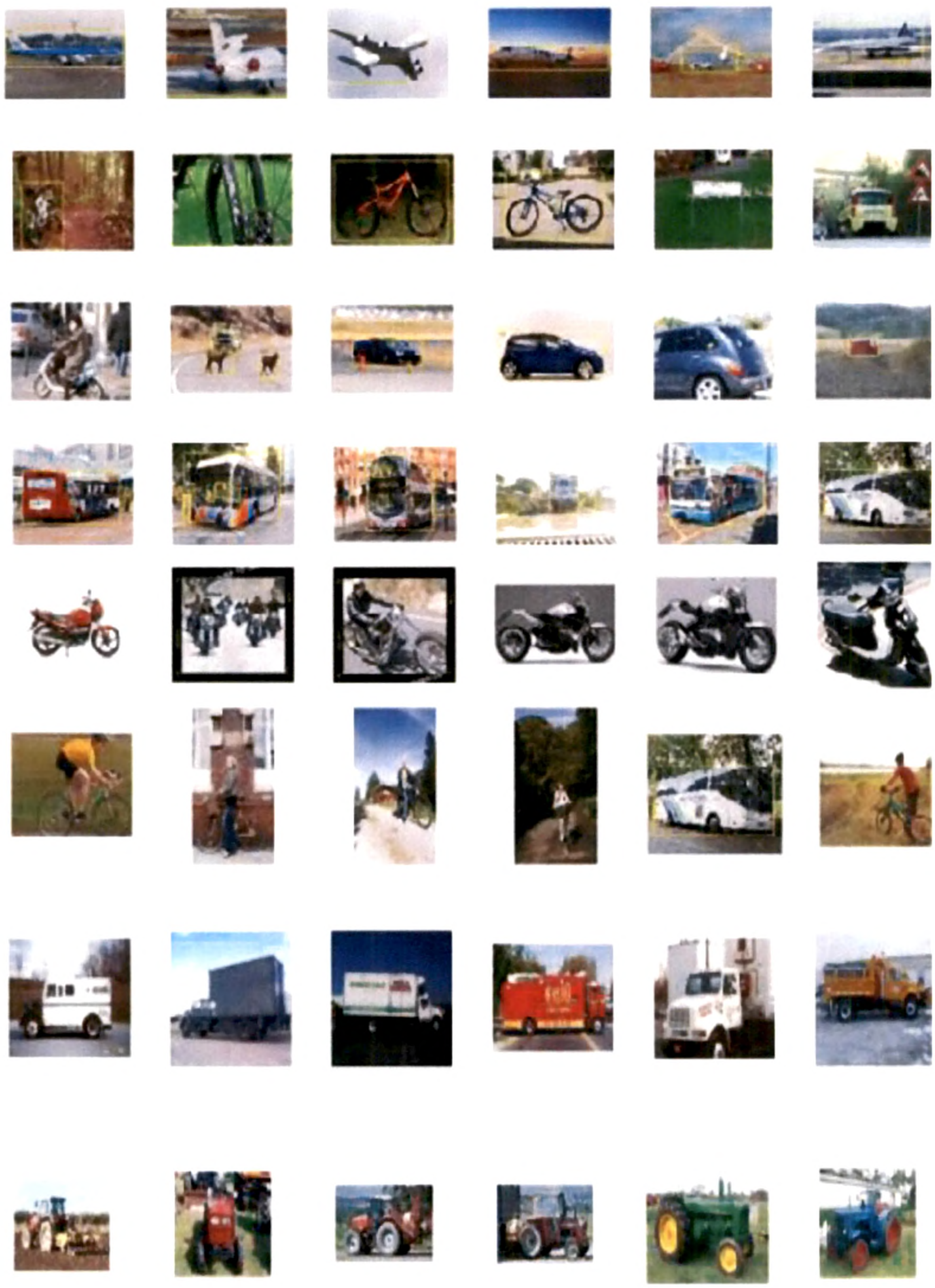


Figure 4.4 : Vehicle Dataset from PASCAL VOC 2006

## 4.1.2 Test Sequences

Different types of available standard test sequences are used to evaluate the proposed tracking algorithm [79] ,[80]. CAVIAR [82], PETS (Performance Evaluation of Tracking and Surveillance) 2000 and PETS 2001 dataset sequences have been used for evaluation. PETS dataset consists of the file in two formats (a) Quick Time movie formation with Motion JPEG compression and (b) individual JPEG files. We selected movie format for practical evaluation. PETS 2000 dataset consists of outdoor people and vehicle tracking sequence using single camera as shown in the Figure 4.5. PETS 2001 consists of five separate sets of training and test sequences. All the datasets are multi-view and are significantly more challenging than in terms of significant lighting variation, occlusion, scene activity and use of multi-view data.

Figure 4.5 : Some of the Sequences from PETS 2000 Dataset used for Visual Tracking

Video Clips from CAVIAR project are used for tracking purpose. These include people walking alone, meeting with each others, window shopping, entering and exiting shops etc. The file sizes of different sequences are between 6 to 12 MB compressed with MPEG2. Apart from the PETS dataset, many sequences available from the internet also have been evaluated. These sequences consist of different format and different conditions of motion. The objects appearing in the sequences are with different size, scale, background and lighting conditions. Three different category of the color image sequences used are (1) Simple Sequence having similar type of foreground and background color (2) object moving near to boundary and then appearing out of frame on ending frames and (3) no motion in all frames. Rainy sequence with bitmap format is used for testing the different boundary conditions.

Multiple Objects tracking algorithm is implemented on traffic sequences of the cars on the Highway. Real time pre recorded road sequences with vehicles also have been implemented for identification of vehicles with tracking.

## 4.2 Camera Modeling Parameters

Motion estimation is the process of determining motion vectors that describe the transformation from one 2D image to another, usually from adjacent frames in a video sequence. The motion estimation module creates a model for the current frame by modifying the reference frames such that it is a very close match to the current frame. The objective of modeling the camera parameters is to estimate the motion vectors from two time sequential frames of the video.

The motion of an object in the 3D object space is translated into two successive frames in the image space at time instants $t_1$ and $t_2$ as shown in Figure 4.6. Translational and rotational motion of the objects can be defined in temporal frames using this model.

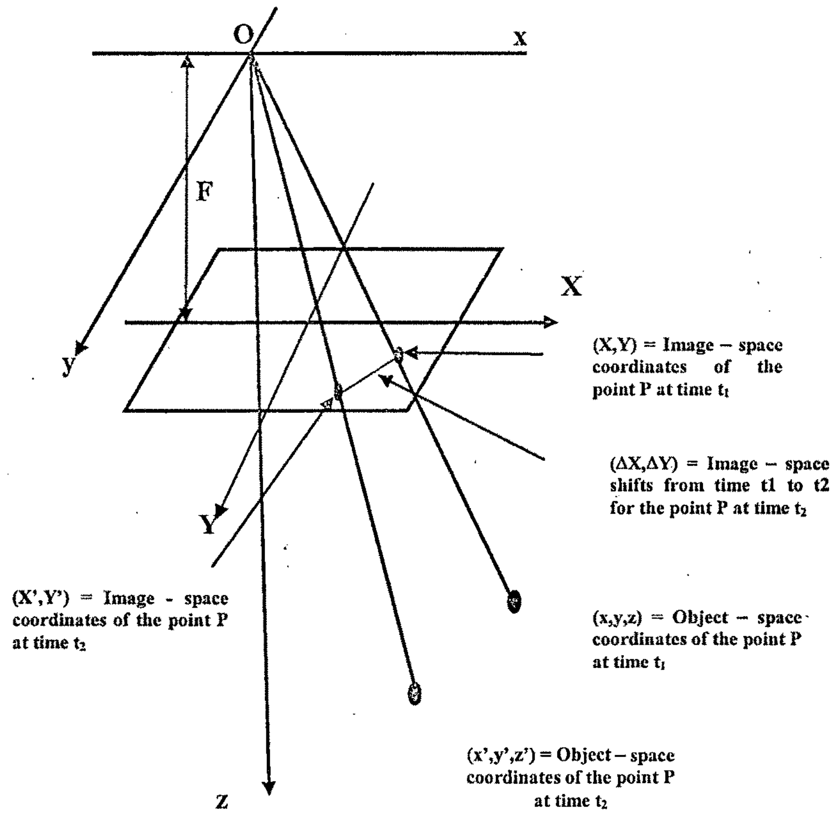



Figure 4.6 : Basic Geometry Model of the Object in 3- D Space

In the Figure 4.6,

$t_1$, $t_2$ represent the time axis such that $t_2 > t_1$.

(X, Y) are the Image space coordinates of P in the scene at time $t_1$

(X', Y') are the Image space coordinates of P at time $t_2$

(x, y, z) are the Object space coordinates at a point P in the scene at time $t_1$

(x', y', z') are the Object space coordinates at a point P in the scene at time $t_2$

The output of the motion-estimation algorithm comprises of the motion vector for each block, and the pixel value differences between the blocks in the current frame and the "matched" blocks in the reference frame.

Different technical parameters of the camera [72] used for motion estimation are considered as follows:

- **Focal Length**

Rays from infinite distance objects are condensed internally in the lens at a common point on the optical axis. The point, at which the image sensor of the CCTV camera is positioned, is called a focal point. Designing of lenses have two principal points, a primary principal point and a secondary principal point. As shown in the Figure 4.7, the distance between the secondary principal point and the focal point (image sensor) determines the focal length of the lens.

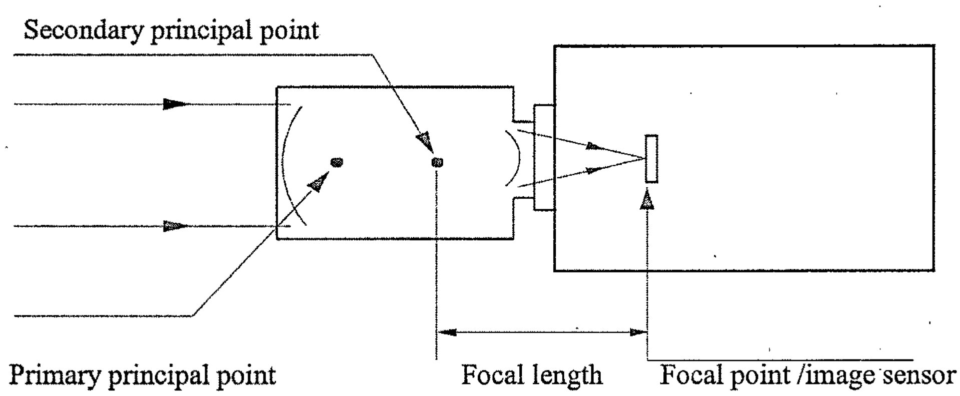



Figure 4.7 : Focal Length in Camera model

- **Angle of View**

The angle formed by the two lines from the secondary principal point to the image sensor is called the angle of view shown in the Figure 4.8. Therefore the focal length of the lens is fixed regardless of the image format size of the CCTV camera.

The Angle of view changes with the focal length of the lens and with the image sensor size of the camera as shown in the Table 4.1. Figure 4.9 shows the effect of angle of view for different focal lengths.

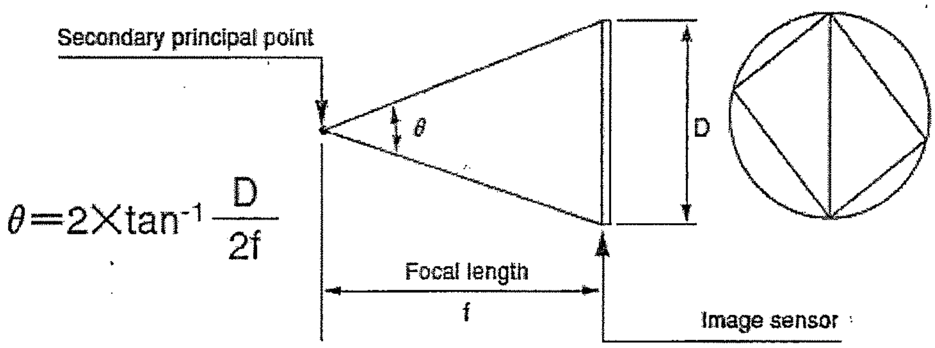



Figure 4.8 : Angle of View in image sensor

The focal length to cover the object can be calculated using the following equation:

$$f = v \times \frac{D}{V} \quad \textbf{(4.1)}$$

$$f = h \times \frac{D}{H} \quad \textbf{(4.2)}$$

f : focal length of the lens
V : Vertical size of the object
H : Horizontal size of object
D : Distance from lens to object
v : vertical size of image
h : horizontal size of image

Table 4.1 : Camera Format

| **FORMAT** | **$^2/_3$ inch** | **$^1/_2$ inch** | **$^1/_3$ inch** | **$^1/_4$ inch** | **$^1/_8$ inch** |
|---|---|---|---|---|---|
| v (mm) | 6.6 | 4.8 | 3.6 | 2.7 | 0.7 |
| h (mm) | 8.8 | 6.4 | 4.8 | 3.6 | 1.6 |

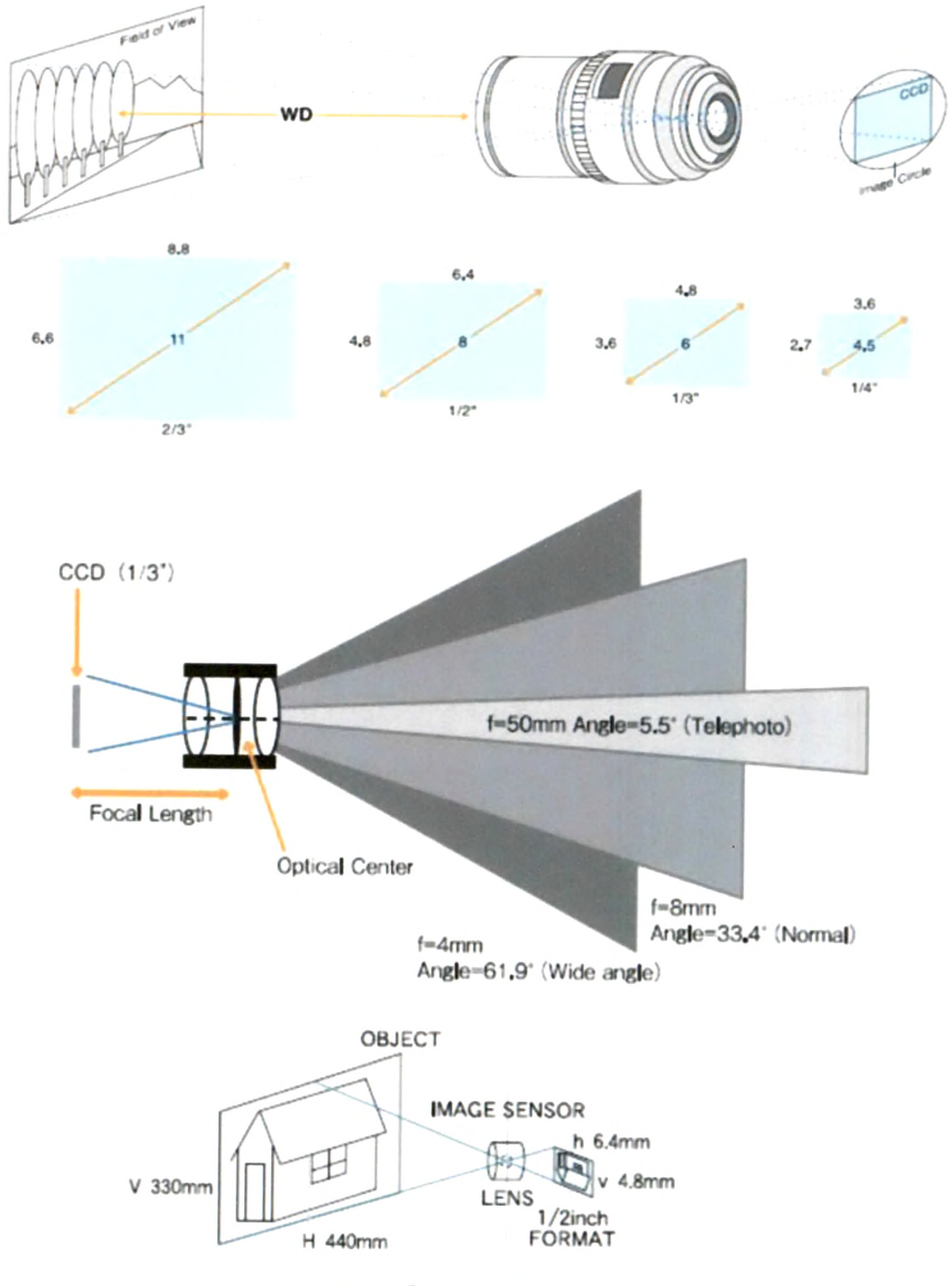



Figure 4.9 : Angle of View in CCTV Camera

- **F-Number**

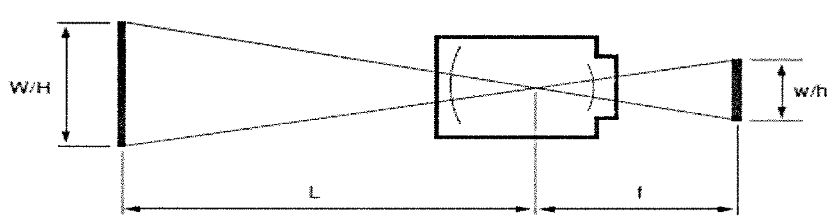
The F number is the index for the amount of light that passes through a lens. Smaller the number greater the amount of light passes through lens. The F number is a ratio between focal length and effective aperture as follows:

$$\text{F Number} = \frac{f}{D} \qquad \textbf{(4.3)}$$

Where $f$ is the focal length, D is the effective Diameter of the lens.

- **Field of View**

The field of view varies along with the focal length of the lens as shown in the Figure 4.10.




$$\frac{w}{W} = \frac{h}{H} = \frac{f}{L}$$

Figure 4.10 : Field of View

W : width of the object

H : height of the object

w : width of the format

$^1/_2$ format = 6.4 mm, $^1/_3$ format = 4.8 mm, $^1/_4$ format = 3.6 mm

h : height of format

$^1/_2$ format = 4.8 mm, $^1/_3$ format = 3.6mm, $^1/_4$ format = 2.7 mm

f : focal length

L : object distance

- **Depth of the field**

When an object is focused, it is observed that the area in front and behind the object is also in focus. The range in focus is called depth of the field. When the background is extended to infinity, the object distance (focusing distance) is called hyper focal distance. Depth of the field is calculated using the following formula.

$$H = \frac{f^2}{C \times F} \tag{4.4}$$

$$T1 = \frac{B\,(H + f)}{H + B} \tag{4.5}$$

$$T2 = \frac{B\,(H - f)}{H - B} \tag{4.6}$$

F : F Number
H : hyper focal distance
f : focal length
B : object distance (measured from image sensor)
T1 : near limit
T2 : far limit
C : circle of least confusion $1/_2$ format = 0.015 mm, $1/_3$ format = 0.011 mm, $1/_4$ format = 0.008 mm

Depth of field increases when

- Focal length is shorter
- F – number is larger
- Object distance is longer

- **Camera Format**

The size of camera's imaging device (image sensor) affects the angle of view, the smaller devices create narrower angles of view when used on the same lens. Lenses are specified as designed for a particular sensor size. On the surface of the image sensor, there are millions of photosensitive diodes, called photosites, each of which captures a single pixel of the photograph to be captured. Cameras with larger sensors and larger pixels collect more light given the lens with same F- number and field of view. Figure 4.11 shows the sensor sizes to be used when calculating fields of view and angles of view.

There are many parameters that can be used to evaluate the performance of an image sensor, which includes dynamic range, signal-to-noise ratio, low-light sensitivity, etc. For sensors of comparable types, the signal-to-noise ratio and dynamic range improve as the size increases.

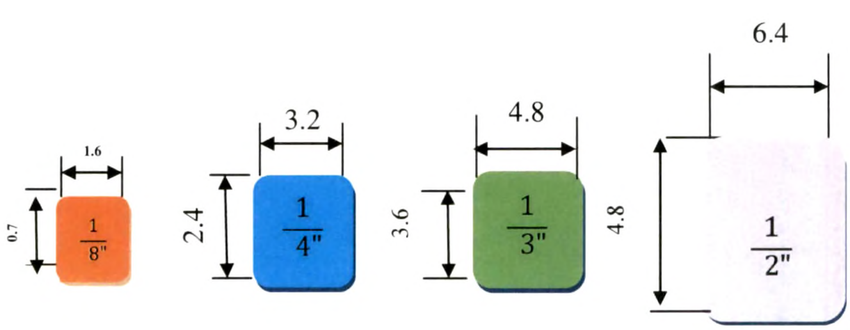



Figure 4.11 : Image Sensor Size

The focal length of the lens is measured in millimeter (mm) and directly relates to the angle of view that will be achieved. Short focal length provides wide angle of view and long focal length provides the narrow angle of view. A normal angle of view is similar to what we see with our own eye and has a relative focal length equal to that of the pickup device.

- **Motion Estimation by camera :**

To find the vehicle speed, successive frame images of the camera can be used. In this case, only the instantaneous speed can be found. This instantaneous speed is computed as follows [73]:

$$\nu = \frac{\Delta \mathrm{p}}{\Delta t} \qquad \textbf{(4.7)}$$

where $\nu$ is instantaneous velocity vector of a point projected on 2D image space and $\Delta$p is the displacement vector of that point in 2D image space.

The displacement vector expresses the spatial displacement of a point during the time interval $\Delta$t. The time interval $\Delta$t is equal to the time which passes between two successive video frames and is equal to the frame replay rate (or frame capture rate) of the camera. In the proposed method frame capture rate of 30 fps (frame per second) is used. So the value of $\Delta$t to be used in equation (4.7) is 33.3 milliseconds.

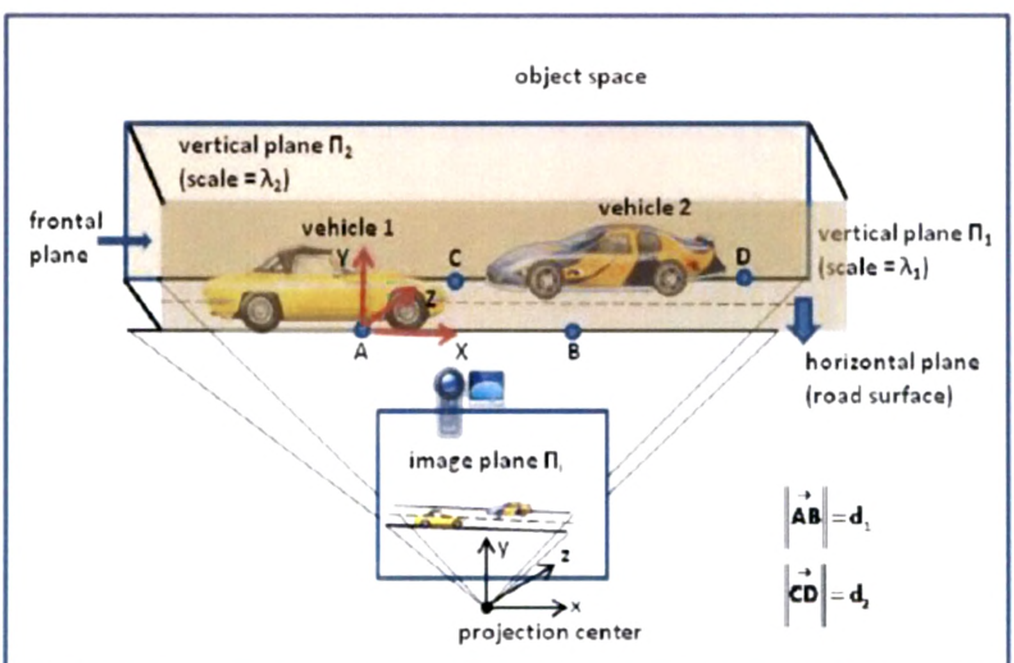



Figure 4.12 : View of Image Acquisition Plane [72]

In order to find the absolute values of displacement vectors or velocity vectors in object space, the vectors computed in video image coordinate system should be transformed to the object coordinate system which is in the object space. It is assumed that the observed scene is flat. In ideal situation, the flat scene must be just as vertical planes as in the Figure 4.12 [72]. The distances from the camera to the vertical planes are different because of the different depths. This difference causes the planes to have different scales in the image plane. On the other side, with only one camera and one image, it is not possible to detect the depths and indirectly the scales of the planes on the image.

To calculate the different scales, let us assume that the vehicle is moving from left to right as vehicle 1 as in Figure 4.12. Then the visible side of the vehicle is right side and it is closer to plane $\Pi_1$ with the scale $\lambda_1$. Scales of the vertical planes $\Pi_1$ and $\Pi_2$ are obtained with the measured distances $d_1$ , $d_2$ and their corresponding distances and on the image plane such that $\lambda_1 = \frac{d_1'}{d_1}$ and $\lambda_1 = \frac{d_2'}{d_2}$ respectively. In the similar way let's assume that the vehicle is moving from right to left. Then its visible side is the left side and it is closer to centre of the road axis. In this case, the scale can be taken as $\lambda = \frac{(\lambda_1 + \lambda_2)}{2}$. According to this configuration and assumptions, if the ideal situation is achieved, then absolute values of the velocity vectors or displacement vectors can be obtained by using the corresponding scale factors. The scale of the camera in the proposed method can be calculated using magnification ratio of the object space to the image space. That is used to find the absolute velocity of the actual object. The distance between two points can be measured either by physical measurement or using the format specified in table 4.1.

## 4.3 Classification and Tracking Results

### 4.3.1 Object Classifier

In order to assess the efficiency of the Object classifier, series of experiments have been carried out using face94 dataset and IIT_Kanpur dataset using Euclidian distance and Neural Network Classifier. Pre-processing stage has been applied to the image dataset. Pre-processing stage includes Unsharp Filtering, Thresholding and Morphological operations. Pre-processed image results have been shown as Figure 4.13. The performance results are obtained for all database using Contourlet-PCA / Curvelet-PCA and both the Classifier. Results of Contourlet-PCA and Curvelet-PCA have been shown in the Figure 4.14. Eigen matrix has been calculated for dimensionality reduction and feature matching. Poor results have been observed in the 6 types of faces from IIT Kanpur male dataset due to the number of variations in the faces. Table 4.2 shows comparative performance of the images with Contourlet Transform. The results of Contourlet transform with PCA using our proposed method gives better result than the discrete Curvelet transform with pre processing and without pre-processing. The Table 4.3 reports the time required to calculate the Curvelet transform and Contourlet transform. The Discrete Contourlet transform is faster than the discrete Curvelet transform.

Figure 4.13 : Images after applying Pre-processing Stage

(a)

(b)

Figure 4.14 : (a) Eigenfaces using Contourlet–PCA after Pre–processing Stage

(b) Eigenfaces using Curvelet–PCA after Pre–processing Stage

Table 4.2 : Recognition Rate for Object Classifier System

(a) Recognition Rate using Discrete Contourlet Transform

| Dataset (JPEG Image) | Size of the Image (Pixel) | Contourlet Transform without Pre-processing Euclidean Classifier ( % ) | Contourlet Transform with Pre-processing Euclidean Classifier ( % ) | Contourlet Transform With Pre-processing Neural Network Classifier ( % ) |
|---|---|---|---|---|
| Faces_94 female | 180 × 200 | 92.57 | 97.27 | 90.90 |
| Faces_94 Male | 180 × 200 | 93.24 | 98.24 | 87.05 |
| IIT_Kanpur Female | 640 × 480 | 91.5 | 96 | 88 |
| IIT_Kanpur Male | 640 × 480 | 75.65 | 82 | 82 |

(b) Recognition Rate using Discrete Curvelet Transform

| Dataset (JPEG Image) | Size of the Image (Pixel) | Curvelet Transform without Pre-processing Euclidean Classifier ( % ) | Curvelet Transform with Pre-processing Euclidean Classifier ( % ) | Curvelet Transform With Pre-processing Neural Network Classifier ( % ) |
|---|---|---|---|---|
| Faces_94 female | 180 × 200 | 93..20 | 97.33 | 90.90 |
| Faces_94 Male | 180 × 200 | 94.6 | 91.76 | 79 |
| IIT_Kanpur Female | 640 × 480 | 90.55 | 90 | 80 |
| IIT_Kanpur Male | 640 × 480 | 74.8 | 78 | 61.6 |

Table 4.3 : Execution Time required for Training and Testing of Face Images.

(a) Discrete Contourlet Transform

| Dataset (JPEG Image) | Pre-processing Time (seconds) | Contourlet Transform Euclidean Distance Classifier | | Contourlet Transform Neural Network Classifier | |
|---|---|---|---|---|---|
| | | Training Time for Dataset (seconds) | Testing Time/Face (seconds) | Training Time for Dataset (seconds) | Testing Time/Face (seconds) |
| Faces_94 female | 30.54 | 86.35 | 1.53 | 92.45 | 0.98 |
| Faces_94 Male | 37.10 | 88.23 | 1.54 | 93.06 | 1.11 |
| IIT_Kanpur Female | 15.98 | 46.35 | 2.18 | 52.37 | 1.52 |
| IIT_Kanpur Male | 16.30 | 50.05 | 2.32 | 56.02 | 1.54 |

(b) Discrete Curvelet Transform

| Dataset (JPEG Image) | Pre-processing Time (seconds) | Curvelet Transform Euclidean Distance Classifier | | Curvelet Transform Neural Network Classifier | |
|---|---|---|---|---|---|
| | | Training Time for Dataset (seconds) | Testing Time/Face (seconds) | Training Time for Dataset (seconds) | Testing Time/Face (seconds) |
| Faces_94 female | 30.54 | 154.07 | 1.90 | 160.23 | 1.23 |
| Faces_94 Male | 37.10 | 184.63 | 2.13 | 190.05 | 1.56 |
| IIT_Kanpur Female | 15.98 | 61.10 | 2.35 | 65.89 | 1.67 |
| IIT_Kanpur Male | 16.30 | 63.61 | 2.65 | 69.60 | 1.89 |

To validate the accuracy of the vehicle classifier system, different images of the vehicles from Pascal VOC 2006 dataset have been used. Vehicle dataset consists of 300 images used for training. VOC dataset contains 10 different classes of dataset that are bicycle, bus, car, motorbike, cat, cow, dog, horse, sheep and person Figure 4.15 shows some of the images considered for training. Testing dataset consists of 100 real world images. Figure 4.16 shows the enhanced images after performing pre-processing on the VOC dataset. The testing dataset is considered as unsupervised data not used for training. The results of the recognition of vehicle using discrete Curvelet transforms with Pre-processing and without Pre-processing has been compared as per Table 4.4. The proposed method gives better and fast recognition results compared to all other three methods for vehicle dataset also.

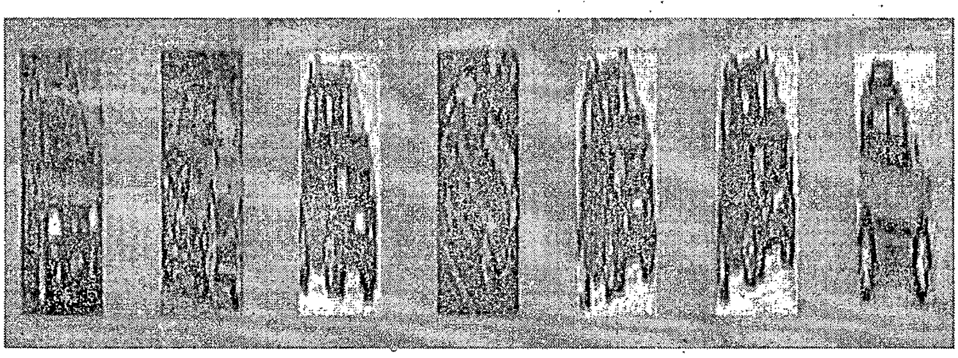


Figure 4.15 : Vehicle Images from the VOC 2006 Dataset

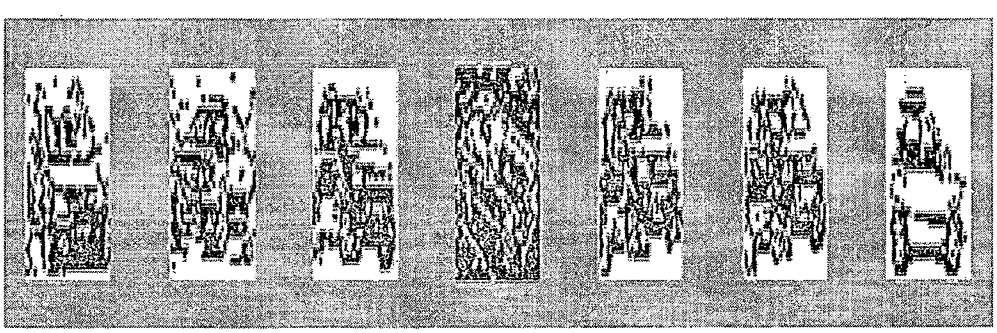


Figure 4.16 : Enhanced Images after performing Pre-processing on VOC 2006 Dataset

Table 4.4 : Performance Evaluation for VOC 2006 Dataset

(a) Recognition Rate using Discrete Contourlet Transform (in %)

| Dataset (JPEG Image) | Size of the Image | Feature matrix Created for training Contourlet transform | Contourlet Transform without Pre-processing Euclidean Classifier ( % ) | Contourlet Transform with Pre-processing Euclidean Classifier ( % ) |
|---|---|---|---|---|
| Vehicle Image | 160 × 120 | 4096 × 300 | 22 | 42 |

(b) Recognition Rate using Discrete Curvelet Transform (in %)

| Dataset (JPEG Image) | Size of the Image | Feature matrix Using Curvelet Transform | Curvelet Transform without Pre-processing Euclidean Classifier ( % ) | Curvelet Transform with Pre-processing Euclidean Classifier ( % ) |
|---|---|---|---|---|
| Vehicle Image | 160 × 120 | 7225 × 300 | 18 | 36 |

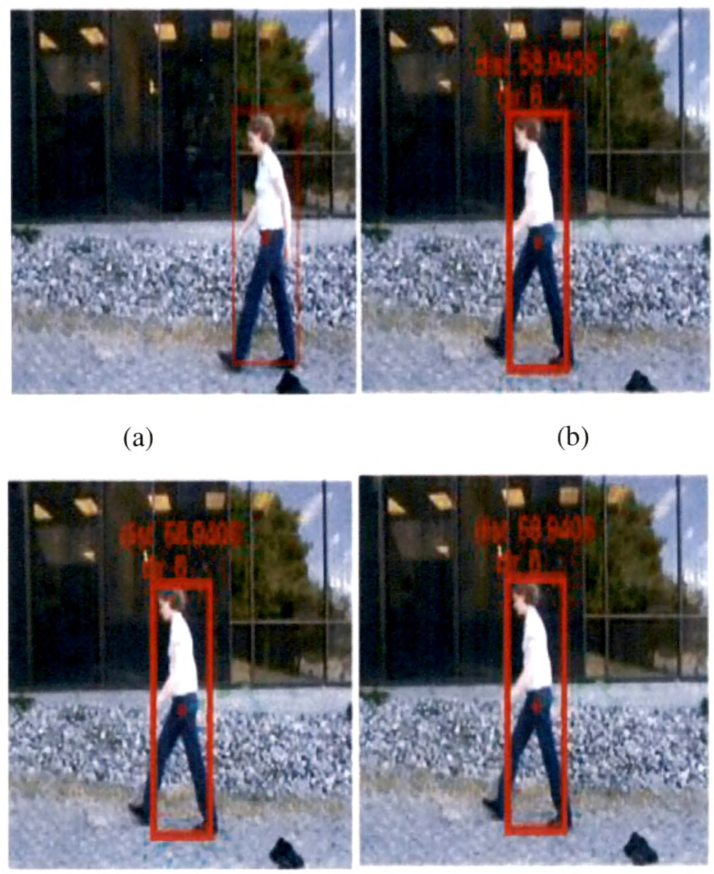
## 4.3.2 Visual tracking System

To check the performance of the proposed method, many real time pre-recorded sequences for single object tracking as well as multiple objects tracking have been used.

### 4.3.2.1 Single Visual Tracking System

In the single visual tracking system, the tracking object is selected by the person in the first frame. Tracking system track the same objects in other frames and finally converted into movie.

First Sequence [79] 'Girl_walking' with 320x240 dimensions of each frame has been used. The frames are randomly selected to prove the efficiency of proposed algorithm. The tracking result of the 'girl' sequence is shown in the Figure 4.17.

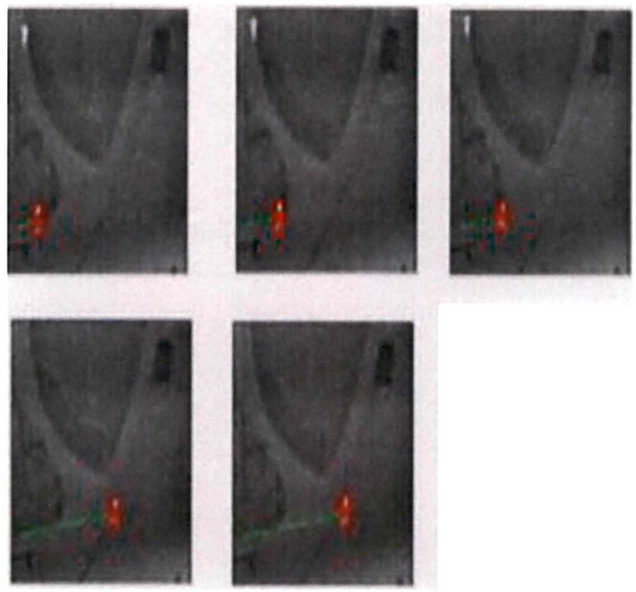


(a) (b)

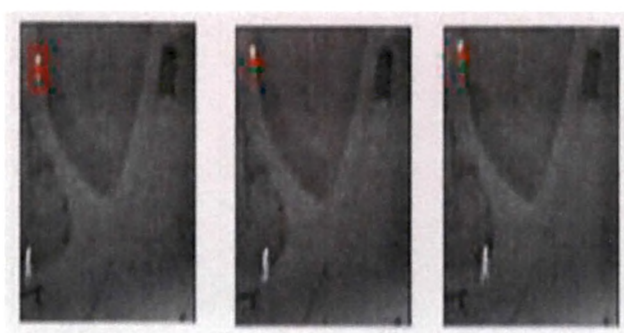
(c) (d)

Figure 4.17 : 'Girl_walking' Sequence and Tracking Results in Different Frames

Another sequence, 'Rain' with 'bitmap' format is shown in the Figure 4.18 having 320x240 dimensions of each frame. This sequence is used for checking the performance of proposed tracking algorithm in poor lighting and rainy condition under the outdoor environment. This sequence is used to track the person1 as shown in the Figure 4.18 (a) having normal motion appearing from first frame to end frame and to track the person2 as shown in the Figure 4.18 (b) appearing from in between frames and disappeared after some frames. Table 4.5 shows the execution time required to track the single object.



(a)



(b)

Figure 4.18 : (a) Tracking Results of Person 1 in 'Rain' Sequence (b) Tracking Results of Person 2 with Boundary Termination Conditions

The proposed method shows better results compared to the standard Mean shift method. Figure 4.19 and Figure 4.20 show the results of tracking with Mean shift method and using Proposed Method. The method is based on the 3D color histogram. So it fails for the sequence with the drastic change in the back ground color and foreground color. Figure 4.21 shows the result of "Helicopter" and "Fight and runaway" sequences fails to track the object after some frame. Table 4.6 reports the sequences used to track the single object visual tracking system.

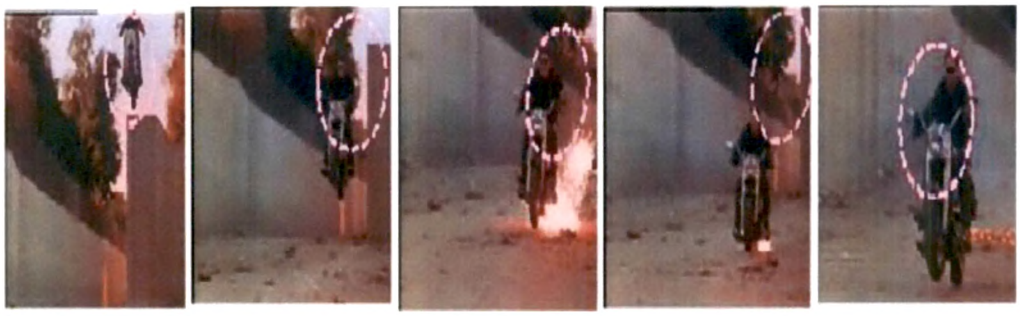
Table 4.5 : Performance Evaluation of Image Sequences

| Sr. No | Name of the sequence | Size of the object selected in first frame (Pixels) | Tracking time to track the selected object (second) |
|---|---|---|---|
| 1 | Girl_walking | 70 × 150 | 64.9060 |
| 2 | Cow _motion | 230 × 150 | 7.2030 |
| 3 | Cow_nomotion | 230 × 150 | 5.6250 |
| 4 | Rain (Person 1) | 15 × 24 | 14.3520 |
| 5 | Rain (Person 2) | 20 × 27 | 9.1570 |

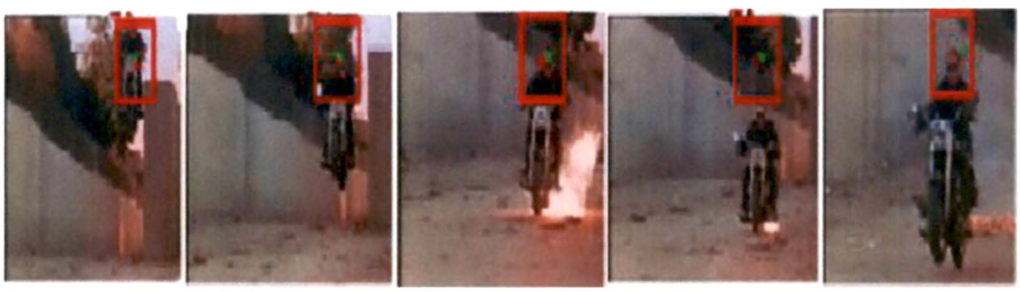
(a)

(b)

Figure 4.19 : Tracking Results of Pedestrian (a) Using Mean shift Method (b) Using Proposed Method

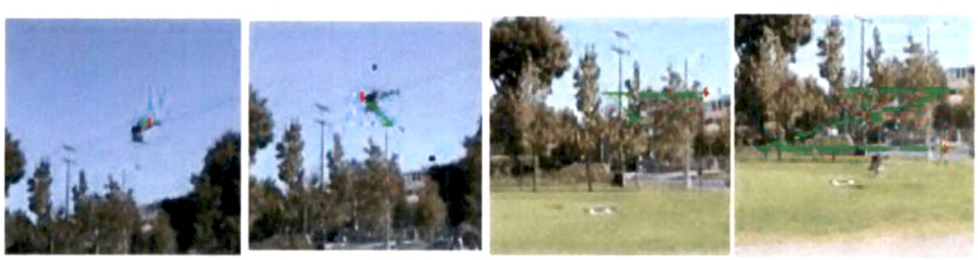


(a)

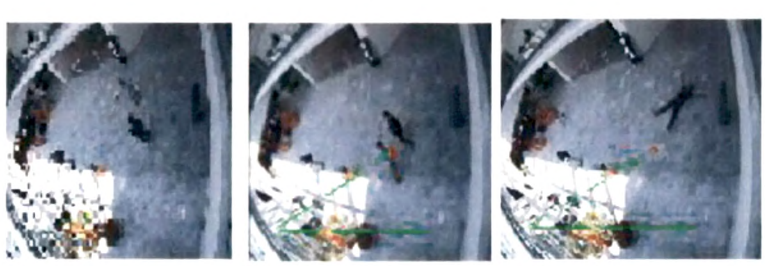


(b)

Figure 4.20 : Tracking Results of bike sequence (a) Using Mean Shift Method
(b) Using Proposed Method



(a)

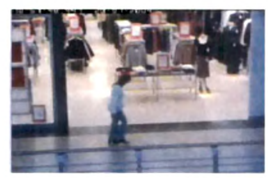


(b)

Figure 4.21 : Tracking Failure (a) Helicopter Sequence – Frame 301,401,501,601
(c) Fight and Runaway Sequence – Frame 121,321,521

Table 4.6 : Sequences used for Single Object Tracking

| Sr. No | Name of the sequence | Format | Total Frames in the sequence | Sequence |
|---|---|---|---|---|
| 1 | Traffic1 | avi | 2851 |   |
| 2 | Pedestrian | avi | 881 |   |
| 3 | Fight Run Away | mpeg | 551 |  |
| 4 | Fight one man down | mpeg | 950 |  |
| 5 | Showroom | avi | 800 |   |

| | | | | |
|---|---|---|---|---|
| 6 | Corridor | mpeg | 383 |  |
| 7 | Leave shop two | mpeg | 600 |  |
| 8 | Shop1front | mpeg | 2360 |  |
| 9 | Bike | mpeg | 150 |  |
| 10 | Car | mpeg | 3381 |  |

### 4.3.2.2 Multiple Objects Tracking

Multiple Objects Tracking algorithms have been implemented on car traffic sequence on Highway. Multiple vehicles are tracked efficiently. Multiple objects tracking cover the background subtraction, blob statistics, region extraction and region matching steps. Figure 4.22 shows the result of blob extraction, and different object tracking with different color boundary. Region tracking is performed by matching the color features. For color features 3D histogram and Hu's seven invariant moments are used. Hybrid tracker is used to increase the performance of tracking. The region having same color features can be tracked using Contourlet with PCA algorithm, which are extracted for object identification purpose. This serves dual purposes one for identification of the object and other for region tracking. This increases the speed for execution of the algorithm. The algorithm also indicates the speed in terms of pixel and direction with respect to the previous frame in terms of angle as shown in the Figure 4.23.

Motion parameters are extracted using camera modeling parameter for motion estimation with the help of equation (4.1) to equation (4.7). Object classifier has been also implemented to identify the object with the motion. In this sequence vehicle classifier is used to identify the vehicle moving on the road. Visualized results with the motion parameters are shown in the Figure 4.24. As shown in the Figure 4.24, proposed algorithm visualizes the result with bounding box in each frame and shows the tracking result in the movie format. Figure 4.25 shows the object classifier system output for vehicle. Object identification task gives almost 94 % correct results for vehicle identification. Table 4.7 shows the sequences used for multiple object tracking.

(a) (b) (c)

Figure 4.22 : Vehicle Tracking in the viptraffic Sequence (a) Tracking Vehicles (b) Blob Extraction (c) Region Tracking with Motion Parameters

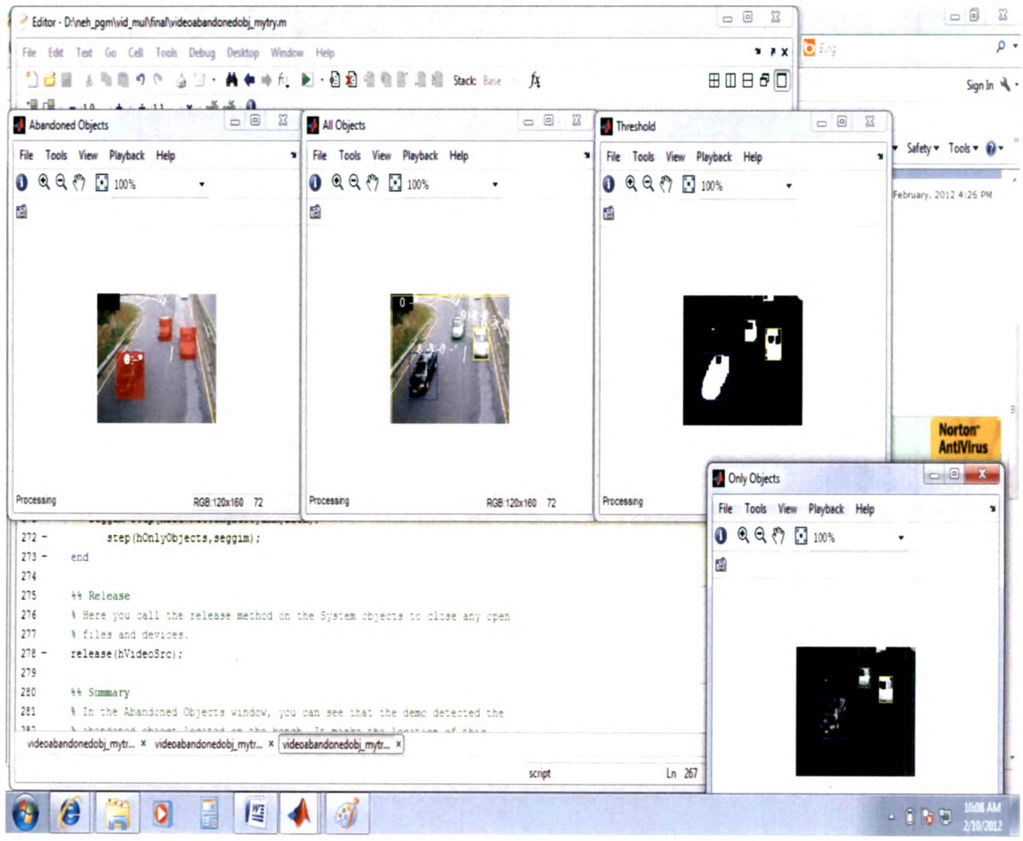


Figure 4.23 : Visual Movie Frame for Proposed Algorithm

(a) (b)

Figure 4.24 : Visualized Results in the Format [Object Number: - Speed: - Direction]
(a) Frame Number 72 (b) Frame Number 113


(a)


(b)

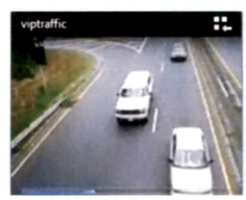
Figure 4.25 : Object Classifier (a) Correct Identification (b) False Identification

## 4.4 Performance Analysis of the Proposed Algorithm

In order to measure the performance of the algorithm, the program has been developed to measure the ground truth of a video sequence. Ground truth refers to the actual presence of the object motion as a human viewer interprets it. Once the ground truth is known for a sequence, the performance of the system in detecting object motion can be evaluated.

In the proposed system, ground truth is annotated by running the detector on a pre-recorded video sequence with the mouse click and labeling each frame.

Table 4.7 : Some of the Sequences used for Multiple Objects Tracking

| Sr. No | Name of the sequence | Format | Sequences |
|---|---|---|---|
| 1 | Traffic_seq1 | avi |  |
| 2 | Jeep_seq | frames |  |
| 3 | Viptraffic | avi |  |
| 4 | Traffic_seq2 | mpeg |  |
| 5 | Traffic_seq3 | mpeg |  |
| 6 | Pedestrian | avi |  |

The software outputs the ground truth of each object with height, width, Centroid and bounding box. Single Visual tracking system is compared with traditional mean shift method. In the proposed method, the block matching method using 3D color histogram has been used. Single object tracking is compared with ground truth variations using Euclidean distance measures. Figure 4.26 shows the ground truth variations for different sequences. Bike sequence using proposed method gives near results with mean shift method. The proposed algorithm gives better result than the mean shift algorithm but at the cost of execution time. The execution time for proposed method is more than the mean shift method.

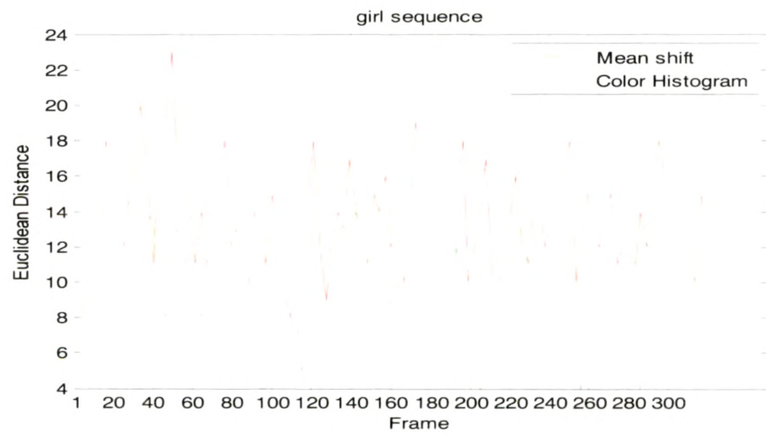
For multiple objects tracking, a procedure has been proposed based on the following principles:

- A set of test sequences are selected. All moving objects are then detected and manually corrected to obtain the ground truth, one frame per second.
- The output of the tracking algorithm is compared with the ground truth.
- The test images are used to evaluate the performance of the object detection algorithms.

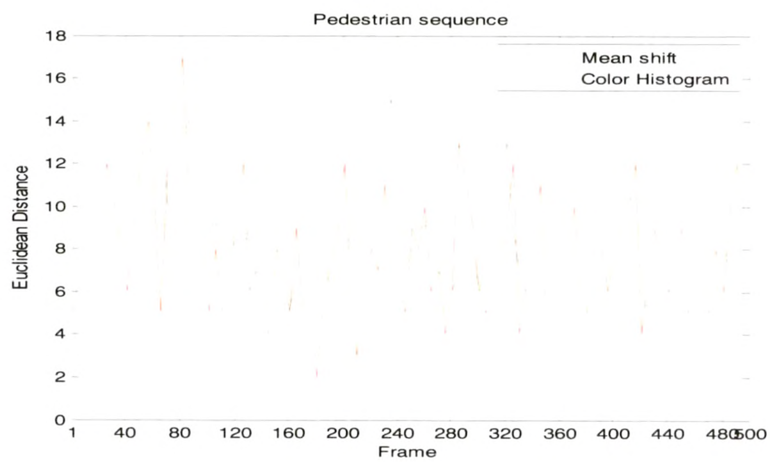
In order to compare the output of the algorithm with the ground truth segmentation, a region matching procedure is adopted which allows to establish a correspondence between the detected objects and the ground truth. Several cases are considered as follows:

1. Perfect Match: the detected region matches with one and only one region.
2. Detection Failure: the test region has no correspondence.
3. False Alarm: the detected region has no correspondence.
4. Merge Region (M): the detected region is associated to several test regions.
5. Split Region (S): the test region is associated to several detected regions.
6. Split-Merge Region (SM): when the conditions 4, 5 are simultaneously satisfied.

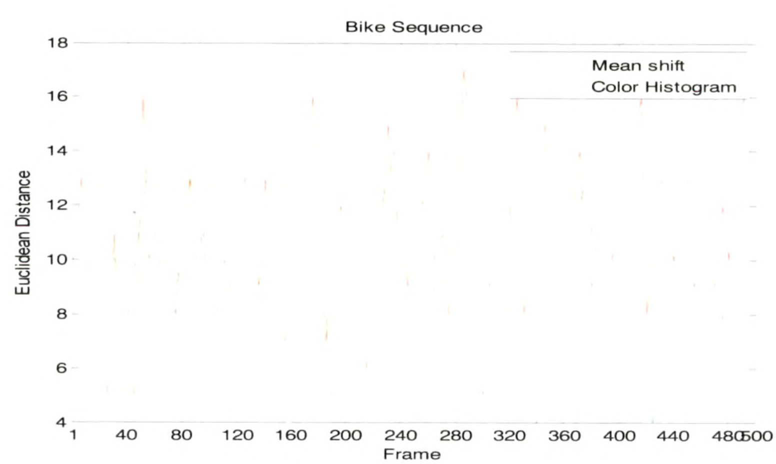
The Figure 4.27 shows the different class target between the actual ground truth and tracked target. The performance matrix for evaluation can be generated using the above conditions. The matrix generated for the Figure 4.27 is shown in the Table 4.8.




(a)

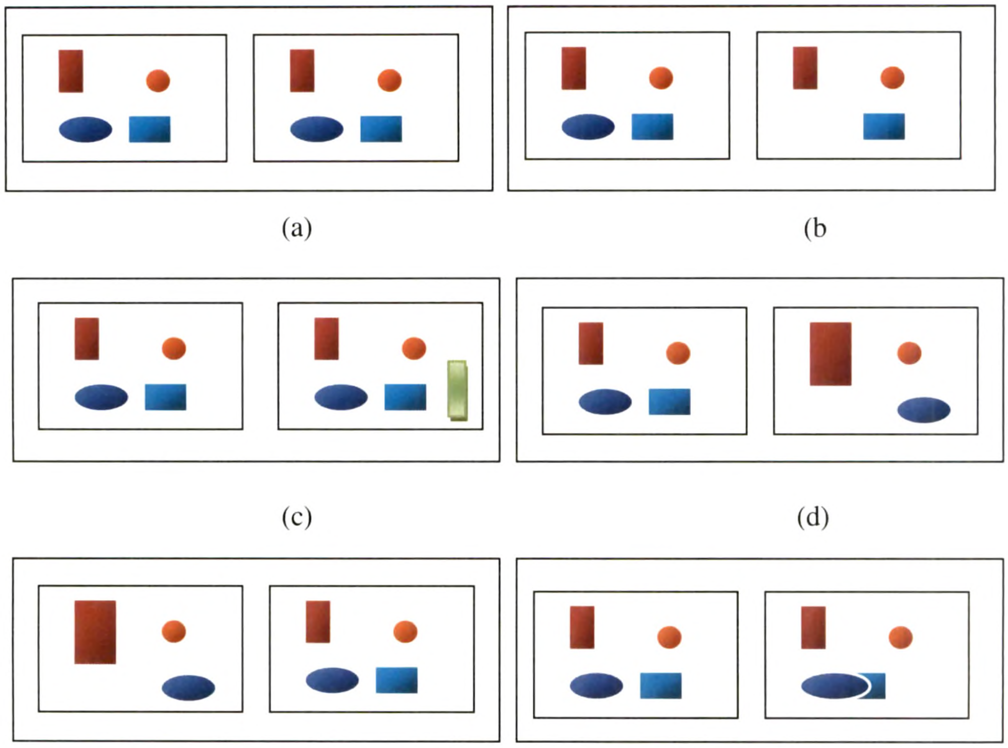



(b)




(c)

Figure 4.26 : Ground Truth Variations (a) Girl_walking Sequence (b) Pedestrian Sequence (c) Bike Sequence

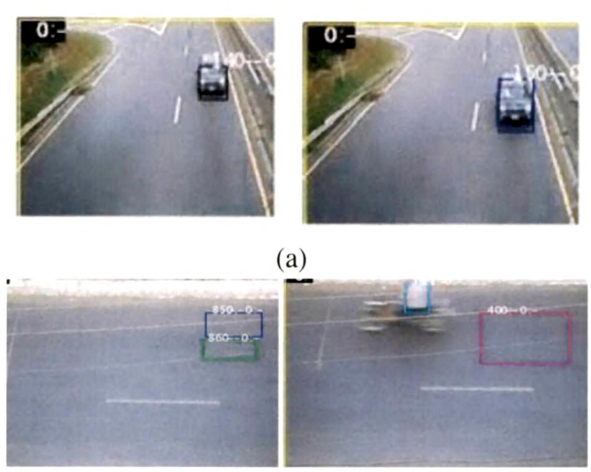


(a) (b

(c) (d)

(e) (f)

Figure 4.27 : Region Matching Cases: (a) Perfect Match (b) Detection Failure (c) False Match (d) Merge (One Correspondence for More than One Target) (e) Split (More than One Correspondence for One Target) (f) Split and Merge (Conditions (d) and (e) Together)



(a)

(b)

Figure 4.28 : Region Matching Failure: (a) Detection Failure of the Same Object in the Frame Number 72 and Frame Number 73 (Tracked as a new object) (b) False Match in Two Different Frames

Some of the different cases mentioned in the Table 4.8 have been shown in the Figure 4.28. Figure 4.28 (a) shows the failure of the same object. Due to the failure of the blob extraction to extract the proper blob, in the next frame the same vehicle is considered as a new vehicle and tracked considering new vehicle. Table 4.9 reports the performance results for different cases handled by the proposed algorithm in some of the traffic sequences.

Table 4.8 : Performance Matrix generated for Different Region Matching Cases

| Figure 4.26 | Correspondence Matrix |
|---|---|
| Perfect Match | $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| Detection Failure | $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ |
| False Match | $M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ |
| Merge | $M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$ |
| Split | $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ |
| Split and Merge | $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ |

Table 4.9 : Comparative Performance of Sequence for Different Region Matching Cases

| Match Case | viptraffic | Traffic_seq1 | Traffic_seq2 |
|---|---|---|---|
| Correct Detection | 97.5 | 93.5 | 90.5 |
| Detection Failures | 2.5 | 2.1 | 4.6 |
| Splits | 0 | 1.3 | 1.8 |
| Merges | 0 | 0 | 2.4 |
| Split/Merges | 0 | 0 | 1.2 |
| False Match | 0.8 | 2.5 | 3.6 |

## 4.5 Motion Estimation using Camera Modeling

To find the actual velocity of the vehicle, the field of view (FOV) of the camera must be set up so that it acquires the moving direction of the vehicles. Camera set up should be such that it takes side view images of the vehicles. This kind of acquisition plan provides advantages on the solution of the scale problem which leads object identification and region tracking task efficiently. On the other side it causes the analysis time of the vehicle to be shortened. In other words, entrance and exit time of a vehicle into the FOV of the camera is shortened. For performing the real time procedures for speed estimation, this situation requires less time for calculations. The mounting of camera on the front side needs highly accurate information about the depth of the road for measuring the speed of the object space. Thus selection of the camera mounting on side view or front view of the camera depends upon the view location of the objects to be tracked.

- **Calculations for image space to object space conversion**

The direction of the object has been calculated by finding the angle using the equation (4.8)

$$\text{Direction} = \tan^{-1} y/x \quad \textbf{(4.8)}$$

Where y and x are the y coordinate and x coordinate of the Centroid pixel respectively.

Actual velocity of the vehicle or moving object is calculated by projecting the object from the image space to actual object space using the camera parameters calculated using equation (4.1) to equation (4.7).

Camera parameters are calculated to find magnification ratio considering camera mounting on height and tilted at some angle. The camera parameter calculation software has been developed to find the actual magnification ratio. Considering the targeted application for video surveillance system, the camera parameter calculation software designed to increase efficiency of security system while lowering costs for finding the best camera locations.

To find the magnification ratio from optimal positions CCD /CCTV cameras, a field of view, viewing angles and lens focal length are calculated using trigonometry functions as shown in the Figure 4.29.

Parameters are needed to calculate the magnification ratio is:

- **Distance from Camera** – Maximum distance from Camera to the target.
- **Camera Installation Height** – CCTV camera installation height.
- **Field of View: Height** – Height of the target. When user select the Field of View (FOV) Height for the camera installation, the software calculates the camera Tilt.
- **Field of View: Width** – The other option is to specify FOV width instead of the height. Just enter the desired width of field of view (viewing area) for the specified camera distance. If you modify **FOV** parameters the **Focal Length** and the **Viewing Angles** will be automatically recalculated. The other option

is to specify viewing angles instead of FOV Width. In this case FOV and **Camera Focal Length** will be calculated automatically.

- **Camera Sensor Format** – CCD or CMOS sensor size (sensor format). User can select the sensor format from: 1/4″, 1/3.6″, 1/3″ , 1/2.5″, 1/2″, 2/3″, 1″ and 1.25″. Usually user can find the sensor format in the camera specification.

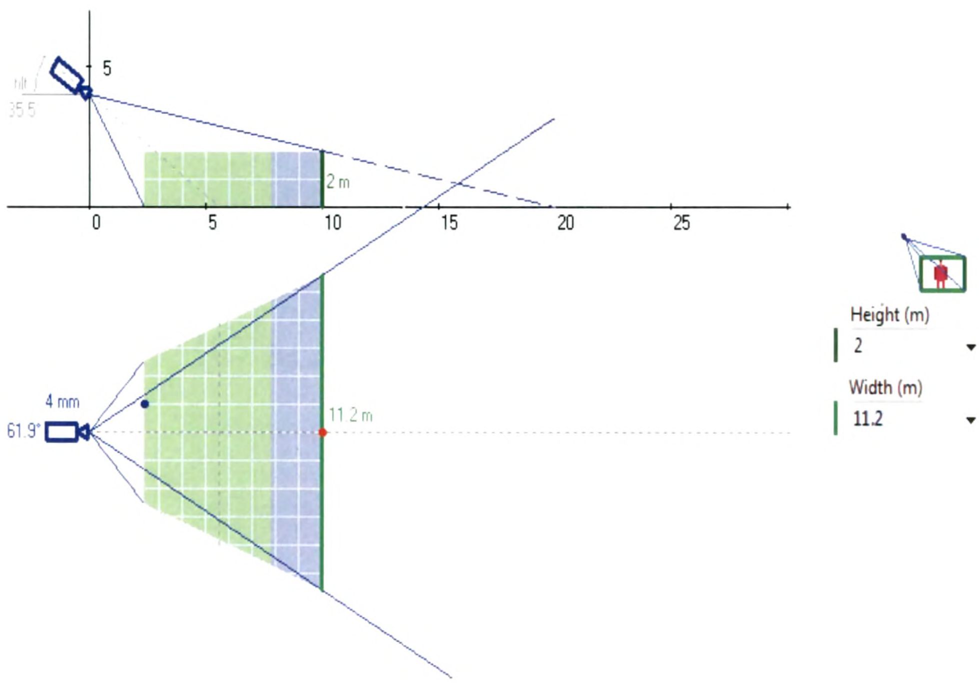



Figure 4.29 : Camera Parameters Calculations using Trigonometry Functions

Magnification ratio can be calculated using the ratio of the distance of the object to lens focal length. Table 4.10 reports the different input parameters considering the height of mounting camera, distance of the object and minimum height of the object required for tracking. Camera motion parameter software calculates the focal length, width of the object visible according to the height consider as input parameter of the object. It also calculates span of the Horizontal Angle of View (H.A.V) and Vertical Angle of View (V.A.V). The camera parameter software calculates the tilting angle of camera required to track the object with reference to the given input parameters. Table 4.10 reports the camera parameters calculations using different input parameters for different sensor size, different values of camera mounting height and distance of the object from camera.

Table 4.10 : Camera Parameters Calculations for Different Image Sensor Size using the Proposed Software

(a) Image Sensor Size = $\frac{1}{2"}$

| **Input Parameters** | | | **Calculated Parameters** | | | | |
|---|---|---|---|---|---|---|---|
| **Image Sensor Size = $\frac{1}{2"}$ (h = 4.8mm ,w = 6.4 mm)** | | | | | | | |
| **Camera Mounting Height (m)** | **Distance of Object (m)** | **Object Height (m)** | **Focal Length (mm)** | **Object Width (m)** | **Angle of Tilt (degree)** | **H.A.V (degree)** | **V.A.V (degree)** |
| 10 | 20 | 2 | 10 | 12.8 | 38.4 | 35.5 | 27.0 |
| 10 | 20 | 2 | 8 | 16.0 | 43.6 | 43.6 | 33.4 |
| 10 | 20 | 2 | 6 | 21.3 | 56.1 | 56.1 | 43.6 |
| 10 | 20 | 2 | 4 | 32.0 | 77.0 | 77.2 | 61.9 |
| 10 | 10 | 2 | 10 | 6.4 | 59.2 | 35.5 | 27.0 |
| 10 | 10 | 2 | 8 | 8.0 | 62.5 | 43.6 | 33.4 |
| 10 | 10 | 2 | 6 | 10.6 | 67.5 | 56.1 | 43.6 |
| 10 | 10 | 2 | 4 | 16.0 | 76.7 | 77.2 | 61.9 |
| 8 | 10 | 2 | 10 | 6.4 | 47.8 | 35.5 | 27.0 |
| 8 | 10 | 2 | 8 | 8.0 | 51.0 | 43.6 | 33.4 |
| 8 | 10 | 2 | 6 | 10.6 | 56.1 | 56.1 | 43.6 |
| 8 | 10 | 2 | 4 | 16.0 | 61.9 | 77.2 | 61.9 |
| 6 | 10 | 2 | 10 | 6.4 | 36.4 | 35.5 | 27.0 |
| 6 | 10 | 2 | 8 | 8.0 | 39.6 | 43.6 | 33.4 |
| 6 | 10 | 2 | 6 | 10.6 | 44.7 | 56.1 | 43.6 |
| 6 | 10 | 2 | 4 | 16.0 | 53.9 | 77.2 | 61.9 |
| 4 | 10 | 2 | 10 | 6.4 | 25.0 | 35.5 | 27 |
| 4 | 10 | 2 | 8 | 8.0 | 28.2 | 43.6 | 33.4 |
| 4 | 10 | 2 | 6 | 10.6 | 33.3 | 56.1 | 43.6 |
| 4 | 10 | 2 | 4 | 16.0 | 42.4 | 77.2 | 61.9 |

(b) Image Sensor Size = $\frac{1}{2.5''}$

| Input Parameters | | | Calculated Parameters | | | | |
|---|---|---|---|---|---|---|---|
| Image Sensor Size = $\frac{1}{2.5''}$ (h = 4.2 mm ,w = 5.6 mm) | | | | | | | |
| **Camera Mounting Height (m)** | **Distance of Object (m)** | **Object Height (m)** | **Focal Length (mm)** | **Object Width (m)** | **Angle of Tilt (degree)** | **H.A.V (degree)** | **V.A.V (degree)** |
| 10 | 20 | 2 | 10 | 11.2 | 34.7 | 31.2 | 23.7 |
| 10 | 20 | 2 | 8 | 14.0 | 37.6 | 38.5 | 29.4 |
| 10 | 20 | 2 | 6 | 18.6 | 42.2 | 50.0 | 38.5 |
| 10 | 20 | 2 | 4 | 28.0 | 50.6 | 69.9 | 55.3 |
| 10 | 10 | 2 | 10 | 5.6 | 57.6 | 31.2 | 23.7 |
| 10 | 10 | 2 | 8 | 7.0 | 60.5 | 38.5 | 29.4 |
| 10 | 10 | 2 | 6 | 9.3 | 65.0 | 50.0 | 38.5 |
| 10 | 10 | 2 | 4 | 14.0 | 73.5 | 69.9 | 55.3 |
| 8 | 10 | 2 | 10 | 5.6 | 46.23 | 31.2 | 23.7 |
| 8 | 10 | 2 | 8 | 7.0 | 49.08 | 38.5 | 29.4 |
| 8 | 10 | 2 | 6 | 9.3 | 53.6 | 50.0 | 38.5 |
| 8 | 10 | 2 | 4 | 14.0 | 62.0 | 69.9 | 55.3 |
| 6 | 10 | 2 | 10 | 5.6 | 34.8 | 31.2 | 23.7 |
| 6 | 10 | 2 | 8 | 7.0 | 37.6 | 38.5 | 29.4 |
| 6 | 10 | 2 | 6 | 9.3 | 42.2 | 50.0 | 38.5 |
| 6 | 10 | 2 | 4 | 14.0 | 50.6 | 69.9 | 55.3 |
| 4 | 10 | 2 | 10 | 5.6 | 23.3 | 31.2 | 23.7 |
| 4 | 10 | 2 | 8 | 7.0 | 26.2 | 38.5 | 29.4 |
| 4 | 10 | 2 | 6 | 9.3 | 30.8 | 50.0 | 38.5 |
| 4 | 10 | 2 | 4 | 14.0 | 39.2 | 69.9 | 55.3 |

(C) Image Sensor Size = $\frac{1}{3"}$

| Input Parameters | | | Calculated Parameters | | | | |
|---|---|---|---|---|---|---|---|
| Image Sensor Size = $\frac{1}{3"}$(h = 3.6mm ,w = 4.8mm) | | | | | | | |
| Camera Mounting Height (m) | Distance of Object (m) | Object Height (m) | Focal Length (mm) | Object Width (m) | Angle of Tilt (degree) | H.A.V (degree) | V.A.V (degree) |
| 10 | 20 | 2 | 10 | 9.6 | 33.1 | 26.9 | 20.4 |
| 10 | 20 | 2 | 8 | 12.0 | 35.6 | 33.4 | 25.3 |
| 10 | 20 | 2 | 6 | 16.0 | 39.6 | 43.6 | 33.4 |
| 10 | 20 | 2 | 4 | 24.0 | 47.1 | 61.9 | 48.4 |
| 10 | 10 | 2 | 10 | 4.8 | 56.0 | 26.9 | 20.4 |
| 10 | 10 | 2 | 8 | 6.0 | 58.4 | 33.4 | 25.3 |
| 10 | 10 | 2 | 6 | 8.0 | 62.5 | 43.6 | 33.4 |
| 10 | 10 | 2 | 4 | 12.0 | 69.9 | 61.9 | 48.4 |
| 8 | 10 | 2 | 10 | 4.8 | 44.5 | 26.9 | 20.4 |
| 8 | 10 | 2 | 8 | 6.0 | 47.0 | 33.4 | 25.3 |
| 8 | 10 | 2 | 6 | 8.0 | 51.0 | 43.6 | 33.4 |
| 8 | 10 | 2 | 4 | 12.0 | 58.6 | 61.9 | 48.4 |
| 6 | 10 | 2 | 10 | 4.8 | 33.1 | 26.9 | 20.4 |
| 6 | 10 | 2 | 8 | 6.0 | 35.6 | 33.4 | 25.3 |
| 6 | 10 | 2 | 6 | 8.0 | 39.0 | 43.6 | 33.4 |
| 6 | 10 | 2 | 4 | 12.0 | 47.1 | 61.9 | 48.4 |
| 4 | 10 | 2 | 10 | 4.8 | 21.7 | 26.9 | 20.4 |
| 4 | 10 | 2 | 8 | 6.0 | 24.1 | 33.4 | 25.3 |
| 4 | 10 | 2 | 6 | 8.0 | 28.2 | 43.6 | 33.4 |
| 4 | 10 | 2 | 4 | 12.0 | 35.7 | 61.9 | 48.4 |

For experiment purpose some of the real time road sequences taken from the ordinary Sony DSC s650 camera have been used. A camera with a frame rate of 30 fps with 320 × 240 pixels has been used. The focal length of the camera can be adjusted from 5.8 to 17.4 mm using 3 × zoom. Experiment has been performed with zoom and without zoom. The scale or magnification factor of the images is related to the camera-to object distance and the focal length of the camera. Scale of a rectified image can be obtained approximately by the relation

$$s = \frac{d}{f} = \frac{h}{H} = \frac{w}{W} \tag{4.9}$$

Where d is the camera to object distance
f is the focal length of the camera
h is the sensor height
H is the actual Height of the object
w is the sensor width
W is the actual width of the object

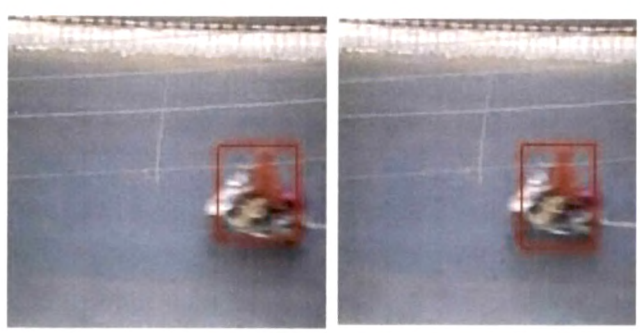


Figure 4.30 : Speed Measurement of Vehicle from the "Traffic 1" Sequence
(a) Frame Number 655 (b) Frame Number 656

Actual velocity $V$ can be calculated using

$$V = m * s * 3.6 \tag{4.10}$$

Where, m is the distance in the image space that can be calculated using the blob statistics derived in the proposed algorithm. Magnification factor or scale factor can be calculated using equation (4.9). Product of m and s calculates the velocity in the meters per millisecond that is converted in to the kilometer per hour.

Table 4.11 : Camera to Object Distance and Minimum Speed Measured with Camera

| Focal Length (mm) | Distance (m) | Minimum Speed that can be measured in km/hr with 5.8 mm / 3 × zoom |
|---|---|---|
| 5.8mm / 3 × zoom = 17.4 mm | 10 | 6.2 / 2.06 |
| | 20 | 12.4 / 4.13 |
| | 30 | 18.6 / 6.2 |

Actual speed measure can be calculated using image space distance. As shown in the Figure 4.30, the image space distance of vehicle is 6 pixels per frame calculated using the proposed algorithm. Speed in object space can be calculated using the equation (4.9) and (4.10).

Table 4.12 : Accuracy Measurement Test

| Experiment | Calculated Speed using Proposed Method (A) | Actual Speed measured using Speedometer (B) | Error $\|A - B\|$ |
|---|---|---|---|
| 1 | 34.14 | 35 | 0.86 |
| 2 | 37.6 | 38 | 0.4 |
| 3 | 44.75 | 45 | 0.25 |
| 4 | 57.8 | 58 | 0.2 |
| 5 | 74.88 | 75 | 0.12 |

Table 4.11 reports the minimum speed calculated using equation (4.10) at different distances from camera to object. To measure the performance of the algorithm, different vehicles are used to measure the speed. Vehicle speed is measured with the speedometer of the vehicle and compared with the calculated speed using the proposed method. Table 4.12 reports the actual speed and speed calculated using the proposed method. The relative errors of estimation using the proposed method are obtained by computing the differences between actual speed and calculated speed.

Summary: Experimental results of the proposed method for visual tracking are compared with the standard methods. For Single visual tracking, the novel block matching algorithm has been proposed. For Multiple objects tracking, the hybrid tracker with color and feature transform using Contourlet transform from blob statistics has been used. Object classifier is implemented and embedded with the proposed hybrid tracker for object identification. The performances of the results have been tested using number of image sequences. The motion parameters of the objects have been calculated using camera model parameters and implemented for best location of camera for object tracking.