

**OBJECT IDENTIFICATION AND
ESTIMATION OF OBJECT MOTION
PARAMETERS FROM IMAGE SEQUENCES
FOR MACHINE INTELLIGENCE
APPLICATION**

A thesis submitted
for the award of the degree of

Doctor of Philosophy

in

Electrical Engineering

By

Nehal Gitesh Chitaliya



**DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA – 390 001
GUJARAT, INDIA
OCTOBER, 2012**

**OBJECT IDENTIFICATION AND
ESTIMATION OF OBJECT MOTION
PARAMETERS FROM IMAGE SEQUENCES
FOR MACHINE INTELLIGENCE
APPLICATION**

A thesis submitted
for the award of the degree of

Doctor of Philosophy

in

Electrical Engineering

By

Nehal Gitesh Chitaliya



**DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA – 390 001
GUJARAT, INDIA
OCTOBER, 2012**

Declaration

I, Nehal Gitesh Chitaliya hereby declare that the work reported in this thesis entitled **OBJECT IDENTIFICATION AND ESTIMATION OF OBJECT MOTION PARAMETERS FROM IMAGE SEQUENCES FOR MACHINE INTELLIGENCE APPLICATION** submitted for the award of the degree of **DOCTOR OF PHILOSOPHY** in Electrical Engineering Department, Faculty of Technology & Engineering, The Maharaja Sayajirao University of Baroda, Vadodara is an original work and has been carried out in the Department of Electrical Engineering, Faculty of Technology & Engineering, The Maharaja Sayajirao University of Baroda, Vadodara. I further declare that this thesis is not substantially the same as one, which might have been submitted in part or in full for the award of any degree or academic qualification in this University or in any other Institution or examining body in India or abroad.

October, 2012

Nehal G. Chitaliya

Dedicated to
My Parents, My Teachers,
My Beloved Husband
And
My Loving Daughter

Acknowledgement

The satisfaction that comes with successful completion of the thesis work would be incomplete without remembering people who have made it possible. It gives me immense pleasure to acknowledge all those who have extended their valuable guidance and magnanimous help.

I could achieve this success only because of the grace and blessing of God. Foremost, I would like to express my sincere gratitude to Prof. A. I. Trivedi for his continuous support during my Ph.D study and research. I am deeply indebted to him, for providing valuable guidance, motivation and encouragement throughout the study. Words are inadequate to appreciate the knowledge, insight and patience with which he guided this work.

Prof. S. K. Shah, the Head of the Department of Electrical Engineering at The M. S. University of Baroda has been my teacher for many years. He offered advice and suggestions whenever I needed them. I am very much thankful for his kind support. I also present my heartiest gratitude towards Prof. S. K. Joshi, Prof. M. S. Gosawi and Prof. S. K. Gosawi for their blessings and valuable suggestions during my research work.

I am thankful to Dr. H. B. Dave from bottom of my heart, who has given me inspiration and encouragement to work in this area. He helped me in identifying the problem and led me in this direction.

I am deeply indebted to entire SVIT Family, Chairman Shri Bhaskarbhai Patel, Principal, Management and all my colleagues who have supported me in my research work. I specially thank to Tejasbhai for helping me by providing number of e-journals and e-books. I would also like to especially thank Mr. J. N. Patel for adjusting my teaching load whenever I needed. I would like to thank all my departmental colleagues for their kind support. I am extremely grateful to my best friend Ms. Jagruti Patel for her encouragement and support throughout.

My deepest gratitude goes to my family for their unflagging love and support throughout my life, this dissertation is simply impossible without them. My research work would have not been completed without the blessing of my father-in-law Late shree Harjivandas Chitaliya and mother-in-law Champaben Chitaliya. I am very much thankful for my father Harshadray shah and my mother Mrs. Nayanaben Shah for their love and care. I have no suitable word for my mother that can fully describe her sacrifice and everlasting love to me. I thank very much to my sister Sheetal Shah, my brother Jignesh Shah and my sister-in-law Asmita Shah for their moral and kind support to me whenever I required.

I express my heartiest gratitude to my beloved husband Gitesh and my daughter Mrunalee, for their unconditional support in every aspect of my life. Gitesh has been a constant source of encouragement during my work. My daughter Mrunalee helped me by doing her study herself and helping me a lot whenever I required. Without their patience and inspiration it would have not been possible for me to start and continue my research.

At last but not least, I thank one and all who have directly or indirectly helped me in my research work.

Nehal G. Chitaliya

Abstract

Machine vision is the process of receiving and analyzing visual information by digital computer. The research on machine vision focuses on methods and systems for analyzing images. Visual analysis of the objects attempt to detect, identify and track the moving objects like people or vehicles. It also interprets the object behavior from image sequences involving the objects. Object motion analysis has attracted a great interest due to its promising application in the real world such as Visual Surveillance, Traffic Monitoring System, Perceptual User Interface, Animation Film, Video Games, Content based Image Storage and Retrieval, Video Conferencing, Athletic Performance Analysis, Virtual Reality, Biometric Security, Animal Behavioral Science, Human Assisted Motion Annotation, Video Compression, Medical Robots, Military Robots, Pick and Place Industrial Automation, Robotics etc.

Recognizing the objects and perceiving actions of the objects are prerequisites for Machine Intelligent System. The goal of this research work is to develop a generalized model that can identify the moving objects and also able to estimate the motion parameters such as location, direction and speed of moving objects from the image sequences generally captured with the help of CCD Camera for machine intelligence application. The Machine Intelligence System involves mainly two tasks that are Object Recognition and Visual Tracking.

Feature extraction is a key element for designing a Classifier used for Object Recognition. Feature extraction has been carried out in the frequency domain. For more efficient feature extraction, Unsharp Filter and binary threshold is used before applying feature extraction. Unsharp filter amplifies the high frequency components

which enhances the edges of an image. Feature extraction coefficients are extracted by applying Discrete Contourlet Transform that overcomes the problem of representing an image with smooth contours in different directions by providing two additional properties that are directionality and anisotropy as compared to the Discrete Wavelet Transform (DWT). Principal Component Analysis (PCA) has been carried out for dimensionality reduction to create the feature matrix. For feature matching, Euclidean distance classifier and back-propagation neural network have been used. The results of discrete Contourlet transform are compared with Discrete Curvelet Transform. The Discrete Contourlet transform is more efficient and robust method than the Discrete Curvelet Transform. Efficiency of the Classifier has been tested using various types of datasets like face dataset and vehicle dataset.

Visual tracking task has been implemented for single visual tracking and multiple object tracking. Two different approaches are used for both tasks. For single object tracking, a novel Block Matching Algorithm using Predictive Motion Vector based on 3D color histogram has been proposed and implemented efficiently. System tracks the single object selected by the user. Different conditions of the object like similar type of background and foreground, object moving near to frame boundary, object with no motion in the frame sequence etc. are efficiently implemented.

For efficient multiple object tracking, hybrid tracker is used. Hybrid tracker is a combination of color statistics and features of objects. Blob tracking algorithm is used for efficient tracking. The objects are tracked by temporal relationships between blobs without using domain-specific information. For further improvement in the conventional blob tracking, color segmentation is applied to retrieve color statistics of the object. To eliminate the effect of the shadow and lighting effect, all color space is converted to YC_bC_r color space which is widely used for video processing. Pre-processing is applied for better blob extraction.

A distinctive feature of the proposed algorithm is that the method operates on region descriptors instead of region themselves. This means that instead of projecting the entire region into the next frame, only region descriptors need to be processed. Therefore, there is no need for computationally expensive models. Object statistics

has been calculated using Blob Analysis Algorithm. Region tracking and matching is implemented using Color Histogram and 2D Moment Invariants. Contourlet transform features are used for matching, to overcome the tracking problem of same color objects (same histogram). Centroid Statistics is used to measure the distance and direction of the object with respect to the previous frame. Different statistical conditions are incorporated for making efficient algorithm. Vehicle classifier is also incorporated which displays the class of vehicle indicating car; bus etc in the visualization of tracking.

For calculating motion estimation parameters like actual speed of the object, the camera modeling parameters are calculated and converted from image space to the actual object space. For low cost development of the software model, simple digital camera is used. For Visual surveillance application, software model has been developed for calculating the actual focal length, distance between the object and camera and Magnification ratio from parameters of the camera.

Contents

Acknowledgement	i
Abstract	iii
List of Figures	viii
List of Tables	xi
List of Abbreviations	xii
1 Introduction	1
1.1 Machine Intelligence System	2
1.1.1 Object Identification	5
1.1.2 Visual Tracking	12
1.2 Overview of the Proposed Work	15
1.2.1 Contribution	16
1.2.2 GUI for the Proposed Method	19
1.3 Layout of the Thesis	19
2 Background and Related work	21
2.1 Designing of Classifier	22
2.1.1 Feature Extraction	22
2.1.2 Distance Measures	28
2.1.3 Performance Matrices	31
2.2 Visual Tracking	33
2.2.1 Background model	33

2.2.2	Statistical Approach	35
2.2.3	Object Tracking Techniques	35
3	Proposed Approach	40
3.1	Object Classifier.....	41
3.1.1	Pre-processing.....	45
3.1.2	Feature Extraction.....	49
3.1.3	Feature Selection.....	57
3.1.4	Feature Matching	60
3.1.5	Proposed Methodology of Object Classifier.....	62
3.2	Vehicle Identification System.....	65
3.3	Visual Tracking.....	65
3.3.1	Single Object Tracking Algorithm.....	65
3.3.2	Multiple Objects Tracking	75
4	Experimental Results	88
4.1	Datasets.....	88
4.1.1	Face Dataset and Vehicle Dataset.....	88
4.1.2	Test Sequences.....	94
4.2	Camera Modeling Parameters.....	95
4.3	Classification and Tracking Results.....	104
4.3.1	Object Classifier.....	104
4.3.2	Visual tracking System	110
4.4	Performance Analysis of the Proposed Algorithm	119
4.5	Motion Estimation using Camera Modeling.....	125
5	Conclusions and Future Scope	134
5.1	Conclusions.....	135
5.2	Limitations and Future Scope	136
	Publications	138
	Bibliography	140

List of Figures

1.1 :	Machine Intelligence System	3
1.2 :	Modules of Object Recognition System	6
1.3 :	Modules of Visual Tracking System.....	13
3.1 :	Eigen Matrix Generation for Training Dataset	42
3.2 :	Pre-processing and Feature Extraction	43
3.3 :	Object Identification of Query Image	44
3.4 :	Spatial Sharpening	45
3.5 :	Pre-processing (a) Image of Car (b) Gray-scale Image (c) Image after applying Unsharp filter (d) Image after applying Threshold	48
3.6 :	(a) Bicycle Image (b) Pre-processed Image	49
3.7 :	Double Filter Bank Decomposition of Discrete Contourlet Transform.....	51
3.8 :	Decomposition of Image using Contourlet Transform (2-Level and ‘pkva’ Filter for Pyramid and Directional Filter).....	52
3.9 :	Curvelet in the Fourier Frequency domain [28].	55
3.10:	Wrapping Wedge Around the Origin by Periodic Tilting of Wedge Data. The Angle θ is in the Range $(\pi/4, 3\pi/4)$	56
3.11:	Decomposition of Image using Curvelet Transform (a) First Level (b) Second Level	56
3.12:	(a) Eigenspace Image after applying Curvelet Transform without Pre- processing (b) Eigenspace Image after applying Pre-processing and Curvelet Transform	59
3.13:	Feed Forward Neural Network Model	61
3.14:	Learning Phase of the Neural Network Classifier	61
3.15:	Block Diagram of Proposed Object Classifier System	62

3.16:	Training of Vehicle Dataset using Three Class Structures	66
3.17:	Vehicle Identification System.....	67
3.18:	Single Object Tracking Algorithm.....	69
3.19:	Large Predictive Search Pattern – Nine points	70
3.22:	Block Matching and Search Predicted Vector Direction for Predictor Vector Found in the Direction 6	73
3.23:	Visual Tracking Algorithm for Multiple Objects	77
3.24:	Visual Object Tracker for performing Region Matching and Tracking	78
3.25:	(a) 4-Neighbourhood Pixels (b) 8-Neighbourhood Pixels	81
3.26:	Blob Segmentation Module	82
3.27:	Visual Tracking System.....	86
4.1 :	(a) Face Images with Different Position and Tilting (b) Gray Scale Images of IIT Kanpur Dataset	90
4.2 :	(a) Sample Images from Face 94 Dataset having Different Pose (b) Some of the Images of Face94 Dataset used for Training	91
4.3 :	Some of the Gray Scale Images of Face94 Dataset used for Testing	92
4.4 :	Vehicle Dataset from PASCAL VOC 2006.....	93
4.5 :	Some of the Sequences from PETS 2000 Dataset used for Visual Tracking ..	94
4.6 :	Basic Geometry Model of the Object in 3- D Space	96
4.7 :	Focal Length in Camera Model	97
4.8 :	Angle of View in Image Sensor.....	98
4.9 :	Angle of View in CCTV Camera.....	99
4.10:	Field of View	100
4.11:	Image Sensor Size	102
4.12:	View of Image Acquisition Plane	103
4.13:	Images after applying Pre-processing Stage	105
4.14:	(a) Eigenfaces using Contourlet-PCA after Pre-processing Stage (b) Eigenfaces using Curvelet-PCA after Pre-processing Stage.....	106
4.15:	Vehicle Images from the VOC 2006 Dataset	109
4.16:	Enhanced Images after performing Pre-processing on VOC 2006 Dataset...	109
4.17:	‘Girl_walking’ sequence & Tracking result in different frame	111
4.18:	(a) Tracking Results of Person 1 in ‘Rain’ sequence (b) Tracking Results of Person 2 with Boundary Termination Conditions	112

4.19:	Tracking Results of Pedestrian (a) Using Mean Shift Method (b) Using Proposed Method	113
4.20:	Tracking of Bike sequence (a) Using Mean Shift Method (b) Using Proposed Method	114
4.21:	Tracking Failure (a) Helicopter Sequence - Frame 301,401,501,601.....	114
4.22:	Vehicle Tracking in the viptraffic Sequence (a) Tracking Vehicles (b) Blob Extraction (c) Region Tracking with Motion Parameters	117
4.23:	Visual Movie Frame for Proposed Algorithm	118
4.24:	Visualized Results in the format [Object Number: - Speed: - Direction] (a) Frame number 72 (b) Frame number 113.....	118
4.25:	Object Classifier (a) Correct Identification (b) False Identification.....	119
4.26:	Ground Truth Variations (a) Girl_walking Sequence (b) Pedestrian Sequence (c) Bike Sequence.....	122
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).....	123
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	123
4.29:	Camera Parameters Calculations using Trigonometry Functions.....	127
4.30:	Speed Measurement of Vehicle From the “Traffic 1” Sequence.....	131

List of Tables

4.1 :	Camera Format.....	98
4.2 :	Recognition Rate for Object Classifier System	107
4.3 :	Execution Time required for Training and Testing of Face mages.	108
4.4 :	Performance Evaluation for VOC 2006 Dataset.....	110
4.5 :	Performance Evaluation of Image Sequences.....	113
4.6 :	Sequences used for Single Object Tracking	115
4.7 :	Some of the Sequences used for Multiple Objects Tracking	120
4.8 :	Performance Matrix Generated for Different Region Matching Cases	124
4.9 :	Comparative Performance of Sequence for Different Region Matching Cases	125
4.10:	Camera Parameters Calculations for Different Image Sensor Size using the Proposed Software	128
4.11:	Camera to Object Distance and Minimum Speed measured with Camera....	132
4.12:	Accuracy Measurement Test.....	132

List of Abbreviations

CCD	:	Charge-Coupled Device
AI	:	Artificial Intelligence
DCT	:	Discrete Cosine Transform
DFT	:	Discrete Fourier Transform
FFT	:	Fast Fourier Transform
DWT	:	Discrete Wavelet Transform
KNN	:	K-Nearest Neighbour
MLP	:	Multilayer Perceptron
RBF	:	Radial Basis Function
SVM	:	Support Vector Machine
SOM	:	Self Organizing Map
PCA	:	Principal Component Analysis
HCD	:	Harris Corner Detector
SIFT	:	Scale Invariable Feature Transform
SURF	:	Speed up Robust feature Transform
RANSAC	:	Random Sample Consensus
LDA	:	Linear Discriminant Analysis
FDCT	:	Fast Discrete Curvelet Transform
USFFT	:	Unequally-Spaced Fast Fourier Transform
SSD	:	Sum-of-Squared Differences
MS	:	Mean Shift
ME	:	Motion Estimation
BMA	:	Block Matching Algorithm
LP	:	Laplacian Pyramid

Chapter 1

1 Introduction

The modern era of automation demands the involvement of machine to perform critical tasks efficiently and accurately for convenience and enhancement of the quality of life. These demands focus on the research to develop an intelligent system having the ability to perceive, calculate and learn from the experiences. One of the key motivations behind the use of intelligent technologies is the fact that they can deal with human cognitive limitations, i.e., human failure to monitor all information, to resolve complex and conflicting situations, to identify high-revenue opportunities or to prevent high-cost mistakes.

Intelligent systems aim to realize the real world applications targeted to Automated Visual Surveillance, Traffic Monitoring System, Intelligent Vehicle System, Robot Navigation and Animations etc [1].

- Automated Visual Surveillance System: The smart surveillance system works over security-sensitive areas such as banks, departmental stores, parking lots, and borders of the countries. In present systems, surveillance camera outputs are recorded into the video archives. These video data are currently used as forensic tools after the incidence. For real time analysis, security officers need to be alert in the control room for continuously watching the progress of the events. There is a need of alert

only when the motion is detected. So it is necessary to build generalized model which can work effectively for real time applications under the normal conditions [3].

- **Traffic Monitoring System:** Traffic monitoring system identifies the types of the vehicles and monitors the flow of vehicles. Traffic monitoring system needs to estimate the speed and direction of the moving vehicles and pedestrians and also the traffic intensity for the prevention of the road accidents. Traffic monitoring and controlling can also be made more effective with the help of automatic toll collection system if any. Automation of toll collection system eliminates the delay on traffic roads and tolls electronically. Presently used electronics sensors like optical, electromagnetic, radar and sonic etc are very costly. These necessitate the designing of low cost system for traffic monitoring and automatic toll collection centre [1], [6].

- **Intelligent Mobile Robot:** Intelligent Mobile Robot has ability to determine its own position in its frame of reference and then to plan a path towards some goal location. For planning the path; object localization, speed and direction are required to avoid the collision and navigating safely in the environment [3].

- **Multimedia Animation and Video Application:** Automatic motion capture algorithm in Camera Tracking is used in broadcasting and cinema for visual effect creation [1].

Currently, very few algorithms exists that can perform motion detection and classification reliably and efficiently with high accuracy and execution speed for the real time applications. This thesis is an attempt to provide an effective and robust solution which is used to identify the object and also to extract the motion parameters from the moving object from image sequences.

1.1 Machine Intelligence System

The Machine Intelligence system extracts the information from image or image sequences stored in digital computer. The image data can take many forms, such as

video sequences, views from multiple cameras, or multi-dimensional data from medical scanners etc. Machine Intelligence systems are linked with Artificial Intelligence (AI) which is the key computer technology applied to manage the knowledge and human resources.

Since the mid-1980s, there has been sustained development of the core ideas of artificial intelligence, e.g. representation, planning, reasoning, natural language processing, machine learning, and perception. In addition, various sub-fields have emerged, such as research into autonomous, independent systems (hardware or software), distributed or multi-agent systems, coping with uncertainty, effective computing/models of emotion, motion and manipulations, general intelligence etc. Figure 1.1 shows the traits which have received the most attention [2].

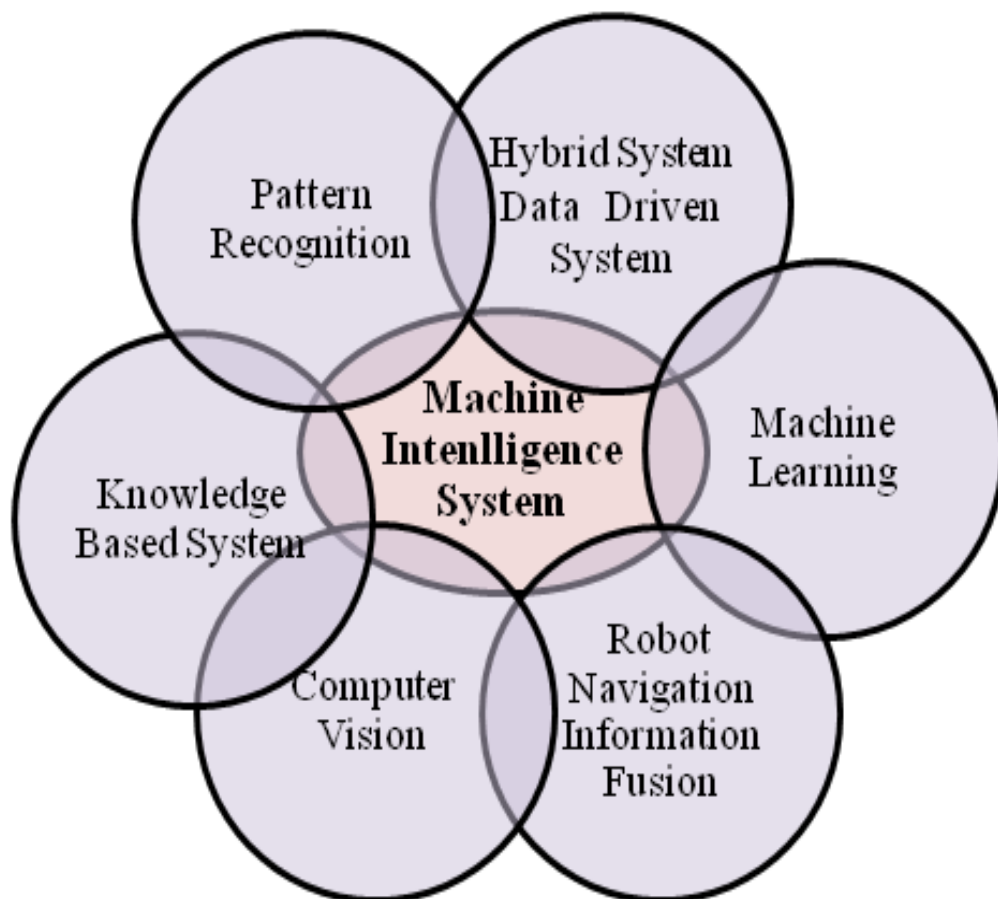


Figure 1.1 : Machine Intelligence System

The fragmentation of AI into specialized sub-fields has produced powerful component methods for standalone algorithms in almost many of the research areas. Many research papers have been published about the different algorithms in different domains, but relatively less attention has been paid to situations involving multiple domains. It is needed to combine sub-field technologies of AI towards the construction of integrated cognitive systems that mimic broad human-level intelligence.

Challenges for developing Machine Intelligence Software System based on image analysis:

- Inferring three-dimensional structure from two-dimensional images is inherently ambiguous.
- Reflection from the object surface, inhomogeneous illumination and discontinuity of the reflecting surface at the object borders are potential causes for uncertain information due to the spatial changes in the intensity.
- Occlusions are the frequent source of the ambiguity.

Limitations:

Machine Intelligence System consists of two main components: Image capturing and Analysis of captured image. Image capturing can be done easily with 2D CCD image sensors with millions of pixels. Line cameras, logarithmic image sensors, CMOS sensors are also used for high resolution, high dynamic range, and low power consumption. However the cost involvement is more in all these devices. While analyzing of the captured image, there is a problem on two aspects: speed and quality of the processing images. Cameras and other image capturing devices produce large amounts of data. Although processing speed and storage capabilities of computers have been increased tremendously in the last decade, processing high resolution images and videos are still a challenging task. Much more work has been carried out for offline or desktop processing, but very few algorithms have been developed for

real time applications. Depending on the applications, Intelligence systems try to extract different aspects of the information contained in an image or a video stream [6]. For example, moving objects are discarded to infer a structural object model from a sequence of images, whereas for the control of mobile robots, analysis may start with motion model of objects.

Two main approaches exist for interpretation of images: bottom-up and top-down. Bottom up approaches are mostly used for recognition and classification systems like character recognition system, biometric recognition system etc. The top-down approach for image analysis start with the object models rather than image. Top-down techniques are used for image registration and for tracking of objects in image sequences [6]. The hypothesis can be generated by predictions which are based on the analysis results from the preceding frames. Recognizing the objects and perceiving actions of the objects are prerequisites for Machine Intelligence System. The Machine Intelligence System involves mainly two tasks: Object Identification and Visual Tracking.

1.1.1 Object Identification

Object Identification or Object classification of moving objects in video streams is the first relevant step of information extraction in many computer vision applications. Object Recognition in machine vision is the task of finding an object in the given image or video sequence. Humans recognize an object with little effort. But for a machine, an image is a projection of 3D structure to 2D. Differing appearance of the same object with variation in the different viewpoints, viewing distance, scaling, translation, rotation, varying illumination, cluttered background, intra-category appearance variations etc. make the task difficult. Object should be recognized even when they are partially obstructed from view. This task is still a challenge in machine vision.

A complete Object recognition system consists of following Modules [4] as shown in the Figure 1.2:

1. Data Acquisition
2. Pre-processing
3. Feature Extraction and Feature Selection
4. Model Selection and Training
5. Performance Evaluation

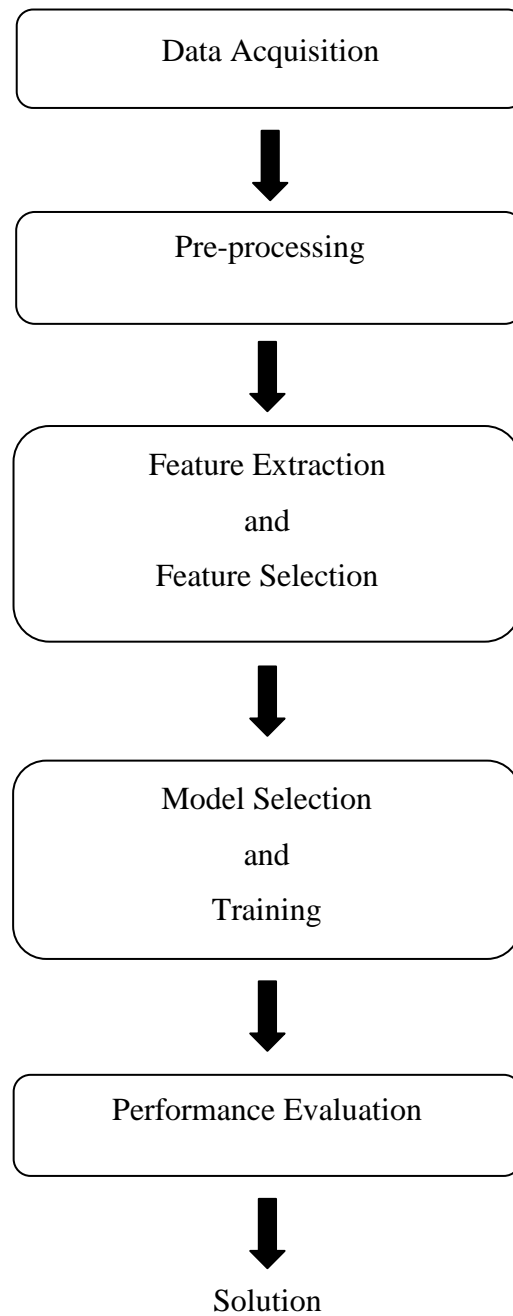


Figure 1.2 : Modules of Object Recognition System

1. Data Acquisition: One of the most important requirements for designing a successful object recognition system is to have adequate and representative training and testing datasets. A sufficient amount of training dataset required to learn a decision boundary as a functional mapping between the feature vectors and the correct class labels. There is no rule that specifies how much data is sufficient. Designer must select the types of sensors or measurement schemes that provide the data such that it should be able to design the classifier effectively [4].

2. Pre-processing: The goal is to condition the acquired data such that noises from various sources are removed as much as possible. Various filtering techniques are used if the user has prior knowledge regarding the spectrum of the noise. Conditioning may include the normalization of the data with respect to the mean and variance of the amplitude of the data normally called feature value. Pre-processing used for deblurring, image enhancement or edge detection depends upon the application domain [4], [6].

3. Feature Extraction and Feature Selection: The goal of this step is to find preferably small number of features that are particularly distinguishing or informative for the classification process and that are invariant to irrelevant transformations of the data. Better discriminating information may reside in the spectral domain or frequency domain. Feature extraction is usually obtained from a mathematical transformation of the data. In the spatial domain, feature descriptors are extracted using colors, geometric primitives like line and circles or textures. In the Frequency domain, feature extractions are performed by applying Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT) etc. Some of the methods find intensity discontinuity points which are invariant to rotation, translation, scale [1], [4].

4. Model Selection and Training: After acquiring, pre-processing and extracting the most informative features of the training dataset, classifier and training algorithms are selected. Classification can be considered as function approximation problem which can use variety of mathematical tools, such as optimization algorithms. Most common object recognition algorithms use statistical approaches or Neural Network

approaches. Statistical pattern recognition uses Bayes Classifier, Naïve Bayes Classifier, K-Nearest Neighbour (KNN) classifier etc. Neural Network approaches use Multi-layer Perceptron (MLP), Radial Basis Function (RBF), Support Vector Machine (SVM), Self Organizing Map (SOM) [1],[3],[4] etc.

- **Statistical Approach:**

Bayes Classifier: Bayes Classifier uses data points those are assumed to be drawn from a probability distribution, where each pattern has a certain probability of belonging to a class, determined by its class conditioned probability distribution. A given d-dimensional $x = (x_1, \dots, x_d)$ needs to be assigned to one of the c classes w_1, \dots, w_c . The feature vector x that belongs to class w_j is considered as an observation drawn randomly from a probability distribution conditioned on class w_j , $P(x|w_j)$. This distribution is called the likelihood probability. The Bayes theorem takes the prior likelihood of each class into consideration. Disadvantages of Bayes classifier is the difficulties in estimating the likelihood probabilities, particularly for high dimensional data. To overcome the problem Naïve Bayes Classifier is used. The main advantages of Naïve Bayes classifier is that it only requires univariate density to be computed, which are much easier to compute than the multivariate densities. The main disadvantage is that dependencies among the class cannot be modeled by Naïve Bayesian classifier [4].

K-Nearest Neighbour (KNN) Classifier: KNN classifier can be used as a nonparametric density estimation algorithm, and is most commonly used as a classification tool [2],[3]. The top k matches are then used to obtain a classification. Typically some sort of majority vote is used to determine the label assigned to the query object. This classification requires no training, although the value of k needs to be determined somehow. If it is too small, the classifier becomes sensitive to noise and if it is too large the computational time increases and becomes biased towards the classes with the larger number of members. A commonly used version of the k -NN is the Nearest Neighbour (NN) classifier where k is equal to one. The disadvantage of this scheme is that the quality of

results depends on the training set. These algorithms have no computational cost of training but more computational cost during the testing of features.

- **Neural Network Approach:**

Multi-Layer Perceptron (MLP): The Multi-Layer Perceptron is one of the most popular classification techniques. Jones [5] showed that a MLP with just two layers using a sigmoid activation function can approximate any function to an arbitrary error. The main disadvantages that, MLP suffers from the computational cost as it increases at an exponential rate as number of dimensions increases. The MLP works by propagating an input pattern through a number of layers with varying numbers of nodes. Each node has a weight assigned to it and has some activation function assigned to it. The activation function of a MLP determines what sort of functions it can represent. If the activation function is linear, then the network is no more powerful than a single layer network. A sigmoid activation function is used which performs a non-linear mapping allowing much more powerful networks to be built. Typically MLPs are trained using the back-propagation algorithm. This algorithm takes into account the individual weightings within the network and can choose the best change to get the weights per iteration. MLP better handles the classification type problems [1],[3],[4].

Radial Basis Function Network (RBF): The Radial Basis Function network classifier [1], [4] is a technique that relies upon casting the classification problem into a much higher dimensional space than the input vector in order to increase the likelihood of creating a linearly separable problem. An RBF network consists of a number of input nodes, a hidden layer and an output layer. The hidden layer typically uses a Gaussian activation functions to perform a non-linear transform. This layer will also contain many more nodes than the input to cast into the higher dimensions. The output layer consists of a number of linear activation functions. The RBF uses a randomly initialized set of weights as it gives the different result each time it is trained. The training process should be able to reduce the effects of initial conditions if enough numbers of iterations are performed. Training a RBF network is faster than training a MLP network. Training is split into two fast

stages. The first stage uses an unsupervised method to determine the parameters of the basis functions. The final stage solves a linear problem, mapping the hidden layer to outputs. RBF performs well on function approximation problems.

Support Vector Machine (SVM): The Support Vector Machine (SVM) is a popular and powerful classification technique [4]. It is a kernel based technique which does not suffer from the curse of dimensionality like those other classification techniques. Due to the kernel nature of the support vector machine, different types of network can be built, such as polynomial learning machines, radial-basis function networks and two-layer Perceptron. An attribute particular to SVMs is that they can provide good generalization performance even though they do not incorporate problem-domain knowledge. The SVM is traditionally a Binary Classifier. This method significantly increases computation expense as the number of classes increase. The one-versus-many method trains one classifier for each class. Training examples are labeled with '1' if they are of the target class or '-1' otherwise. The label associated with the classifier returning the largest positive distance from the decision boundary is selected to make the classification. The one-versus-one method trains a classifier on every possible pairing of classes. The Final classification is made by a voting process where each classifier can vote for one of the two classes. The winning class is then used to make the classification. The third method, DAG (Directed Acyclic Graph)-SVM, is similar to the one-versus-one method except a directed acyclic graph is constructed such that each classifier is a node in the tree and the leaves are the resulting classes. This reduces the number of classifications required whilst keeping a similar level of performance.

Self Organizing Map (SOM): Self Organizing Map (SOM) transforms an input pattern of arbitrary dimension into a one- or two-dimensional discrete map and performs this transformation in a topologically ordered fashion[1], [4]. The SOM algorithm is simple to implement, however very difficult to analyze mathematically. There are three essential processes called competition, co-operation, and synaptic adaptation. During competition, neurons in the network compute their respective clause of a Discriminant function. This Discriminant

function provides the basis for competition among the neurons. The neuron with the largest value of discriminant function is declared the winner. During co-operation the winning neuron determines the spatial location of a topological neighbourhood of excited neurons, providing the basis for co-operation. During synaptic adaptation the excited neurons increase their individual values of the discriminant function in relation to the input pattern through suitable adjustments to their synaptic weights. Adjustments are made so that a similar pattern returns an enhanced discriminant value.

5. Performance Evaluation: To get a good estimate of the generalization ability of a classifier, different methods like split-sample, cross-validation, Boot strapping are used. Performance evaluation can be done by splitting the entire dataset into two parts, training dataset and testing dataset where the training dataset is used for actual training and testing dataset is used for testing the true field performance of the algorithm. Since adequate and representative training dataset is highly important for successful training, no rules are fixed about the size of training and testing dataset [2], [3].

Split-Sample: A commonly used method for classifier training and validation is Split-Sample Validation [6]. The data set is split into a training and validation set (often a 50% - 50% or 75% - 25% split). The classifier is trained using the training set, and validation is performed on the validation set. The greater the number of samples, the closer to the true error the estimate will be. This means that for small numbers of samples the estimate is likely to be inaccurate.

Cross-validation: Cross-validation has been used to select classifier training parameters [7] and to estimate the generalized performance of a particular set of parameters. In this situation, training and testing data sets are produced. The training data set is further partitioned into an estimation and validation set. For each set of parameters, a classifier is trained using the estimation set and its performance evaluated using the validation set. The best performing set of parameters is then selected and its performance is evaluated using the test data set to avoid problems of over fitting the training data.

Alternatively cross-validation can be used to provide a better generalization estimate than split-sample when small data sets are available. One approach that has been accepted to provide the reasonably good estimate of the true generalized performance uses k-fold cross-validation. In k-fold cross validation, the entire available training dataset is split into $k > 2$ portions, creating k blocks of data. Among these k blocks, $k-1$ blocks are used for training. This procedure is repeated k times using different blocks for testing each case. The average performance is declared as the estimate of the generalized performance of the algorithm. It is computationally expensive for larger data sets or high values of k .

Boot Strapping: Boot strapping is similar to cross-validation except that it uses sub-samples of the data set instead of sub-sets. A sub-sample is random sampling with replacement of the original data set allowing sub-samples to be of nearly any size as required. This is useful when data sets are unbalanced or too small [2].

1.1.2 Visual Tracking

In the Visual Tracking, the motion of an object and interpretation of the object behavior are performed from image sequences or consecutive video frames. Object motion analysis has attracted a great interest due to its promising applications in the real world. Motion Parameters like location, directions and speeds are derived for the learning of Machine Intelligence System. Visual tracking task becomes complicated due to static occlusion, dynamic occlusion, varying size and shape of objects in video sequences. Also object tracking algorithm must be capable of handling trajectory part like static occlusion, dynamic occlusion and tracking complications like splitting and merging as well as object appearance and disappearance successfully. For video tracking, the task is difficult when objects are moving fast relative to the frame rate. Complexity of the problem increases when tracked object changes shape and orientation over a time. For these situations 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.

Typical motion tracking system is described as shown in the Figure 1.3. Motion tracking algorithms mainly involve Motion Segmentation, and Tracking [3].

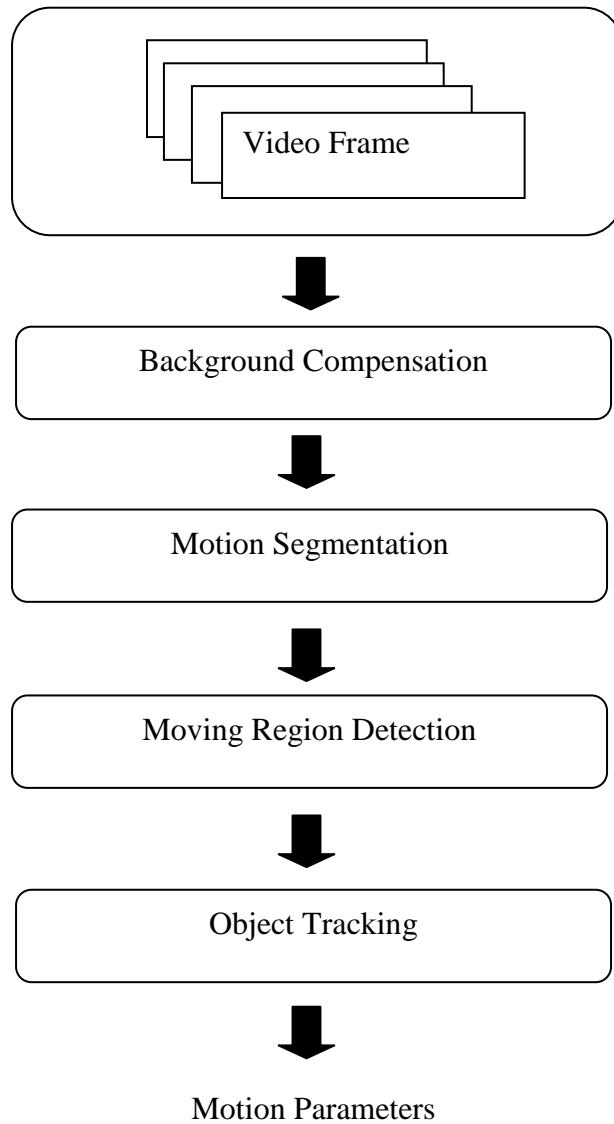


Figure 1.3 : Modules of Visual Tracking System

Motion segmentation aims at decomposing an image in objects and background to find the motion of the object. The information like textures or statistical descriptors, edges, colors can be extracted from a single image which used for object segmentation. Typical segmentation techniques such as region growing, splitting and

merging, watershed, and histogram based algorithms, active contours, graph partitioning, and level sets [8], are some of the most widely used techniques.

Motion segmentation can be done in many ways:

Background Subtraction is particularly popular method with a relatively static background. It is extremely sensitive to changes of dynamic scenes due to lighting. To avoid the problem, a procedure is used to update the background model. Each image in current image can be classified into foreground and background by comparing the statistics of current background model. This approach is popular due to its robustness to noise, shadow, change of lighting conditions etc. Intensity thresholding, Gaussian mixture models are used for background model [3], [34].

An approach of temporal differencing makes use of pixel-wise difference between two or three consecutive frames in an image sequence to extract the moving regions. Temporal differencing is very adaptive to dynamic environments and having poor result of extracting features. An improved version uses three frames differencing instead of two frame differencing [8], [9].

Optical flow is generally used to describe coherent motion of points or features between image frames. Motion segmentation based on optical flow uses characteristics of flow vectors of moving objects over time to detect change of regions in an image sequence. For Most flow computation, methods are computationally complex and very sensitive to noise, and cannot be applied to video streams in real-time without specialized hardware [9].

Tracking can be employed using two approaches: Shape based tracking in which image blob area, blob bounding box are considered for tracking. Motion based tracking considers periodicity property of the object motion or time frequency analysis of the object [1], [17].

Tracking over time typically involves matching objects in consecutive frames using features such as points, lines or blobs. That is, tracking may be considered to be

equivalent to establishing coherent relations of image features between frames with respect to position, velocity, shape, texture, colour, etc [3]. Blob Tracking involves detection of blobs of the object interior using thresholding technique or block matching techniques. Model based Tracking uses the geometric structure of an object and can be represented as stick figure, 2D contour or volumetric model. For small number of objects, model based approaches are efficient. Since it depends upon the geometric model it is not able to handle the occlusion as geometric model is 2D model. Feature based Tracking method uses sub-features such as distinguishable points or lines on the object to realize the tracking task. Its benefit is that even in the presence of partial occlusion, some of the sub-features of the tracked objects remain visible. Feature-based tracking includes feature extraction and feature matching. Low-level features such as points are easier to extract. It is relatively more difficult to track higher-level features such as lines and blobs. Active Contour Based Tracking uses active contour models, or snakes, aims at directly extracting the shape of the subjects. The idea is to have a representation of the bounding contour of the object and keep dynamically updating it over time. It reduces the computational complexity compared to the region based tracking. Initializations of contours for moving objects are quite difficult, especially for complex articulated objects. Region based tracking is used to identify a connected region associated with each moving object in an image, and then track it over time using a cross-correlation measure. Region-based tracking approach has been widely used today. Video tracking has been difficult in congested situation.

1.2 Overview of the Proposed Work

This thesis is an attempt to develop effective solutions for object tracking with recognition. It has been observed that the existing methods offer scope for improvement. The objective is to propose a new and efficient approach suitable for Machine Intelligence System which identifies the object and extract the motion parameters by analysis of visual tracking of the object. Development of the software model is targeted to the applications such as automated Visual surveillance systems,

Traffic monitoring system; self guided vehicles, automatic guided machines and different robotics application under the normal conditions. The thesis presents the software model with the following components:

- Perform the motion segmentation.
- Tracking of the moving objects by locating them in each frame of the sequence.
- Classification of the object.
- Extraction of motion parameters such as location, direction and speed.

Research is conducted in each area of motion segmentation, motion tracking and object classification. A suitable approach for each of these components is chosen by analyzing the strengths and weaknesses of the reviewed techniques. Each process has been implemented and tested with real image sequences, and combined to produce an efficient machine intelligence system which can automatically detect, classify and track the object.

1.2.1 Contribution

To develop the software model for object identification and estimation of motion parameters, two different tasks are considered: Object Classification and Visual Tracking. After an exhaustive comparative study of available alternatives for each method, several enhancements to achieve efficient results for both the tasks have been proposed. Basically these include choice of best approaches for the task and proposed modifications to these approaches for improving their results.

Object Classification:

Context understanding is the key element when developing an information retrieval model for machine intelligence application. An efficient model is required that

encodes and classifies video objects such as humans, vehicles, buildings, etc. Thus, it is required to design a generalized object classifier to identify or classify the objects of interest based on the application.

For designing the general Classifier system, a novel feature based object classification using Discrete Contourlet Transform and Principal Component Analysis (PCA) has been proposed. Feature extraction has been carried out in the frequency domain. For better result compared to the conventional discrete Contourlet Transform, pre processing and filtering stages are applied which enhance the edge details of the images. Feature extractions are performed with the preprocessed images that give more efficient result than the discrete Contourlet transform method. For efficient edge point feature extraction, Unsharp Filter is used before feature extraction. Unsharp filter amplifies the high frequency components which enhances the edges of an image.

Feature extraction coefficients are extracted by applying Curvelet transform that overcomes the problem of representing an image with smooth contours in different directions by providing two additional properties: directionality and anisotropy [28] as compared to the Discrete Wavelet Transform (DWT). The Curvelet Transform was developed initially [28] in the continuous domain via Multiscale filtering followed by a block based Ridgelet transforms applied to the subband images. Since it is a block based transform, either the approximated images have blocking effects or overlapping windows are required for calculations that increase the redundancy. Also the use of the Ridgelet transform, which is defined on a polar coordinate, makes the implementation of the Curvelet transform for discrete images on a rectangular coordinate to be very challenging. The second generation Curvelet transform [71] was defined directly via frequency partitioning without using the Ridgelet transform. Both Curvelet transforms require a rotation operation and correspond to a 2D frequency partition based on polar coordinate. This makes the Curvelet construction simple in continuous domain but caused the implementation for discrete images sampled on a rectangular grid to be very challenging [75]. This makes algorithm implementation difficult. This fact leads the development of a directional multiresolution transform like Curvelet, but directly in the discrete domain. Contourlet, as proposed by Do and

Vetterli [23], form a discrete filter bank structure that can deal effectively with piecewise smooth images with smooth contours. This discrete transform can be seen as a discrete form of a Curvelet Transform. In the discrete Contourlet transform, the image is decomposed by a double filter bank structure where the first filter bank captures the edges and second filter bank link the edge points into the contour segments.

Eigenvalue (Principal Component Analysis) of feature matrix has been calculated from the feature matrix that helps for dimensionality reduction for feature matching which increases the execution speed of algorithm. The results with discrete Contourlet with PCA are compared with Discrete Curvelet Transform with PCA. For feature matching, Euclidian Distance Measure and Neural network classifier is used to match the test feature vector with the trained feature vector and compared for analysis.

Visual Tracking:

An efficient method is to be developed to track all the moving objects with high accuracy. The method should be adaptable using ordinary camera for designing the cost effective application system. Further the motion parameters like direction; speed etc. should be extracted from data obtained. Two different algorithms proposed are: Single object visual tracking and multiple object visual tracking.

(1) Single object visual tracking: User selected object has been tracked in the video sequences. 3D Colour histogram and Euclidean distance measures are used for object tracking. Novel Block matching algorithm has been proposed to find the location of the object being tracked in the video sequence frames. Object without motion and object motions out of boundary conditions are also included in the algorithm.

(2) Multiple object visual tracking: Multiple object tracking has been performed using the statistics from data obtained with Blob analysis. Blob segmentation has been carried out by background subtraction and Thresholding. As Blob analysis includes domain independent information, for establishing temporal relationship between the block, colour segmentation is used. Template matching is implemented

using 3D histogram and Hu's seven Invariant moments to track the object. Different termination and decision conditions are included for making algorithm fast and efficient.

1.2.2 GUI for the Proposed Method

GUI for the proposed method has been designed with user friendly features. Each task is designed and implemented in a such a way, that it can be used for many applications like Object Recognition, Finger Print recognition and similar other applications also. Different Parameters for different tasks can be selected individually. The Object Identification task can be used for other object classifier applications also. In the GUI of the Object Classification task, user can obtain the training images and testing images by selecting the folder which contains the images. For training the dataset, user has the options for choosing the Curvelet or Contourlet transform approaches. For visual tracking, the Contourlet transform is implemented, which is faster than the discrete Curvelet transform. For traffic monitoring application, vehicle classifier has been designed using three class structures to improve the efficiency.

For tracking task, user can select the single object tracking method or multiple object tracking. For single object tracking, user needs to select the object of interest using mouse. The Tracking of the object is visualized using rectangular bounding boundary. The multiple object tracking algorithms tracks the moving objects with boundary. It also displays the name of object, object speed in terms of pixels, and direction of the object. Camera calibration has been done for actual speed measurement of vehicle and implemented.

1.3 Layout of the Thesis

The content of the thesis is summarized as:

Chapter 1: This chapter describes the brief history and overview of the problems. It also introduces the objective of the research work and scope of the improvement in the existing methods.

Chapter 2: Various approaches and methods are discussed related to the research problem in the object classification and visual tracking tasks for machine intelligence application. It also covers the suitability conditions, merits and demerits of the existing methods in Literature Survey.

Chapter 3: Describes the Proposed Methods with algorithms in each task: Object Classification and Visual Tracking. Two different approaches have been proposed for Visual Tracking of moving objects: Single object tracking and multiple object tracking.

Chapter 4: Discusses the detailed results of the proposed method and its comparison with the results of the conventional method.

Chapter 5: Concludes with the remarks regarding proposed solutions and their applicability under various situations. Future work possible in the area is also suggested.

Chapter 2

2 Background and Related work

Object tracking can be defined as the process of segmenting an object of interest from a video scene and keeping track of its motion, orientation, occlusion etc. in order to extract useful information. Visual tracking of the objects attempts to detect, track and identify the people or vehicles and interpret the object behavior from image sequences involving the objects.

In visual tracking two different approaches are merged together: Designing of Classifier and Motion Analysis of moving object. Different approaches used to classify the object are reviewed in the first section of this chapter. Different approaches used for designing the modules of object classifier like feature extraction, distance measures and performance metrics are reviewed.

In the second section, different algorithms and techniques used for visual tracking have been discussed. Different motion segmentation approaches using different background model and tracking methods have been reviewed and the merits and demerits of the each method have been discussed. From the different methods and reviews, the best approaches in terms of results and execution speeds are combined to make the efficient algorithms for object classification and visual tracking task.

2.1 Designing of Classifier

Designing of Classifier task involves different modules like feature extraction and distance measures. To validate the effectiveness of classifier, different performance metrics are measured. This section covers the literature review on the related work done so far in the above area.

2.1.1 Feature Extraction

A pattern recognition system that adjusts its parameters to find correct decision boundaries, through a learning algorithm using a training set such that a cost function (mean square error between numerically encoded values of actual and predicted labels) is minimized can be referred as a classifier or model [6].

Object Classification task uses two types of learning method: Supervised and Unsupervised learning. Supervised learning is the term used to describe the training of a classifier with target data available for the training set. The aim of object classifier is to find the correct mapping between input data and the target data. Unsupervised learning does not use target data. The goals of learning are finding clusters in data or modeling distributions as opposed to find a mapping.

Feature Extraction (A set of variables which carries discriminating and characterizing information about an object) and Feature Selection algorithms are mainly important for object classification. There are basically three approaches used for feature extraction [2], [4], [6]:

1. Geometry based approaches
2. Feature Point based approaches
3. Appearance based approaches

1. Geometry based approaches: Geometrical model based feature extraction can be done by extracting the geometric primitives like lines, curves or circles. They cannot handle the variation in the lighting and view points with certain occlusions. An excellent review on geometry based object recognition has been discussed in Mundy [10]. This paper reviews the key advances of the geometric era and the underlying causes of the movement away from formal geometry and prior models towards the use of statistical learning methods based on appearance features. Although geometry based approaches are invariants to view points and illumination, dependency and complexity on statistical functions have made limited use of the method.

2. Feature Point based approaches: The main idea of feature point based object recognition algorithm lies in finding interest points, often occurring at intensity discontinuities that are invariant to change due to scale, illumination and affine transformation. Feature Point based approaches find the different points that are invariant to the affine, rotation, translation or scaling. Various Feature Based Algorithms are reviewed such as Harris Corner Detector (HCD) [11], Scale Invariable Feature Transform (SIFT) [12], Speed up Robust feature Transform (SURF) [13], Random Sample Consensus (RANSAC) [14] etc.

Harris Corner Detector (HCD) method uses a combined corner and edge detector method based on the local correlation function to find out the image regions containing texture and isolated features. It shows good consistency and performance over a natural image. Scale Invariant Feature transform are invariant to image scaling, translation, rotation and partially invariant to illumination changes and affine or 3D projection.

Scale Invariant Feature Transform (SIFT) consists of four major stages: scale-space extrema detection, key point localization, orientation assignment and key point descriptor. The first stage identifies key locations in scale space by looking for locations that are maxima minima of a difference-of-Gaussian function [12]. Each point is used to generate a feature vector that describes the local image region sampled relative to its scale-space coordinate frame. Image keys are created from the feature vector that allow for local geometric deformations by representing blurred

image gradients in multiple orientation planes and at multiple scales. The keys are used as input to a nearest-neighbour indexing method that identifies candidate object matches. Final verification of each match is achieved by finding a low-residual least-squares solution for the unknown model parameters.

Speeded Up Robust Feature (SURF) is a robust image detector & descriptor, first presented by Herbert Bay [13]. SIFT and SURF algorithms employ slightly different ways of detecting features. SIFT builds an image pyramids, filtering each layer with Gaussians of increasing sigma values and taking the difference. SURF is based on sums of approximated 2D Haar wavelet responses and makes an efficient use of integral images. It calculates determinant of Hessian blob detector, from which it uses an integer approximation. These computations are extremely fast with an integral image.

Random Sample Consensus (RANSAC) proposed by Fischler and Bolles [14] is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers. It is a non-deterministic algorithm which produces a reasonable result only with a certain probability. The probabilities increase as more number of iterations is allowed. In RANSAC a procedure exists which can estimate the parameters of a model that optimally explains or fits in the small data.

SURF is less time consuming than SIFT where as RANSAC is invariant to affine transform. SIFT based methods are expected to perform better for objects with rich texture information as sufficient number of key points can be extracted but require sophisticated indexing and matching algorithms for effective object recognition. An advantage of RANSAC is the ability to do robust estimation of the model parameters, i.e., it can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set. A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters and a good initialization is needed.

3. Appearance based approaches: Better discriminating information may reside in the spectral domain or frequency domain. Most recent appearance based techniques

involve feature descriptors and pattern recognition algorithms in the frequency domain.

Most widely used approaches perform linear transformations such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA also known as Karhunen-Loeve transformation, most commonly used as dimensionality reduction technique in pattern recognition. It was originally developed by Pearson in 1901 [15] and generalized by Loeve in 1963. PCA does not take class information into consideration. The classes are best separated in the transformed space better handled by LDA, which consider inter cluster as well as intra cluster distances in the classification. Principal Component Analysis is used with two main purposes. First, it reduces the dimensions of data to computationally feasible size. Second, it extracts the most representative features out of the input data so that although the size is reduced, the main features remain, and still be able to represent the original data [15], [16].

A concept of Eigen picture was defined to indicate the Eigen functions of the covariance matrix of a set of face images. Turk and Pentland [17] have developed an automated system using Eigenfaces with the similar concept to classify images in four different categories, which helps to recognize true/false of positive of faces and build new set of image models. For night time detection and classification of vehicle, Thi et al. used Support Vector Machine with Eigenvalue [18]. Sahambi and Khorasani [19] used a neural network appearance based 3D object recognition using Independent component analysis. The Eigenfaces approach has been adopted in recognizing generic objects across different viewpoints and modeling illumination variations [20].

The frequency domain analysis is more attractive as it can give more detailed information about the signal and its component frequencies. Over the past 10 years, the wavelet theory has become one of the emerging and fast-evolving mathematical and signal processing tools for its many distinct merits. Different from the Fast Fourier transform (FFT), the wavelet transform can be used for multi scale analysis of the signal through dilation and translation, so it can extract the time-frequency features of the signals effectively.

Wavelet transforms have been used in the past for time series classification [21]. Originally it was proposed to use DFT to map the time domain function to frequency domain. The wavelet transform [22] is expressed as decomposition of a signal $f(x) \in L^2(R)$ a family of functions, which are translations and dilations of a mother wavelet function $\psi(x)$. The 2D filter coefficients can be expressed as

$$\begin{aligned} h_{LL}(m,n) &= h(m)h(n), & h_{LH}(k,l) &= h(k)g(l) \\ h_{HL}(m,n) &= g(m)h(n), & h_{HH}(k,l) &= g(k)g(l) \end{aligned} \quad (2.1)$$

Where, the first and second subscripts denote the low-pass and high-pass filtering respectively along the row and column directions of the image. Wavelet transform can be implemented (convolution and down sample) along the rows and columns separately. 2D discrete Wavelet Transform is performed using low-pass and high-pass filters. After the decomposition four subbands, LL,LH,HL and HH are obtained, which represent the average (A), horizontal (H), vertical (V), and diagonal (D) information respectively. The iteration of the filtering process produces multi level decomposition of an image. Wavelet transforms provide the effective multi scale analysis but are not effective to represent the image with smooth contours in different directions. For acquiring more directional information, Multiscale Geometric Analysis (MGA) tools were proposed such as Curvelet [28], Ridgelet [24], Bandlet [28] and Contourlet [23] etc.

Contourlet transform [23] is a multi scale and directional image representation that uses first a wavelet like structure for edge detection, and then a local directional transform for contour segment detection. A double filter bank structure is used for obtaining sparse expansions for typical images having smooth contours. In the double filter bank structure, Laplacian Pyramid (LP) is used to capture the point discontinuities, followed by a Directional Filter Bank (DFB), which is used to link these point discontinuities into linear structures. The Contourlet have elongated supports at various scales, directions, and aspect ratios. This allows Contourlet to

efficiently approximate a smooth contour at multiple resolutions. Nonsubsampled Contourlet was pioneered by Do and Zhou as the latest MGA tool [24] in 2005. Yan et al. [25] proposed a faced recognition approach based on Contourlet transform. Yang et al. [26] proposed a multisensor image fusion method based on Nonsubsampled Contourlet transform. Extensive experimental result show that proposed scheme by Yan's based on Contourlet Transform performs better than the method based on stationary wavelet transforms [23]. Srinivasan Rao, Srinivas Kumar and Chatterji [27] used feature vector using Contourlet Transform for Content Based Image Retrieval System

Candes and Donoho [28] introduced a new multiscale transform named Curvelet transform which was designed to represent edges and other singularities along curves much more efficiently than traditional transforms, i.e., using fewer coefficients for a given accuracy of reconstruction. Implementation of Curvelet transform involves the steps: (1) Subband decomposition, (2) Smooth partitioning (3) Renormalization (4) Ridgelet Analysis. There are two separate Fast Discrete Curvelet Transform (FDCT) algorithms introduced by Starck, Candes and Donoho [29]. The first algorithm is called the Unequally-Spaced Fast Fourier transform (FDCT via USFFT), where the Curvelet coefficients are found by irregularly sampling the Fourier coefficients of an image. The second algorithm is the wrapping transform, which uses a series of translation and a wrap around technique. The wrapping FDCT is more intuitive and has less computation time. Use of the Curvelet Transform for Image Denoising is explained by Starck, Candes and Donoho [29]. A comparative study based on wavelet, Ridgelet and Curvelet based texture classification is well explained by Dettori and Semler [30].

Contourlet transform can represent information better than Wavelet transform for the images having more directional information with smooth contour [23] due to its properties like directionality and anisotropy. Curvelet transform represents edges and other singularities along curves much more efficiently [28]. These two methods have been selected to extract the features for performing the object classification task and also for comparison.

2.1.2 Distance Measures

In order to establish the similarity or closeness of two feature vectors in some feature space, a wide range of distance matrices are used. A distance matrix calculates the distance between two point sets in matrix space [31].

- **Minkowski Norms**

The most commonly used distance matrices are the Minkowski norms. It is defined based on the L_p norm. The Norms are popular for their simplicity, speed of calculation and quality of results. Similarity Distance d between two feature vectors is calculated using the following equation:

$$d_p(x, y) = ((\sum_{i=1}^N |x_i - y_i|^p)^{\frac{1}{p}}) \quad (2.2)$$

where $x = \{x_1, x_2, \dots, x_N\}$ and $y = \{y_1, y_2, \dots, y_N\}$ are the query and targeted feature vectors respectively. N is the number of elements in the vectors.

When $p=1$, $d_1(x, y)$ is the city block distance also known as Manhattan distance (L_1)

$$L_1 = d_1(x, y) = \left| \sum_{i=1}^N |x_i - y_i| \right| \quad (2.3)$$

When $p=2$, $d_2(x, y)$ is the Euclidean distance (L_2) and calculated as

$$L_2 = d_2(x, y) = ((\sum_{i=1}^N |x_i - y_i|^2)^{\frac{1}{2}}) \quad (2.4)$$

- **Histogram Intersection**

The histogram intersection is another simple distance matrix that is often used. It was proposed by Swain and Ballard [84]. Their objective was to find known objects within

images using color histograms. It is able to handle partial matches when the size of the object with feature vector x is less than the size of the image with the feature vector. The histogram distance d is given as

$$d_{hist}(x, y) = 1 - \frac{\sum_{i=1}^N \min(x_i, y_i)}{\min(|x|, |y|)} \quad (2.5)$$

Colors not present in the query image, do not contribute to the intersection distance. This reduces the contribution of background colors. The sum is normalized by the histogram with fewest samples.

- **Bhattacharyya Distance**

A statistical measure known as , Bhattacharyya Distance measure is often used for comparing two probability density functions, which are most commonly used to measure color similarity between two regions [85]. It is very closely related to the Bhattacharyya Coefficient, which is used to measure the relative closeness of the two samples taken into consideration. Bhattacharyya distance can be calculated as

$$d_{bha}(x, y) = \sum_{i=1}^N \sqrt{x_i} \sqrt{y_i} \quad (2.6)$$

Where x_i and y_i are the probability density function.

- **Cosine Distance**

The cosine distance computes the difference in direction, irrespective of vector lengths. The distance is given by the angle between the two vectors [84]. By the rule of dot product the distance can be calculated using the equation (2.8).

$$x \cdot y = |x| \cdot |y| \cos \theta \quad (2.7)$$

$$d_{cos}(x, y) = 1 - \cos \theta = 1 - \frac{x \cdot y}{|x| \cdot |y|} \quad (2.8)$$

- **Chessboard Distance**

The Chessboard or Chebyshev distance is the maximum distance between the components of two points. This measure creates a space similar to the Manhattan distance but rotated [85].

$$d_{che} = \max_i (|x_i - y_i|) \quad (2.9)$$

- **Mahalanobis Distance**

The Mahalanobis distance is a special case of the quadratic-form distance matrices in which the transform matrix is given by the covariance matrix obtained from a training set of feature vectors, that is $A = \Sigma^{-1}$. In order to apply the Mahalanobis distance, the feature vectors are treated as random variables $\mathbf{X} = [x_1, x_2, \dots, x_N]$, where x_i is the random variable of i^{th} dimension of the feature vector [31].

Mahalanobis distance between two feature vector x and y can be calculated as

$$d_{mah} = [(x - y)\Sigma^{-1}(x - y)]^{\frac{1}{2}} \quad (2.10)$$

In the special case where x_i are statistically independent, but have unequal variances, Σ is a diagonal matrix as shown in the equation (2.11).

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_N^2 \end{bmatrix} \quad (2.11)$$

The Mahalanobis distance is reduced to a simpler form [31]:

$$d_{mah} = \sum_{i=1}^{N-1} \frac{(x_i - y_i)^2}{\sigma_i^2} \quad (2.12)$$

Chi-squared, Kullback-Leibler, Earth Mover's Distance, χ^2 Statistics and Quadratic Distances are also well known distance measures used for different applications. The choice of distance metric to be used greatly depends upon application. For general usage, the Minkowski norms will often be a good choice. For applications where speed is preferred over accuracy, the L1 norm or histogram intersection can be used. For applications where the different components cannot be assumed to be independent, a metric such as the Mahalanobis distance may be preferable. In this thesis, Euclidean distance measure has been used for feature matching of training dataset and testing dataset. Bhattacharya distance measure is used for object tracking using the color features.

2.1.3 Performance Matrices

There are many ways of evaluating the performance of a classifier system. A commonly used statistic to measure the performance is 'accuracy'. There are several versions of the accuracy statistics. The basic statistic just measures the percentage of correct classifications out of all the classifications. Accuracy per class and per classification can also be found using statistic [1], [4], [74].

- **Confusion Matrix**

Typical performance measure statistics are calculated using a confusion matrix. This records the true and predicted classification of each object. The two class confusion matrix records four values. The True Positive (TP) value is the number of positive examples correctly classified. Likewise the True Negative (TN) value is the number of negative examples correctly classified. The False Negative (FN) value is the

number of positive examples classified as negative and the False Positive (FP) value is the number of negative examples classified as positive.

The User's accuracy (Precision) is the number of correct classifications over all the objects classified as that class.

$$Accuracy = \frac{TP}{TP + FP} \quad (2.13)$$

The Producers accuracy is also known as recall or sensitivity. Sensitivity is the number of correct classifications over all the objects of that class.

$$Producers\ accuracy = \frac{TP}{TP + FN} \quad (2.14)$$

Specificity measures the proportion of negative examples correctly classified.

$$Specificity = \frac{TN}{FP + TN} \quad (2.15)$$

- **Receiver Operating Characteristics Graph**

A Receiver Operating Characteristics (ROC) graph [32] is a visual tool to evaluate classifier performance. A key feature is that it is invariant to class distribution, The ROC graph plots true positive rate against false positive rate. ROC calculates the overall accuracy of a classifier; it does not gauge the accuracy of an individual classification.

- **A Priori and A Posteriori methods**

The work by Giacinto and Roli [33] highlights a priori and a posteriori methods as good confidence estimators. These techniques make use of a validation set. If the k

nearest objects in a validation set were correctly classified, then it is likely that the query object also classified correctly. A priori method estimates the confidence without requiring the query to be classified. It simply bases the confidence on how many of the neighbouring objects were correctly classified. A posteriori method requires the query object to be classified first. After that, on the bases of the estimation of confidence on number of the neighbouring objects, the classes are predicted correctly.

2.2 Visual Tracking

Most commonly used visual tracking techniques include Motion Segmentation and Object Tracking. Background subtraction is one of the most common and effective methods of segmenting foreground objects from the background scene. The process of background subtraction involves locating the areas in an image which differ sufficiently, from an image of the background.

A background modeling process has three phases:

1. Model representation – kind of model used to represent the background.
2. Model initialization – Initialization of assumed model.
3. Model adaptation – Mechanism for adapting to illumination changes in the background.

2.2.1 Background model

A number of different approaches to these issues have been proposed. At the naive level, a simple frame difference can be used to obtain the moving objects [34]. In the simplest method, the background model is just the previous frame in the image

sequence. The foreground objects are isolated by comparing the difference in color between the current image and the previous image to a threshold value. If the difference exceeds the threshold value, the pixels are referred as part of the foreground. The strength of this approach is the simplicity of the background model. Limited processing power and memory are required to maintain the background model as only the previous frame in the sequence needs to be remembered. It can provide adequate results in situations where the background scene is relatively static but problems arise when the background is constantly changing. In an attempt to address this issue, a number of adaptive background models have been proposed. Considering static motion for the application, simple background approach has been selected.

Background adaptation may be classified as either predictive or non-predictive [35]. Predictive algorithms are known to model the scene as a time series and they would make use of a dynamic model to recover the current input based on past observations. The absolute error between the predicted and the actual observation can then be used as a measure of change. Non-predictive methods try to build probabilistic representation from the observations at a particular pixel. An alternative way for classifying background adaptation methods is either non-recursive or recursive [36]. A non-recursive technique uses a sliding-window approach for background estimation. For non-recursive estimation the L previous video frames are first stored in a buffer and then a background image is constructed making use of the temporal variation of each pixel in the buffer. Non-recursive method requires a large storage memory for slow moving objects. Recursive techniques do not rely on a buffer for estimating the scene. Instead, they recursively update a single or multiple background model based on each input frame. Even though recursive method requires less memory, any error in background model remains for a longer time. To alleviate this problem exponential weighting and positive decision feedback can be used. Non-recursive adaptation techniques include temporal differencing (frame differencing), average filtering, Median filtering and Minimum-Maximum filtering. Recursive techniques on the other hand include Approximated Median filtering, Single Gaussian, Kalman Filtering, Mixture of Gaussians, Clustering based segmentation methods, and Hidden Markov Models.

2.2.2 Statistical Approach

The background as a realization of a random variable with Gaussian distribution (SGM - Single Gaussian Model) is represented by Wren et al.[37]. The mean and covariance of the Gaussian distribution are independently estimated for each pixel in SGM. Stauffer and Grimson [38] presented an adaptive background mixture model for real-time tracking. In their work, they modeled each pixel as a mixture of Gaussians and used an online approximation to update it. The Gaussian distributions of the adaptive mixture models were then evaluated to determine the pixels most likely from a background process, which resulted in a reliable, real-time outdoor tracker which can deal with lighting changes and clutter. A study by Haritaoglu et al. [39] built a statistical model known as w4 which represents each pixel with three values: its minimum and maximum intensity values and the maximum intensity difference between consecutive frames observed during the training period. The model parameters were updated periodically.

Lehigh Omni directional Tracking System (LOTS) presented by Boulton et al. [40] is tailored to the detection of non cooperative targets under non stationary environments. This algorithm uses two gray level background images. This allows the algorithm to cope up with intensity variation due to noise or fluttering objects which move in the scene. Each pixel of the input frame is compared to the closest background value and classified as active if the difference exceeds a given threshold.

2.2.3 Object Tracking Techniques

The ability of the model to handle shadows and changing lighting conditions is also increased by utilizing the differing properties of the Color information. Different color spaces also have different advantages when performing background subtraction. The HSV color spaces separate a RGB image into its hue, saturation and value or intensity components. The $YCbCr$ color space is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance

information is stored as two color-difference components (C_b and C_r). C_b represents the difference between the blue component and a reference value. C_r represents the difference between the red component and a reference value.

Most commonly used visual tracking techniques include Mean Shift Tracking Algorithms [41], [42], Blob Tracking, Particle Filter Algorithms [43], Block Matching [44], [45] and Optical Flow Based Tracking Algorithms [3]. Mean shift tracking algorithm has become popular due to its simplicity and robustness. Tracking is accomplished by iteratively finding the local minima of the distance measure functions using the mean shift algorithm. The mean shift algorithm was originally invented by Fukunaga and Hostetler [46] for data clustering, which they called a “valley-seeking procedure”. It was first introduced into the image processing community several years ago by Cheng [47]. Mean Shift based on Color Distribution and Simulated Annealing (SACD-MS), is proposed for human body tracking by Hong [48]. R. Venkatesh Babu [49] proposed a new method to track objects by combining two well-known trackers, Sum-of-Squared Differences (SSD) and color-based mean-shift (MS) tracker. Zoran [50] applied a new 5-Degree Of Freedom (DOF) color histogram based non – rigid object tracking algorithm using Expectation Maximization (EM) Mean shift. Huiyu Zhou [51] proposed the method based on SIFT features and mean shift to track the object. Disadvantage of Mean shift algorithms is to specify the kernel for further tracking. Also some time mean shift tracking algorithms gets stuck at local minima. The similarity measures like Bhattacharya coefficients and Kullback - Leibler divergence are not very discriminative, especially for higher dimensions [52] and difficult to use them due to the sample based calculation for the real time object tracking.

Particle filters [46], [47] are kind of stochastic tracking algorithms that use multiple discrete “particles” to represent the distribution over the location of the target. It has been shown to be very suitable for performing tracking in cluttered environments due to their ability of maintaining multiple hypothesis of probability distribution. More importantly, particle filters exhibit superior characteristic of recovering from the temporary lost track. Sanjeev Arulampalam [53] proposed a method based on Particle Filter using Bayesian Tracking for the Nonlinear/Non-Gaussian tracking problem.

Hybrid tracker [43] using particle filter and Mean Shift are used by Bo Zhang, Weifeng Tian, and Zhihua Jin. Tang Sze Ling [54] described the characteristic of the motion tracker based on color as the key feature to compare the object's similarity for object detection and tracking. Lowe [55] used model based object tracker using Marr–Hildreth edge detector to extract edges from an image. Stanley T. Birchfield and Rangarajan [56] presented a particle filtering framework for region-based tracking using spatiogram. M. A. Zaveri, S. N .Merchant and U. B. Desai [57] proposed a neural-network-based tracking algorithm. Erdem [44] proposed the method based on “defocus energy” which is utilized for automatic segmentation of the object boundary and it is combined with the histogram method to track the object more efficiently. Since particle filters require a large number of particles for accurately representing the probability distribution, it limits their applications to real time occasions.

In the field of Motion Estimation (ME), many techniques have been proposed [58],[59],[60],[61],[62],[63],[64],[65],[66]. Basically ME techniques can be broadly classified as: gradient techniques, pixel-recursive techniques, block matching techniques and frequency-domain techniques.

Among these four groups, block matching is particularly suitable in video compression schemes based on Discrete Cosine Transform (DCT) such as those adopted by the recent standards H.261, H.263 and MPEG family [59],[60]. Block-based motion estimation uses a Block-Matching Algorithm (BMA) to find the best matched block from a reference frame. The basic idea of BMA is to divide the current frame in video sequence into equal-sized small blocks. For each block, we try to find the corresponding block from the search area of previous frame, which “matches” most closely to the current block. Therefore, this “best-matching” block from the previous is chosen as the motion source of the current block. The relative position of these two blocks gives the so-called Motion Vector (MV), which needs to be computed and transmitted. When all motion vectors of the blocks in tracking area have been found, the motion vector happened most frequently is chosen for the correction of tracking area size. Typically, the Sum of Absolute Difference (SAD) is selected to measure how closely two blocks match with each other, because the SAD

doesn't require multiplications; in other words, less computation time and resources are needed. There are several methods used to find out the best matching block.

The most commonly used Block Matching Algorithm (BMA) is the Full search (FS)/Exhaustive search (ES), which exhaustively searches for the best matching block within the search window. Full Search Algorithm is the most straight forward strategy. But the computational complexity of Full Search is always too high. As a result Fast Search Algorithm has been developed. In fast BMA using a fixed set of search patterns, the assumption is that, the matching error decreases monotonically as the search moves towards the position of the global minimum error and the error surface is uni-modal. Few fast block matching motion estimation algorithms were Two-Dimensional Logarithmic Search, Three Step Search [61], Four Step Search [62], Block-Based Gradient Descent Search [63], Diamond Search (DS) [64], Cross-Diamond Search (CDS) [65] etc. Adaptive Rood Pattern Search (ARPS) is proposed in [67] to track large motions, with less number of computations by using Zero Motion Prejudgment (ZMP) for the reduction in the computation complexity. Novel Hexagon-based Search (NHS) [66] algorithm has been incorporated in recently developed H.264/AVC video coding standard. These methods are mainly used for image compression but for object tracking. These methods are more time consuming than the standard mean shift and particle filter techniques.

Other object tracking methods involve the shape based and motion based tracking. Cutler and Davis [68] proposed a method that used the periodic shape changes that occur during the walking motion. To analyze the periodic nature of a particular object, its appearance throughout the image sequence must be remembered. A similarity measure between each object image is then generated. If the motion is periodic, this similarity measure will also be periodic as the appearance of objects will repeat. The Fourier transform of the similarity measure can be used to identify peaks in the power spectrum corresponding to the fundamental frequencies of the motion. If a peak exceeds some threshold value, the motion is regarded as periodic. Cutler and Davis also suggest a method to distinguish different types of periodic motion by comparing the similarity images to those generated by a training set. In this fashion, they are able to classify motion as human, animal, or other. While this

approach provided reliable classification results, the method is memory intensive, requiring an image of each object in every frame to be stored. Calculating the self similarity between each of these images is also computationally expensive. Another problem with using periodic motion as a classifier is that it is only effective when the subject is moving. If the subject pauses to look at something or talk to another subject, the decision made by the classifier is unreliable.

A method outlined by Lipton [69] uses an optical flow based technique to classify moving objects as rigid or non-rigid. This is achieved using the observation that rigid objects will generate less residual flow than non-rigid objects during non-rotational motion. To calculate the residual flow of a moving body, its net motion, defined as the absolute position change of the object, was determined using a tracking algorithm. The optical flow vector for each pixel in the object was then computed. The residual flow of the body was calculated by subtracting the net motion of the body from the optical flow vector associated with each pixel. Rigid bodies have little residual flow as all pixels that make up the object are moving in the same direction. The optical flow of each pixel is approximately equal to the net motion of the body, resulting in a small residual flow value. Non-rigid objects will display greater residual flow as some pixels that make up the object are moving in different directions to the overall body. The optical flow directions of these pixels are different to the net motion, resulting in a larger residual flow value. Thus, Lipton distinguished between rigid objects such as vehicles and non-rigid objects such as humans.

In the summary with this chapter, background work and literature survey based on the object classification and motion analysis have been described. Based on the literature survey, efficient and computationally fast algorithms are selected for implementation of the proposed algorithm.

Chapter 3

3 Proposed Approach

The proposed Visual Tracking algorithm identifies moving object and also tracks the moving objects. Estimation of the motion parameters such as location, direction and speed of the moving objects are derived from the image sequences with the help of CCD Camera. Two different approaches are proposed for single object tracking and multiple object tracking. To overcome the problems in Mean Shift Tracking Algorithm, a novel Block Matching Tracking Algorithm using Predictive Motion Vector based on 3D color histogram has been proposed and implemented efficiently for single object tracking. To improve the efficiency of the multiple objects tracking over the conventional methods like particle filtering and Kalman filtering, improved blob tracking method has been proposed. Multiple trackers have been used for region tracking of the objects. Feature based on Contourlet transform and color features with invariant moments are used for region tracking. Object Identification has been carried out by edge feature extraction and Contourlet transform with Principle Component Analysis (PCA) and compared with Curvelet Transform with PCA.

Effective techniques have been implemented for object Identification, background subtraction, motion segmentation; color descriptor and feature extraction for region tracking algorithms.

The proposed techniques involve mainly two tasks:

- Designing of an Object Classifier algorithm for Object Identification that has been used for Visual Tracking Process.
- Designing of Visual Tracking Algorithms for Estimating the Motion Parameters.

3.1 Object Classifier

Generalized Classifier has been designed that can be used in many applications. Classifier has been designed using two approaches: (1) Fixed sizes of the Objects in the dataset like Face dataset and Fingerprint dataset. (2) Varying size of the objects in the dataset like Vehicle dataset. It has different sizes according to the viewing angle for which three Class structures have been designed according to the length and width ratio. Designing of object classifier involves mainly two tasks: (A) Training of Classifier and (B) Object Identification of Query Image.

(A) Training of Classifier

As shown in the Figure 3.1 following steps are performed during Training of classifier:

- (1) Resize all the images of dataset.
- (2) Pre-processing to get the sharp images from the given dataset and perform the feature extraction of enhanced images. Flowchart including the steps for feature extraction is explained in the Figure 3.2.
- (3) Generate Eigen matrix for dimensionality reduction for fast retrieval.

(B) Object Identification of Query Image

Object Identification task involves the steps as shown in the Figure 3.3:

- (1) Resize the unidentified image to the same size as training dataset.

- (2) Do Pre-processing to get the sharp images and perform the feature extraction of enhanced images.
- (3) Project the image into Eigenspace using the Eigen matrix of trained classifier.
- (4) Compute Similarity measure using the Euclidean distance classifier or neural network classifier for best match feature vector from Eigen matrix of trained dataset.
- (5) Identify and label the objects using best match feature vector.

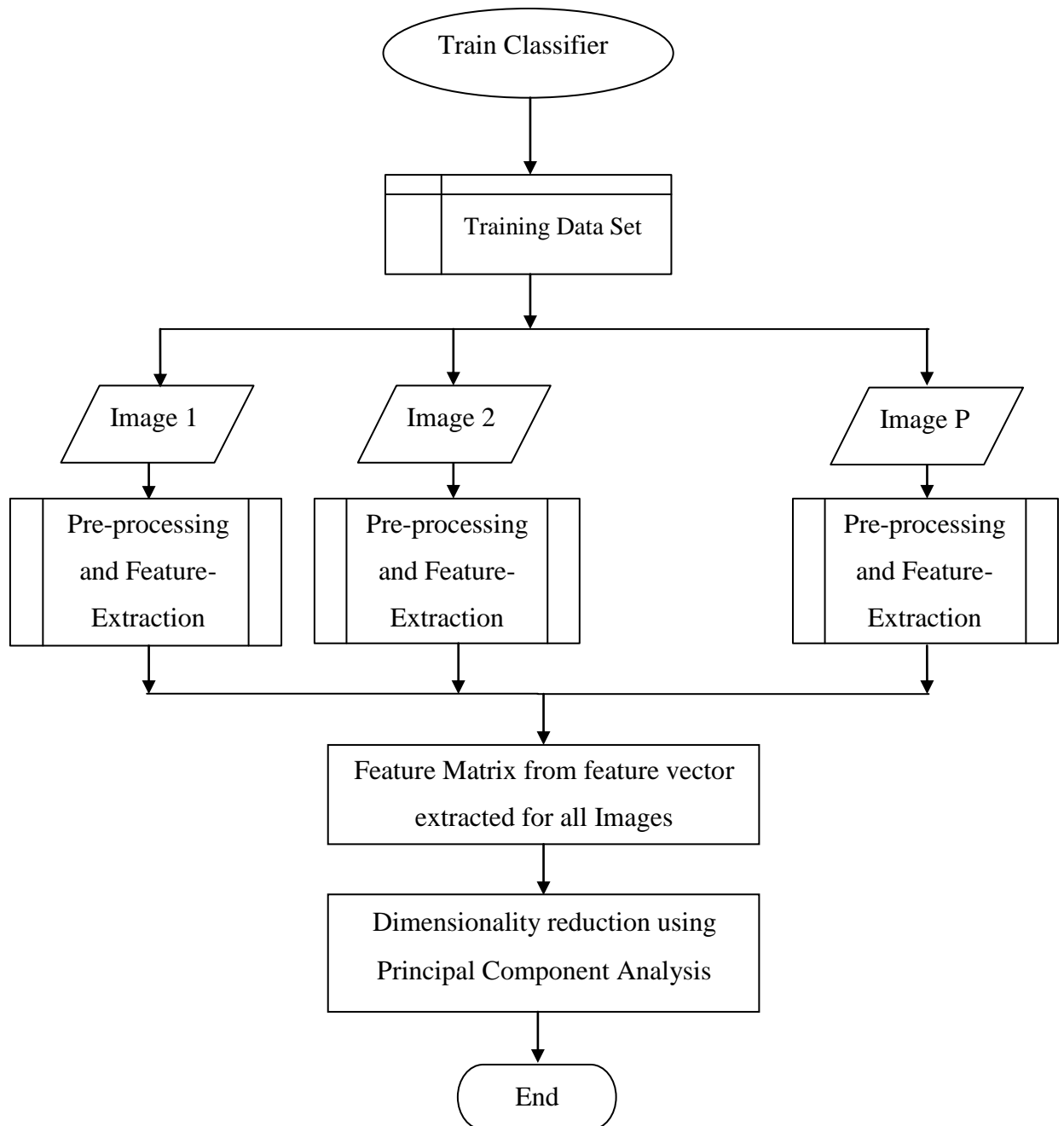


Figure 3.1 : Eigen Matrix Generation for Training Dataset

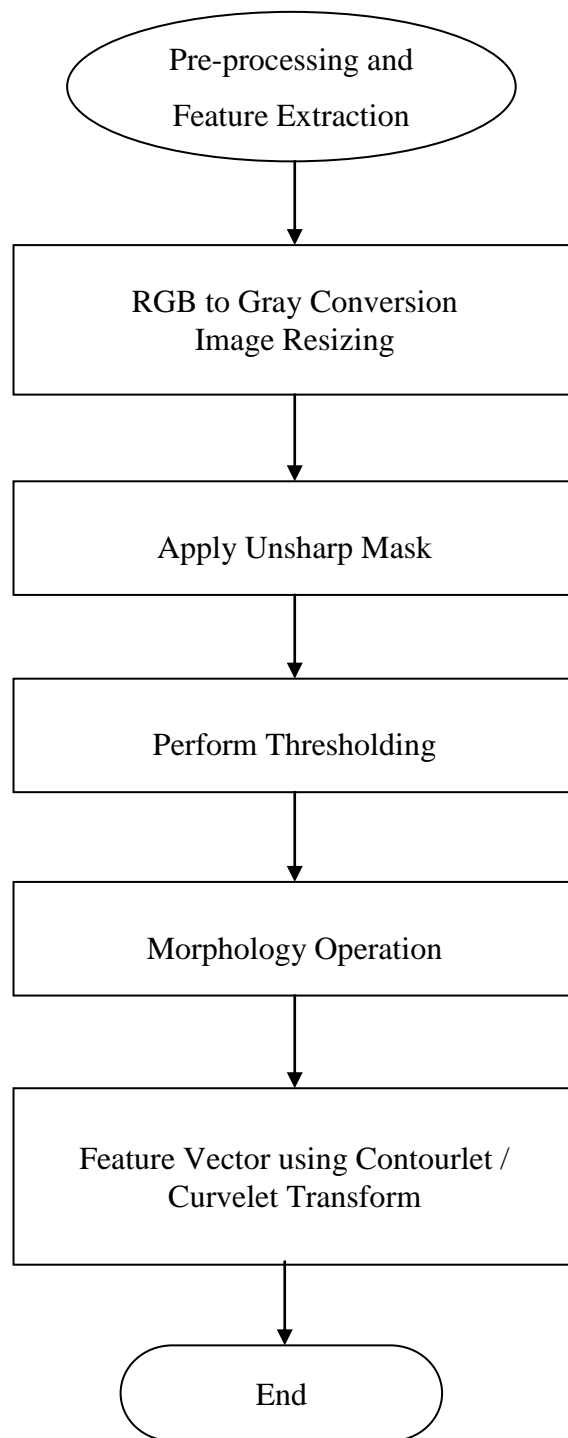


Figure 3.2 : Pre-processing and Feature Extraction

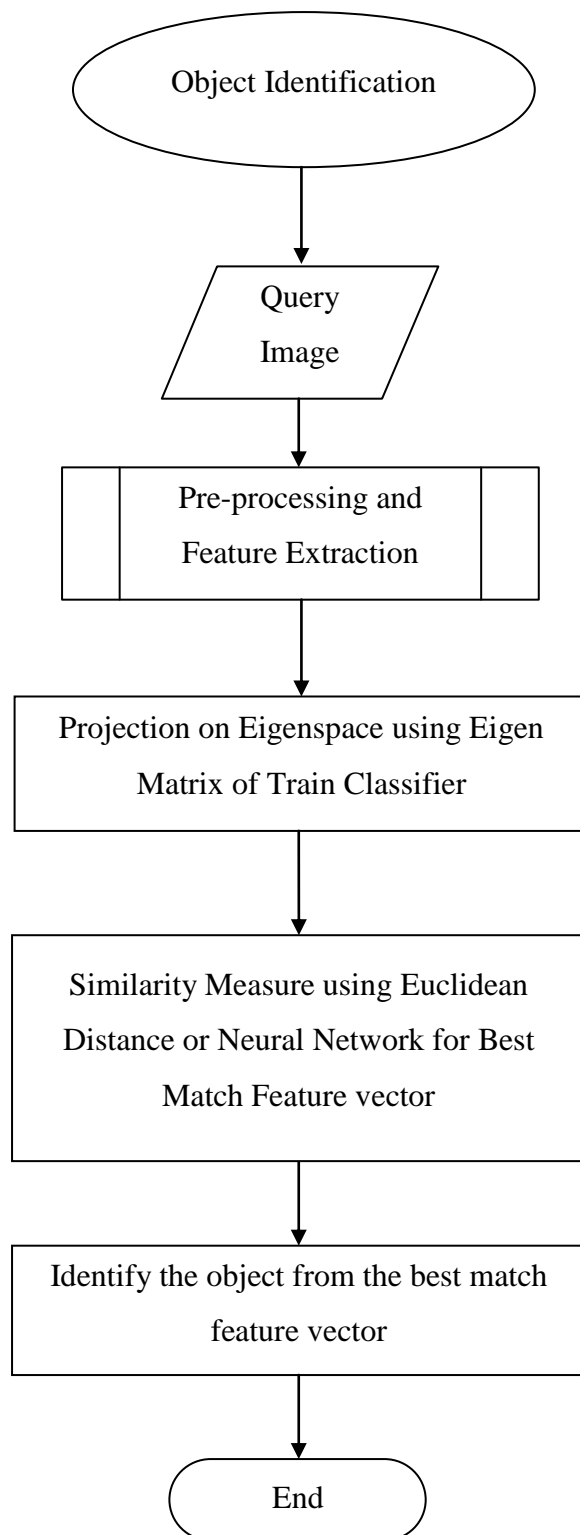


Figure 3.3 : Object Identification of Query Image

The following subsections include the mathematical models involved in each stage which are explained briefly. The subsections describe Pre-processing, Feature Extraction, and Principal Component Analysis for Eigen matrix generation and similarity measure for feature matching.

3.1.1 Pre-processing

3.1.1.1 Unsharp Filter

The Unsharp filter is a simple sharpening operator that enhances edges and amplifies high frequency components in an image via a procedure which subtracts an unsharped or smoothed version of an image from the original image [87]. Let S be the dataset having P images for training and q images for testing. Color image $f1(m,n)$ of size $m \times n$ is converted into the gray scale image. Unsharp masking produces an edge image $g(m,n)$ from an input image $f1(m,n)$ by performing negative of Laplacian filter $f_{smooth}(m,n)$ as shown in the Figure 3.4.

$$g(m,n) = f1(m,n) - f_{smooth}(m,n) \quad (3.1)$$

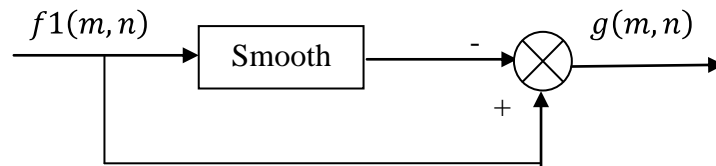


Figure 3.4 : Spatial Sharpening

Convolution has been performed with unsharp mask U and the image $f1(m,n)$ to get the edge image $g(m,n)$.

$$U = \frac{1}{(\alpha+1)} \begin{bmatrix} -\alpha & \alpha-1 & -\alpha \\ \alpha-1 & \alpha+5 & \alpha-1 \\ -\alpha & \alpha-1 & -\alpha \end{bmatrix} \quad (3.2)$$

The value of α controls the shape of Laplacian function. The range of α is from 0 to 1.

3.1.1.2 Thresholding using Otsu's Method

Thresholding has been applied on the Image after applying the unsharp filter. Global Thresholding has been applied using Otsu's method [87]. Otsu's method is one of the better threshold selection methods with respect to uniformity and shape measures. The Otsu method is optimal for thresholding large objects from the background [86].

The Otsu's algorithm performs the following steps:

1. Computes the normalized histogram of the input image. Denotes the components of the histogram by p_i where $i = 0, 1, 2, \dots, L-1$, where L denotes distinct intensity levels in a digital image of size $m \times n$, and h_i denotes the total number of pixels with intensity i . The normalized histogram p_i is calculated as

$$p_i = \frac{h_i}{m \times n} \quad (3.3)$$

2. Calculates the cumulative sums, $P1(k)$, for $k = 0, 1, 2 \dots L-1$ using

$$P1(k) = \sum_{i=0}^k p_i \quad (3.4)$$

3. Computes the cumulative means, $m(k)$, for $k = 0, 1, 2 \dots L-1$ using

$$m(k) = \sum_{i=0}^k ip_i \quad (3.5)$$

4. Computes the global intensity mean m_G using

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (3.6)$$

5. Calculates the class variance $\sigma_B^2(k)$ for $k = 0, 1, 2 \dots L-1$ using

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (3.7)$$

6. Obtain the Otsu's threshold, k^* as the value of k for which $\sigma_B^2(k)$ is maximum. If the Maximum is not unique, obtain k^* by averaging the values of k corresponding to the various maxima detected.
7. Obtain the separable measure, η^* , by evaluating the equation (3.8) at $k=k^*$.

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2} \quad (3.8)$$

Where σ_G^2 is the global variance and can be derived by

$$\sigma_G^2(k) = \sum_{i=0}^{L-1} (i - m_G)^2 p_i \quad (3.9)$$

3.1.1.3 Removing Border Objects

If the original image is considered as a mask, the marker image $f_{mask}(m, n)$ can be obtained [87] using equation (3.10).

$$f_{mask}(m,n) = \begin{cases} g(m,n) & \text{if } g(m,n) \text{ is on the border of image} \\ 0 & \text{otherwise} \end{cases} \quad (3.10)$$

The clear border image can be constructed by

$$f(m,n) = g(m,n) - f_{mask}(m,n) \quad (3.11)$$

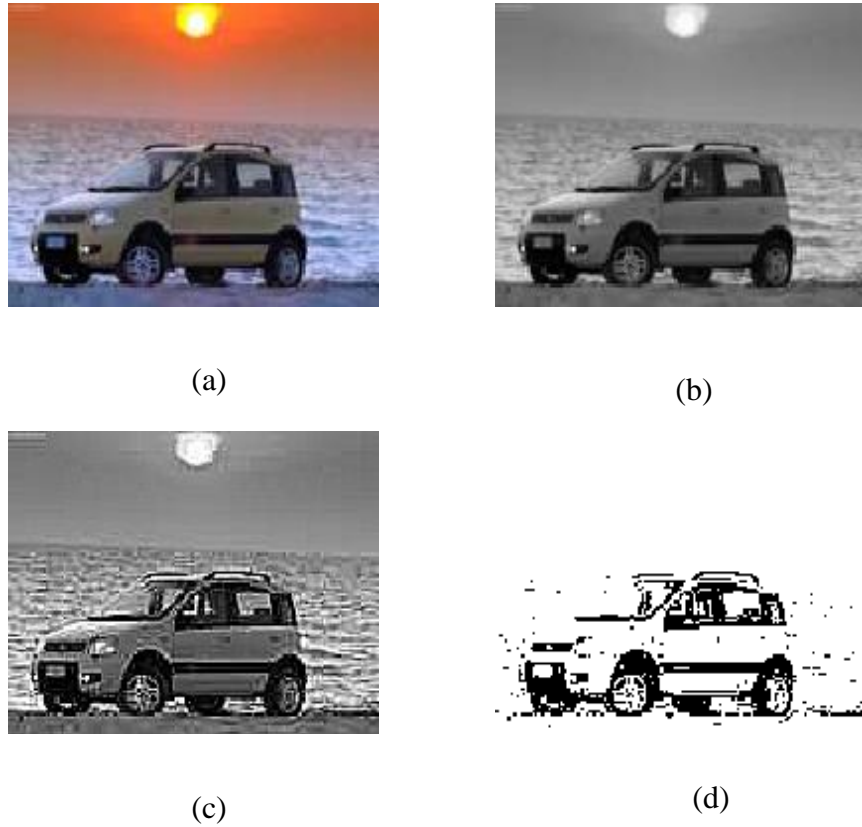


Figure 3.5 : Pre-processing (a) Image of Car (b) Gray-scale Image (c) Image after applying Unsharp filter (d) Image after applying Threshold

Figure 3.5 shows the pre-processing steps performed on each image of dataset. Figure 3.5 (a) is the original color image. Figure 3.5 (b) is the resultant gray scale image converted into 0-255 gray levels. Figure 3.5 (c) is the resultant image after performing Unsharp filter mask. Thresholding is applied on unsharp filtered image which is followed by clearing border objects. Clearing boarder removes the border

point pixels to avoid the false classification of the object. Figure 3.5 (d) shows the resultant image after performing the thresholding and clear border operations.

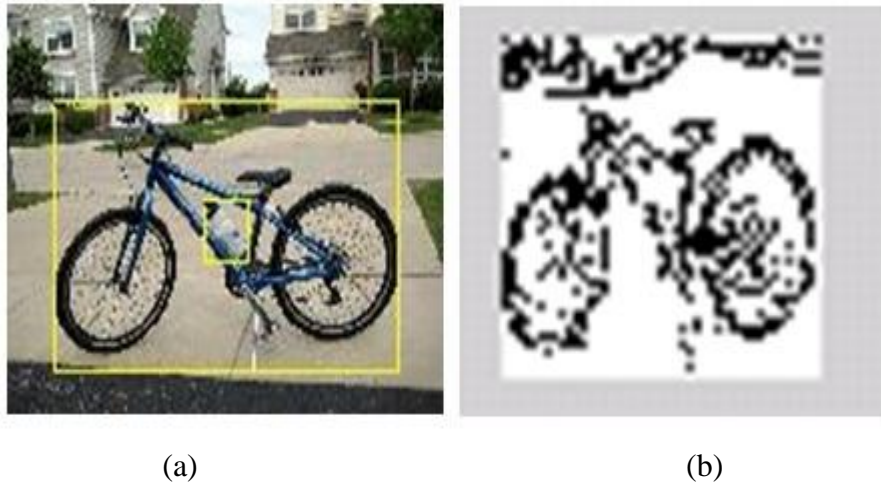


Figure 3.6 : (a) Bicycle Image (b) Pre-processed Image

Figure 3.6 (a) shows the original image on which the pre-processing has been applied. Figure 3.6 (b) shows the resultant image after performing the pre processing. The preprocessed image has been used for the feature extraction purpose.

3.1.2 Feature Extraction

Feature extraction is an essential pre-processing step in pattern recognition and machine learning problems. Feature extraction maps a larger information data space into a smaller feature space. The fundamental idea of feature extraction is to perform all computations in a smaller, simpler space.

Feature extraction pattern involves three design steps:

Feature Construction: This is the most challenging part of the pattern recognition system.

Feature Selection: This decision determines the balance between the search time and the post-processing time. For fast retrieval of dataset from feature matrix, Eigenvalues are constructed using the feature matrix in proposed methodology.

Feature Matching: This determines how fast the system can search the feature space. Euclidean distance classifier and Neural network classifier have been selected and compared. After comparison, Euclidean distance is found more efficient method while comparing recognition rate. So, finally for visual tracking Euclidean distance classifier has been implemented.

3.1.2.1 Feature Construction

After literature review, Discrete Contourlet Transform and Fast Discrete Curvelet Transform using wrapping have been found the efficient transforms for extracting the feature points. So both these transforms are implemented for feature construction and the results have been compared.

3.1.2.1.1 Discrete Contourlet Transform

Multiscale and time-frequency localization of an image is offered by wavelets. Wavelet transforms are not effective in representing the images with smooth contours in different directions. Contourlet Transform (CT) eliminates this problem by providing two additional properties known as directionality and anisotropy [23], [24].

Contourlet transform are divided into two main steps that are (1) Laplacian Pyramid (LP) decomposing and (2) Directional Filter Banks (DFB). Laplacian Pyramid decomposes the original image into a low-pass image and a band-pass image. Each band-pass image is further decomposed by DFB. The Multiscale and multidirectional decomposition of the image will be obtained by repeating the same steps upon the low-pass image [23]. Contourlet transform is a multi scale and multi directional image representation that uses a wavelet like structure for edge detection in the first

stage, and then a local directional transform for contour segment detection.

A double filter bank structure of the Contourlet is shown in Figure 3.7 .The Contourlet transforms obtains sparse expansions for images having smooth contours. In the double filter bank structure, Laplacian Pyramid (LP) [23] is used to capture the point discontinuities. Directional Filter Bank (DFB), followed by Laplacian Pyramid is used to link these point discontinuities into linear structures. The Contourlet have elongated supports at various scales, directions, and aspect ratios. These allow Contourlet to efficiently approximate a smooth contour at multiple resolutions. In the frequency domain, the Contourlet transform provides a Multiscale and directional decomposition.

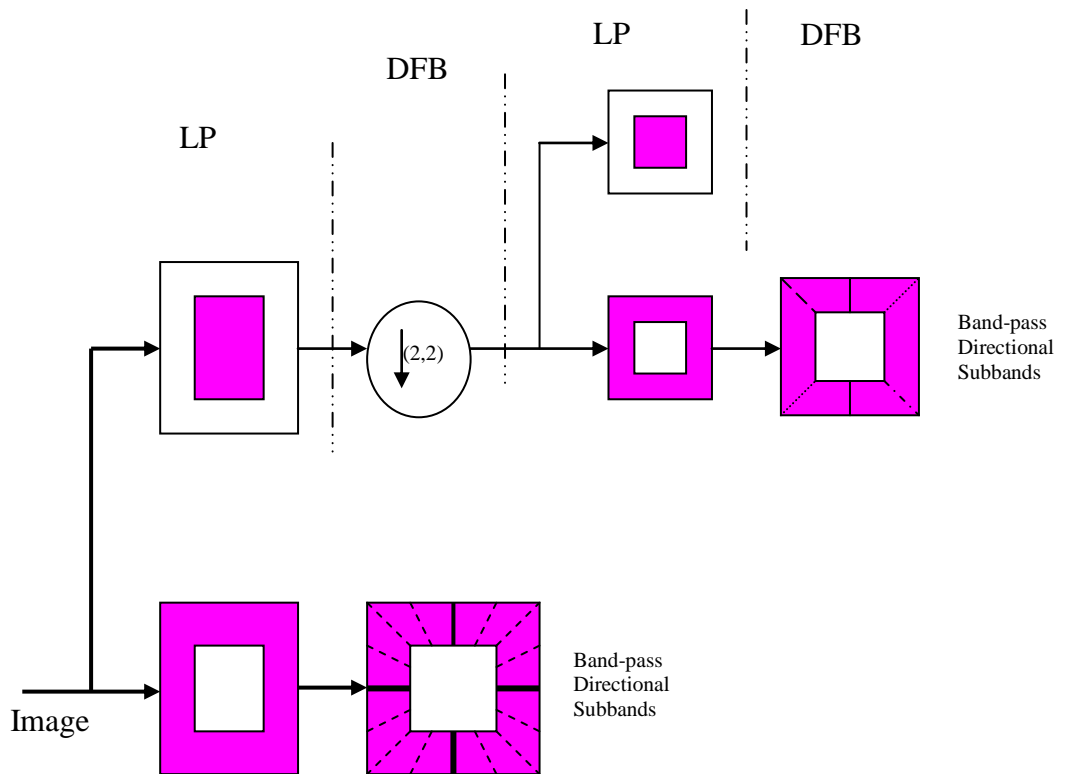


Figure 3.7 : Double Filter Bank Decomposition of Discrete Contourlet Transform

A. Pyramid frames

Multiscale decomposition is obtained by using the Laplacian Pyramid (LP) introduced by Burt and Adelson [70]. Band-pass image is obtained by first generating

the down sampled low-pass version using LP decomposition and then taking the difference between the original image and the prediction. This image is then processed by DFB stage. LP with orthogonal filters provides a tight frame with frame bounds equal to 1.

B. Directional filter banks

DFB is applied to capture the high frequency content like smooth contours and directional edges. The DFB is implemented by using a k - level binary tree decomposition that leads to 2^k directional subbands with wedge shaped frequency partitioning. DFB is constructed from two building blocks that are two channel quincunx filter bank with fan filters and shearing operator. Quincunx filter bank divides a 2D spectrum into two directions, horizontal and vertical. Shearing operators are used to the reordering of image pixels. Due to these two operations, directional information is preserved. This is the desirable characteristic in classifier system to improve retrieval efficiency.

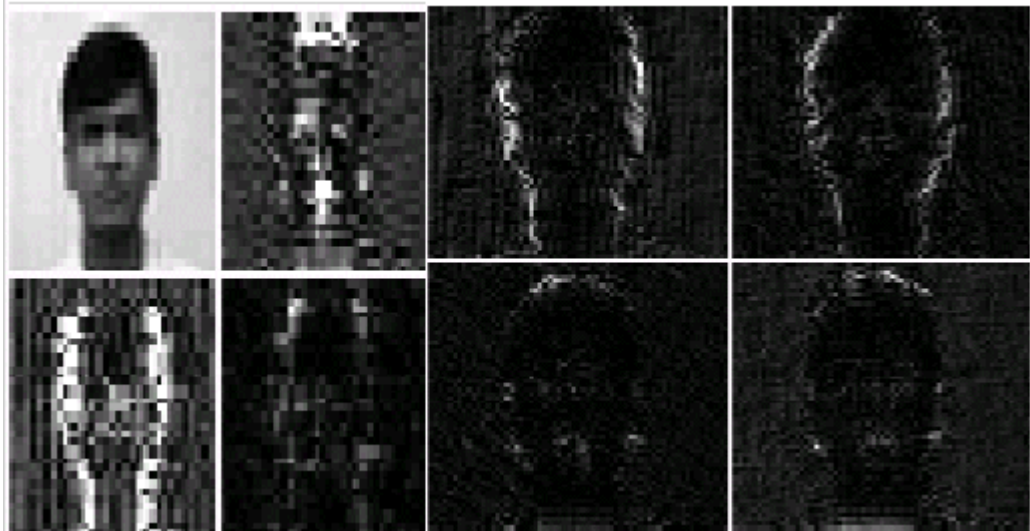


Figure 3.8 : Decomposition of Image using Contourlet Transform (2-Level and 'pkva' Filter for Pyramid and Directional Filter)

Band-pass images from the LP are applied to DFB so that directional information can be captured. The algorithm is applied on the coarse image. This combination of LP and DFB stages results in a double iterated filter bank structure known as Contourlet

filter bank .The Contourlet filter bank decomposes the given image into directional subbands at multiple scales. Figure 3.8 shows the decomposition of image using Contourlet Transform for level-2 using ‘pkva’ filter for both low-pass filter and direction filter bank.

The Contourlet Transform of two levels with ‘pkva’ filter is applied on the dataset images $f(m,n)$. Resulting image gives the decomposed coefficients as $C_1, C_{2-1}, C_{2-2}, \dots, C_{n-1}, \dots, C_{n-v}$, where v is the number of directions as shown in the Figure 3.8. These Coefficients are used to reorder the column vector I_i of the images. Image Vector I_i is constructed by converting coefficients to a column vector and then concatenation of all coefficient vectors. Let $I = [I_1, I_2, I_3, \dots, I_p]$ be the Feature Image Matrix constructed by Discrete Contourlet Coefficient, then Eigenvalue and Eigenvectors are calculated for I .

3.1.2.1.2 Discrete Curvelet Transform via Wrapping

Candes and Donoho introduced a new multiscale transform named Curvelet transform that was designed to represent edges and other singularities along the curves much more efficiently than traditional transforms by using fewer coefficients for a given accuracy of reconstruction [28],[29]. Implementation of Curvelet transform involves Subband Decomposition, Smooth Partitioning, Renormalization and Ridgelet Analysis steps [29]. There are two separate Discrete Curvelet Transform (DCT) algorithms introduced by Candes, Donoho and Demanet [71]. The first algorithm is the UnequiSpaced FFT transform (Fast Discrete Curvelet Transform via USFFT), where the Curvelet coefficients are found by irregularly sampling of the Fourier coefficients of an image. The second algorithm is the wrapping transform, which uses a series of translation and a wrap around techniques. Fast discrete Curvelet transform based on the wrapping of Fourier samples has less computational complexity as it uses fast Fourier transform instead of complex Ridgelet transform [29]. In the fast discrete Curvelet Transform via wrapping, a tight frame has been introduced as the Curvelet support to reduce the data redundancy in the frequency domain [71]. Normally, Ridgelet have a fixed length that is equal to the image size and a variable

width, whereas Curvelet have both variable width and length and represent more anisotropy. Therefore, the wrapping based Curvelet transform is simpler, less redundant and faster in computation [30] than Ridgelet based Curvelet transform.

Curvelet transform based on wrapping of Fourier samples takes a 2D image as an input in the form of a Cartesian array $f[m, n]$ such that $0 \leq m < M$, $0 \leq n < N$. It generates number of Curvelet coefficients indexed by scale j , an orientation l and two spatial location parameters (k_1, k_2) as output. To form the Curvelet texture descriptor, statistical operations are applied to these coefficients. Discrete Curvelet coefficients can be defined by [29].

$$c^D(j, l, k_1, k_2) = \sum_{\substack{0 \leq m \leq M \\ 0 \leq n \leq N}} f[m, n] \varphi_{j,l,k_1,k_2}^D[m, n] \quad (3.12)$$

Here, each $\varphi_{j,l,k_1,k_2}^D[m, n]$ is a digital Curvelet waveform. This Curvelet approach implements the parabolic scaling law on the subbands in the frequency domain to capture the curved edges within an image more effectively. Curvelet exhibit an oscillating behavior in the direction perpendicular to their orientation in the frequency domain [29].

Wrapping based Curvelet transform is a Multiscale transforms with a pyramid structure consisting of many orientations at each scale. This pyramid structure consists of several subbands at different scales in the frequency domain. Subbands at high and low frequency levels have different orientations and positions. The Curvelet is non-directional at the coarsest scale and becomes fine like a needle shape element at high scale.

With increase in the resolution level the Curvelet becomes finer and smaller in the spatial domain and shows more sensitivity to curved edges which enables it to effectively capture the curves in an image.

To achieve higher level of efficiency, Curvelet transform is usually implemented in the frequency domain. In the Fourier frequency domain both the Curvelet and the

image are transformed and then multiplied. Combination of the frequency response of Curvelet at different scales and orientations gives the frequency tilting that covers whole image in Fourier frequency domain as shown in the Figure 3.9. The product of multiplication is called a wedge. The product is then inverse Fourier transformed to obtain the Curvelet coefficient.

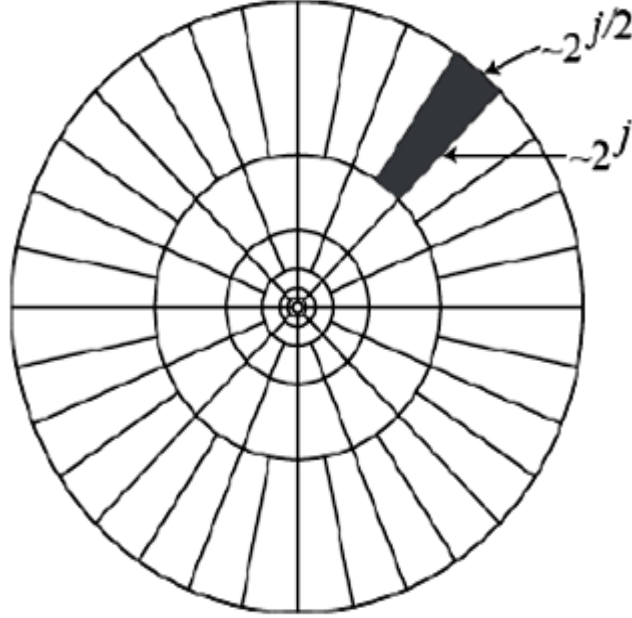


Figure 3.9 : Curvelet in the Fourier Frequency Domain [28]

For collecting Curvelet coefficients, inverse Fast Fourier transform is used. But, the trapezoidal wedge in the spectral domain is not suitable for use with the inverse Fourier transform. The wedge data cannot be accommodated directly into a rectangle of size $2^j \times 2^{j/2}$. To overcome this problem, Candes et al. have formulated a wedge wrapping procedure [71] where a parallelogram with sides 2^j and $2^{j/2}$ is chosen as a support to the wedge data as shown in the Figure 3.10. The wrapping is done by periodic tiling of the spectrum inside the wedge and then collecting the rectangular coefficient area in the center. The center rectangle of size $2^j \times 2^{j/2}$ collects all the information in that parallelogram. Discrete curvelet coefficients are obtained by applying 2D inverse Fourier transform to this wrapped wedge data. Wrapping based fast discrete curvelet transform is much more efficient and provides better transform results than ridgelet based curvelet transform.

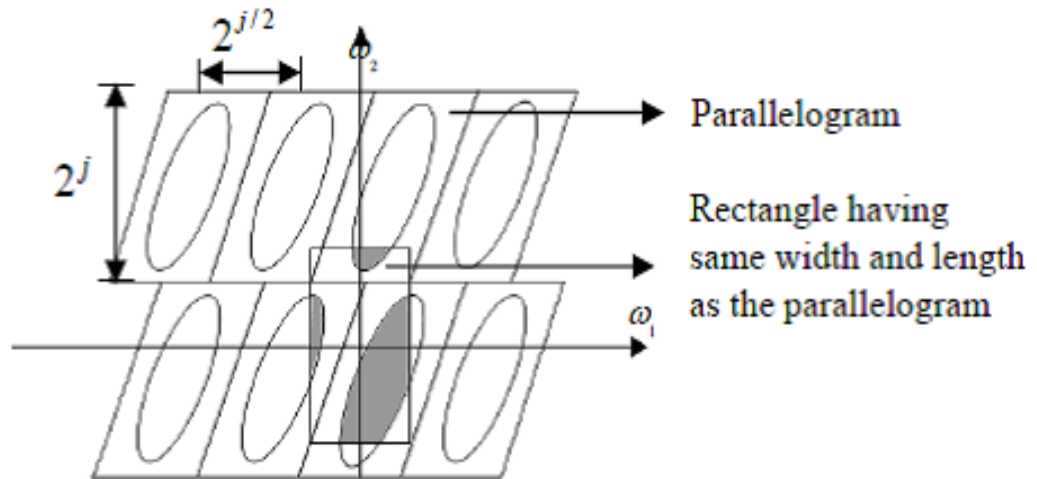


Figure 3.10 : Wrapping Wedge Around the Origin by Periodic Tilting of Wedge Data.

The Angle θ is in the Range $(\pi/4, 3\pi/4)$

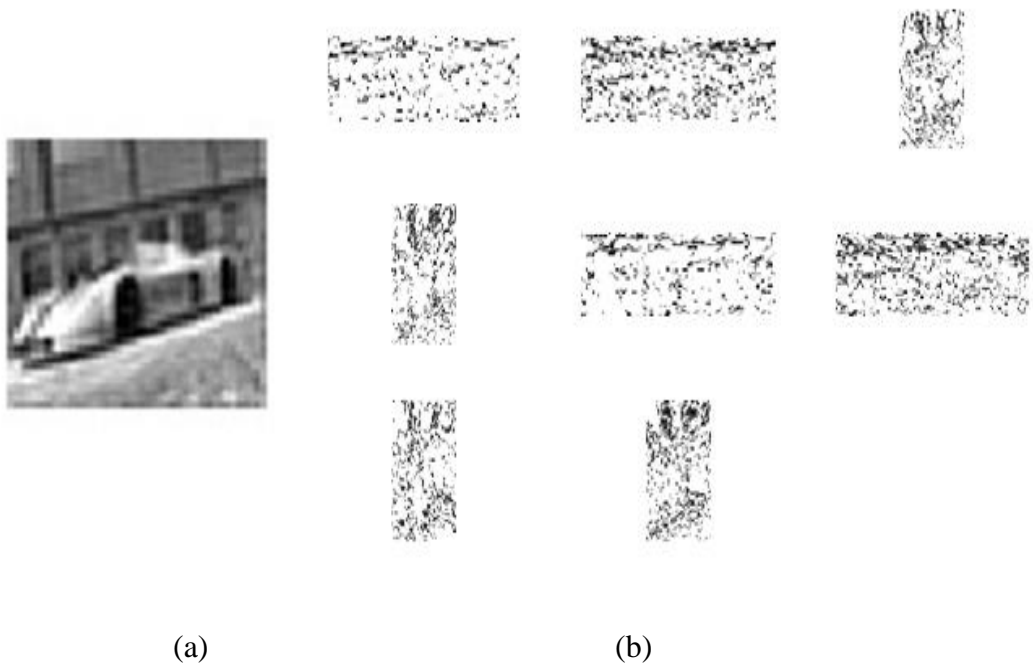


Figure 3.11 : Decomposition of Image using Curvelet Transform (a) First Level

(b) Second Level

Considering total numbers of coefficients generated in the different levels of the Curvelet Transform and execution speed for generation of the coefficients, lower coarsest level is considered in the proposed method as it takes less execution time and it gives minor difference in the recognition rate compared to other levels.

The Curvelet transform of 1 coarsest level and 8 angles are applied on the dataset images $f(m, n)$. In the proposed method, the images are decomposed into single scales using real-valued Curvelet. The coefficients obtained using the Curvelet transform are shown in the Figure 3.11. These resultant Curvelet Coefficients are used to reorder the column vector I_i of the images. The discrete Curvelet transform via wrapping with pre-processing has been implemented in the proposed algorithm. Image Vector X_i is constructed by converting coefficients to a column vector and then performing catenation of all coefficient vectors. Let $X = [X_1, X_2, X_3, \dots, X_p]$ be the Feature Image Matrix constructed by Discrete Curvelet Coefficient, then Eigenvalue and Eigenvectors are calculated for X .

3.1.3 Feature Selection

For selecting most efficient features, Eigenvalues are calculated using Principal Component Analysis (PCA). PCA is used with two main purposes. First, it reduces the dimensions of the data to a computationally feasible size. Second, it extracts the most representative features out of the input data so that although the size is reduced, the main features remain, and still be able to represent the original data [17].

Eigenvectors and Eigenvalues are calculated for the Principal Component Analysis. Eigenvectors are derived from the covariance matrix calculated from the Feature matrix. Eigenvectors are invariant to the direction. The covariance matrix C of the input data is calculated from the equation (3.13)

$$C = \frac{1}{P} \sum_{i=1}^P \phi_i \phi_i^T \quad (3.13)$$

Where the difference ϕ_i between image vector I_i and mean Ψ are calculated as equation (3.14) and (3.15)

$$\phi_i = I_i - \psi \quad (3.14)$$

$$\Psi = \frac{1}{P} \sum_{i=1}^P I_i \quad (3.15)$$

All Eigenvectors v_i and Eigenvalues λ_i of this covariance matrix are derived from the equation (3.16) as

$$\lambda_i = \frac{1}{P} \sum_{i=1}^P (v_i^T \phi_i)^2 \quad (3.16)$$

The set of Eigenvectors will have corresponding Eigenvalues associated with them; indicate the distribution of these Eigenvectors in representing the whole dataset. Typical references have shown that, only a small set of Eigenvectors with top Eigenvalues are enough to build up the whole image characteristic. PCA tends to find a P-dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. New basis vectors define a subspace of images called Eigenspace.

The value of Eigenspace is represented using equation (3.17).

$$\varepsilon = \sum_{i=1}^P v_i \quad (3.17)$$

The weight ω_i of each input image vector I_i is calculated from the matrix multiplication of the different Φ_i with the ε Eigenspace matrix.

$$\omega_i = \phi_i \times \varepsilon \quad (3.18)$$

The image weight calculated from the equation (3.18) is the projection of an image on the object Eigenspace, which indicates the relative “weight” of the certainty that whether such image is an image of a training Dataset or not.

The initial training set S consists of P different Images. These images are transformed into a new set of vector T^w of all input training weight. Figure 3.12 shows the EigenImage after applying PCA to the Curvelet transform with pre-processing and without pre-processing.

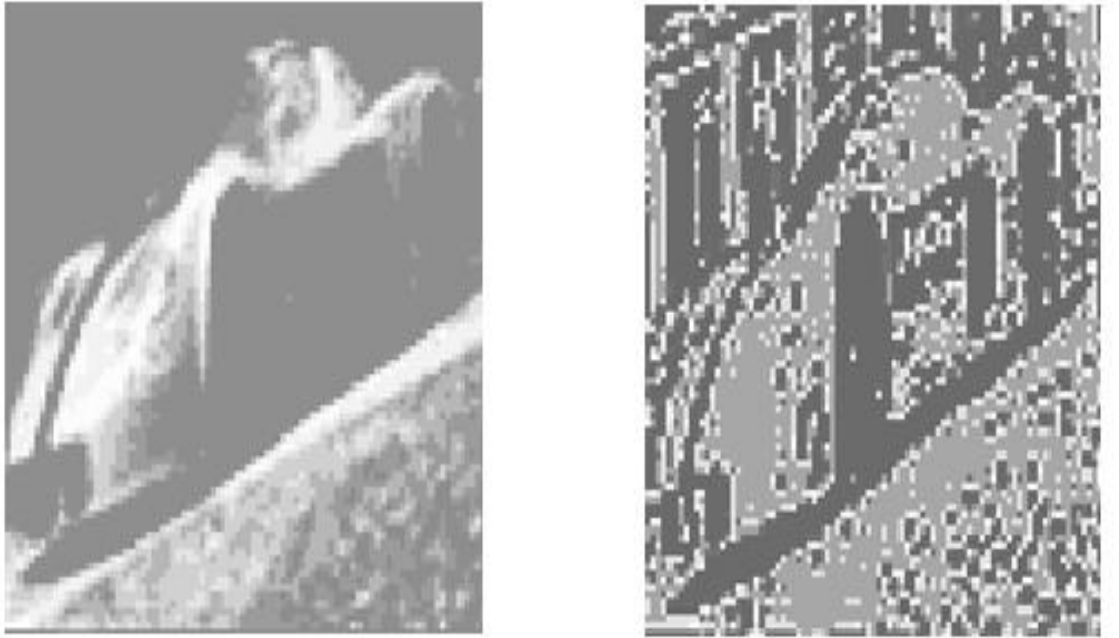


Figure 3.12 : (a) Eigenspace Image after applying Curvelet Transform without Pre-processing (b) Eigenspace Image after applying Pre-processing and Curvelet Transform

This transformation has showed how PCA has been used to reduce the original dimension of the dataset ($P \times m \times n$) to T^w (Size ($P \times P$)) where generally $P \ll m \times n$. Thus the dimensions are greatly reduced and the most representative features of the whole dataset still remain within P Eigen features only.

3.1.4 Feature Matching

For matching best feature vector from Eigen matrix, similarity measures like Euclidean distance measure and Neural network are calculated and compared.

3.1.4.1 Euclidean Distance Measure

With the coordinates (m, n) and (s, t) the Euclidean distance between coordinates p and q is defined as

$$De(p, q) = \sqrt{[(m - s)^2 + (n - t)^2]} \quad (3.19)$$

3.1.4.2 Backpropagation Neural Network

Backpropagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by the Application. As shown in Figure 3.13. Networks with biases, a sigmoid level, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Neuron Model (tansig, logsig, purelin) is an elementary neuron applied to the inputs. Each input is weighted with an appropriate weight matrix. The sum of the weighted inputs and the bias, form the input to the transfer function f . Neurons use any

differentiable transfer function f to generate the output. The Feedforward Neural network uses the Initialization, Activation, Weight Training and Iteration Steps to perform the Learning Phase. For training Neural Network, weight matrix T^w of Training Dataset obtained from the Contourlet-PCA/Curvelet-PCA is used as the input nodes of the Neural Network as shown in Figure 3.14.

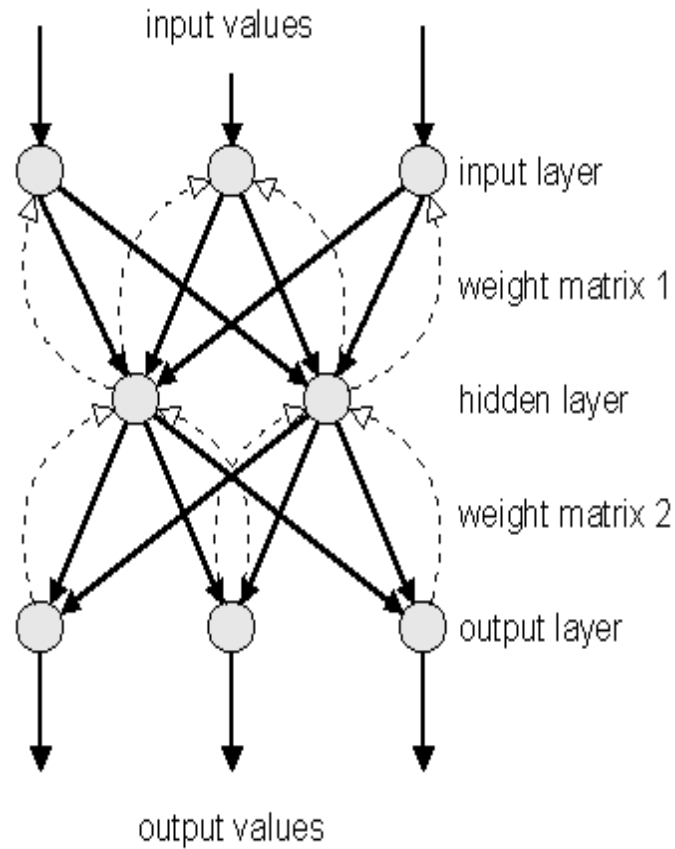


Figure 3.13 : Feed Forward Neural Network Model

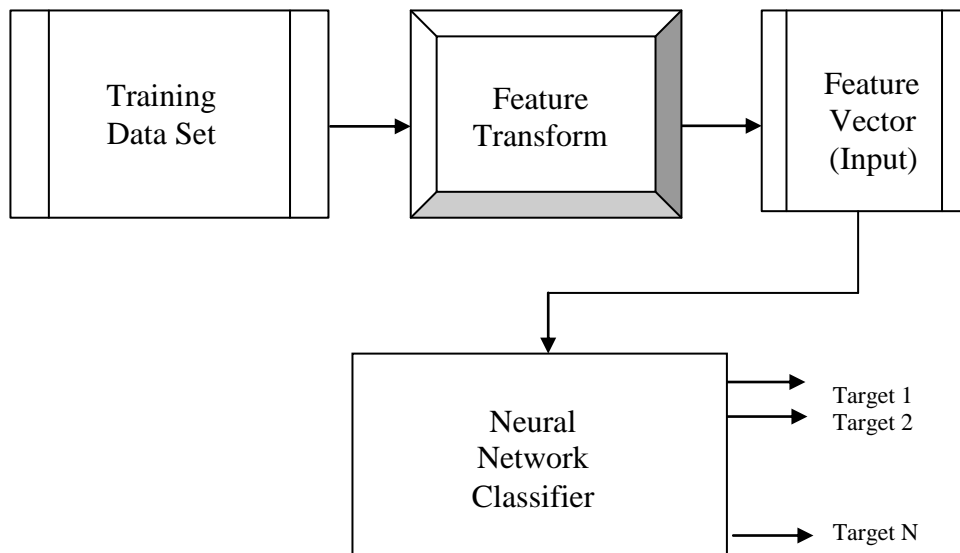


Figure 3.14 : Learning Phase of the Neural Network Classifier

When a new image from the test set is considered for recognition, the image is mapped to Contourlet-PCA / Curvelet-PCA subspace and weights are calculated for the particular image. The number of output nodes is equal to the number of total images, to be classified.

3.1.5 Proposed Methodology of Object Classifier

The objective of the proposed work is to extract the feature vectors for image Identification. Figure 3.15 illustrates overall process of calculating Contourlet transform / Curvelet transform and PCA applied to the training images and recognition of testing dataset. The first task for Feature extraction and selection and second task for Feature matching and object identification are executed as follows.

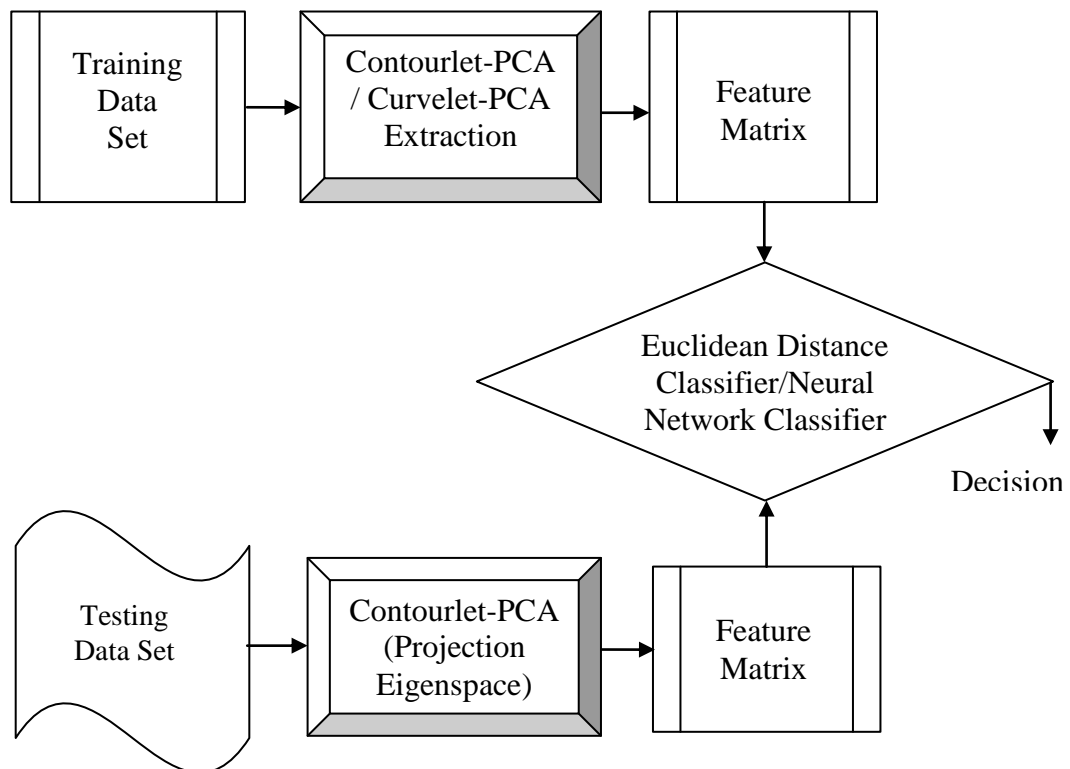


Figure 3.15 : Block Diagram of Proposed Object Classifier System

Let X_Image and Y_Image represent the training and testing datasets respectively. For gaining the best feature vector from the training dataset, at first, all the images are normalized.

Feature Extraction and Selection

The following steps are performed for feature extraction.

- RGB image is converted into grey scale image and resized to 64×64 .
- Filtering is applied to remove noise and sharpening the image. Unsharp Contrast Enhancement filter is used for the pre-processing of face images. Thresholding has been applied for retrieving edge points.
- Feature extraction is performed using Discrete Contourlet Transform and Discrete Curvelet transform.
- **Contourlet Transform:** Decompose each image into the Contourlet transform. As a result of performing Contourlet Transform, coefficients of low frequency and high frequency in different scales and various directions will be obtained. Decomposed coefficients with the same size $k \times k$ as $C_1, C_{2-1}, C_{2-2}, \dots, C_{n-1}, \dots, C_{n-u}$, where u is the number of directions. These Coefficients are used to reorder the column vector I_i of the images. All the coefficients are arranged to make a column vector.

Curvelet Transform: Decompose each image into the Curvelet transform. As a result of performing Curvelet Transform, coefficients of low frequency and high frequency in different scales and angles are obtained. Decomposed coefficients of different sizes are obtained as $C_1, C_{2-1}, C_{2-2}, \dots, C_{n-1}, \dots, C_{n-v}$ where v is the number of angles. These Coefficients are used to reorder the column vector I_i of the images. All the coefficients are arranged to make a column vector.

- The Feature image matrix $I = [I_1, I_2, I_3, \dots, I_P]$ is constructed from the coefficient column vector I_i , where i represent the number of images.
- Feature matrix I is transformed to lower dimension subspace T^w using PCA.
- T^w consists of Weight calculated for each image of the respective Dataset.
- Neural Network / Euclidean Classifier are used to measure the distance between the images.

Feature Matching and Identification

Feature Matching is performed by Euclidean distance classifier and neural network classifier for Object identification. The results of both methods compared and implemented in the visual tracking task.

A. Euclidean distance Classifier

In this classification method, each image transformed to a lower order subspace using Contourlet-PCA / Curvelet –PCA using the above steps. Upon observing an unknown test image X , the weights are calculated for that particular image and stored in the vector w_x . Afterwards, w_x is compared with the weights of training set T^w using the Euclidian distance using equation (3.20).

$$De(p, q) = \sqrt[2]{[T^w - w_x]} \quad (3.20)$$

If average distance does not exceed some threshold value, the weight vector of the unknown image w_x is matched with the training dataset.

B. Neural Network Classifier

Weight vector is used to feed the respective neural network for obtaining the object recognition results. A threshold value near to 1 represents the classification matching

to the target and 0 represents the classification far away from the target. Logarithmic sigmoid transfer function is used for input layer and hidden layer. Back propagation training is implemented with Gradient descent with momentum.

3.2 Vehicle Identification System

Considering Visual Surveillance system and Traffic monitoring system, Vehicle Identification system has been implemented. The novel approach using three Class structures has been proposed to improve the efficiency of vehicle identification. Three classes have been classified according to the length and width ratio. For example as classification of vehicle has been categorized in bus, truck, cycle, scooter, rickshaw etc. If bus or truck is considered from side view, the length to width ratio becomes less than one. These types of vehicles which are rectangular in shapes are considered as class one. Class one vehicles are stored separately as a training set one. The second class of vehicles is considered having length to width ratio greater than one. Pedestrians are considered in this class. Vehicles having equal length and width where ratio is close to one is considered to be class 3. Different feature vectors are calculated for each class. The flow chart for training the feature matrix and Vehicle identification tasks are explained in Figures 3.16 and 3.17.

3.3 Visual Tracking

3.3.1 Single Object Tracking Algorithm

To overcome the problems arising in conventional tracking algorithms which were discussed in the literature review earlier, a novel Block Matching Tracking Algorithm using Predictive Motion Vector based on 3D color histogram has been proposed and implemented efficiently.

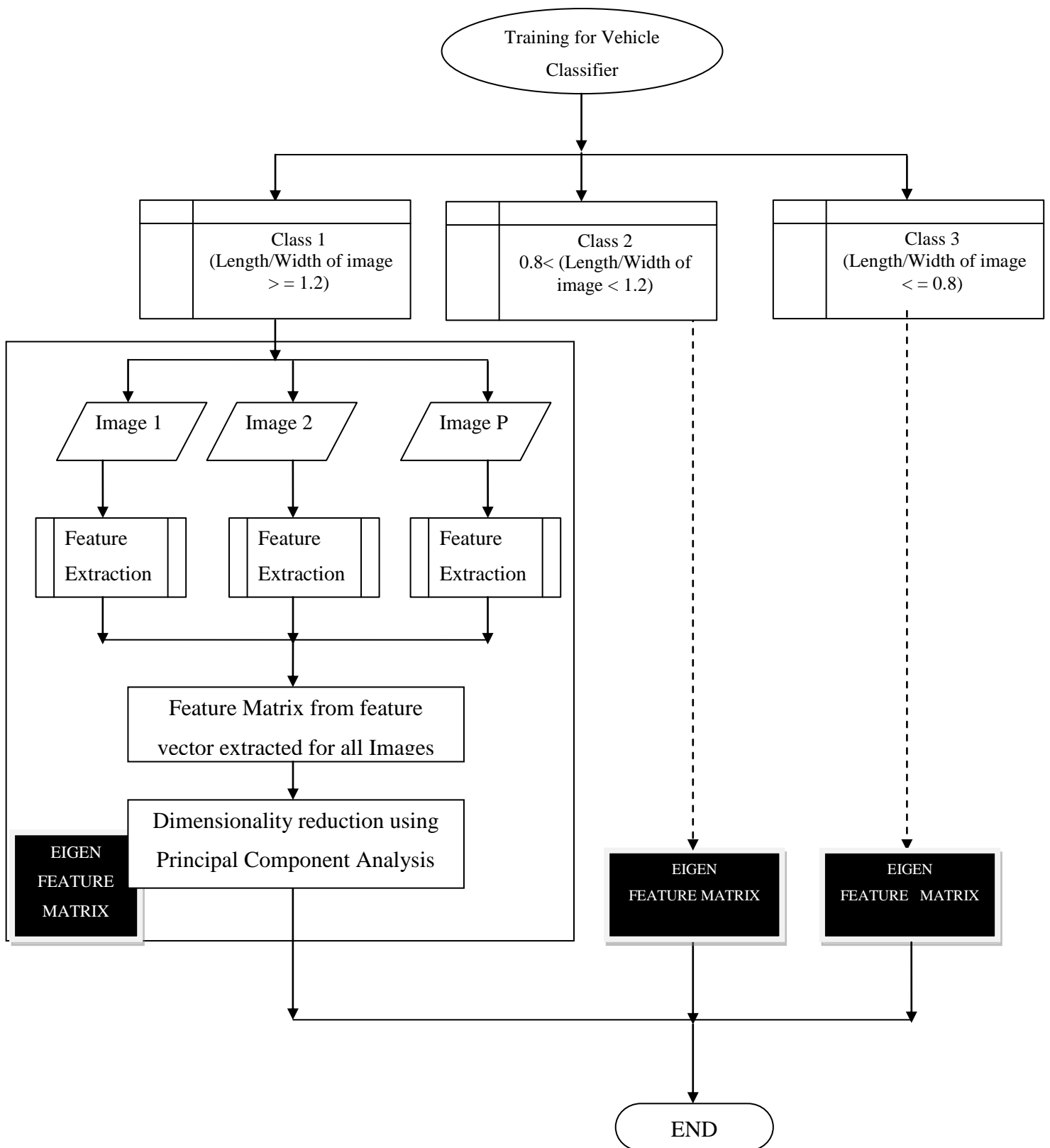


Figure 3.16 : Training of Vehicle Dataset using Three Class Structures

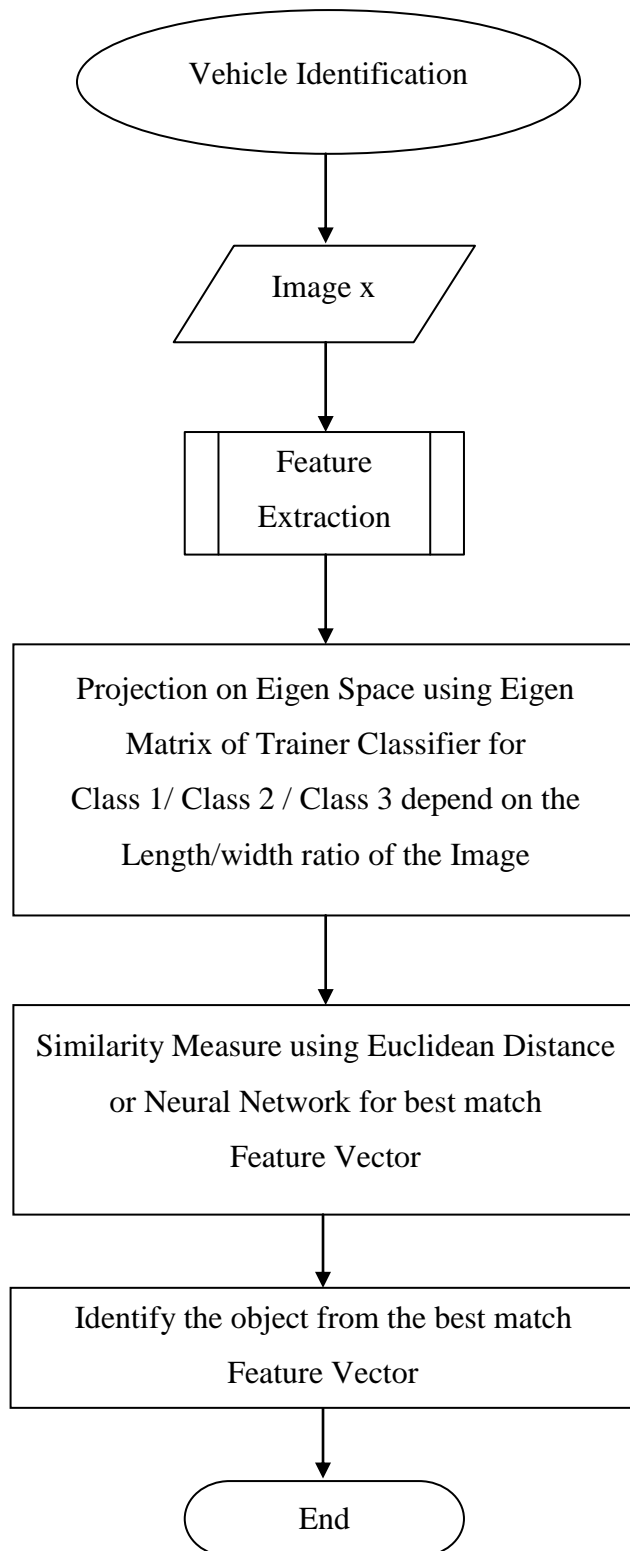


Figure 3.17 : Vehicle Identification System

To minimize the searching time of the object block in the frame, prediction based probabilistic search block matching algorithm has been implemented. The proposed algorithm is considered with a flexible size of block as well as pixel displacement. System tracks the single object selected by the user. Accuracy increased by implementing different conditions of the object like-object having similar type of background and foreground, object moving near to frame boundary, object with no motion in the frame sequence, object of boundary etc.

3.3.1.1 Object Tracking Algorithm

The main algorithmic flow chart as shown in the Figure 3.18 can be summarized as follows:

1. Define a rectangular block on the region of interest B_o in the first frame of a video sequence.
2. Compute the 3D colour histogram $h1$ ($m \times m \times m$) of the B_o region. Here $8 \times 8 \times 8$ bins have been used to find colour histogram.
3. In the second frame, start from the former location and examine the surrounding windows by calculating histogram $h2$ ($m \times m \times m$) for each window B_j having block sizes same as B_o . Similarity measure by Histogram matching using Bhattacharya coefficient is applied across the Frame using equation (3.21).

$$\rho[p, q] = \sum_{M=1}^m \sqrt{p^{(u)} q^{(u)}} \quad (3.21)$$

Where $p^{(u)}$ and $q^{(u)}$ are the histogram of two different images and ρ is the similarity measure. The large the value of ρ , more will be the similarities between the distributions. For two identical normalized histograms we obtain $\rho=1$, indicating a perfect match.

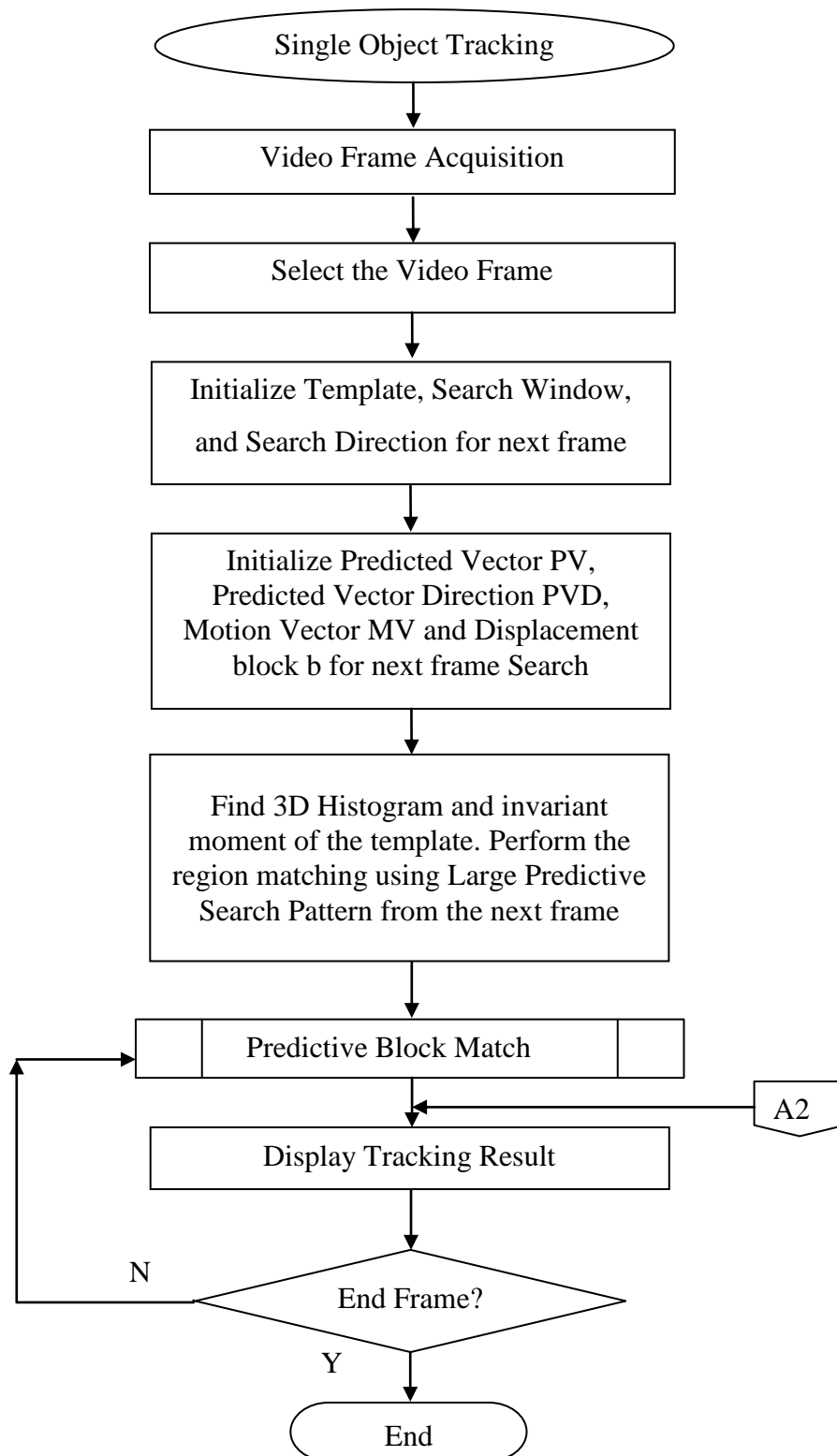


Figure 3.18 : Single Object Tracking Algorithm

4. Apply Predictive Block Matching (PBM) algorithm to find the search region and find appropriate similarity region while minimizing the distance between the detected locations using the matching criteria. The flow chart of PBM algorithm is explained in the Figure 3.20.
5. Iterate the above steps for all the frames in a sequence.

3.3.1.2 Predictive Block Matching Algorithm

Search Pattern

The proposed PBM algorithm employs pattern as illustrated in the Figure 3.19. The pattern called Large Predictive Search comprises of nine checking points from which eight points surround the centre one.

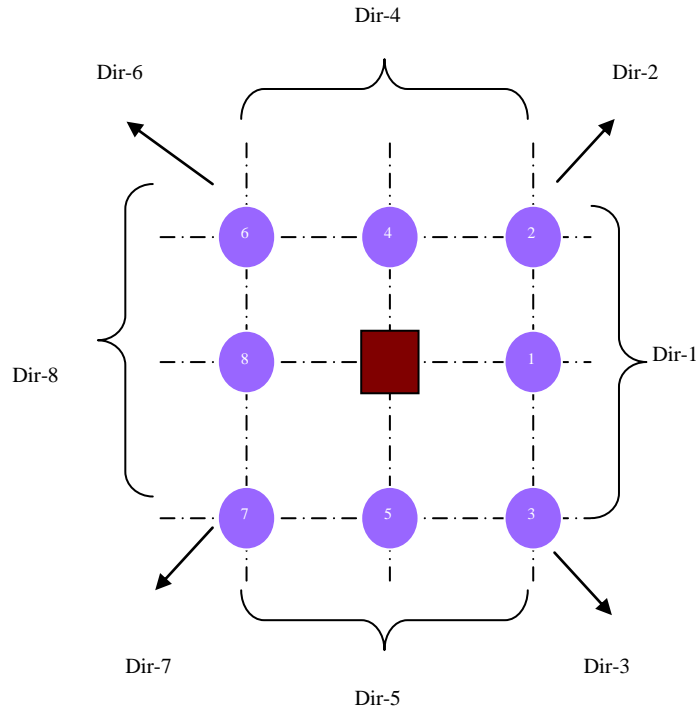


Figure 3.19 : Large Predictive Search Pattern – Nine points

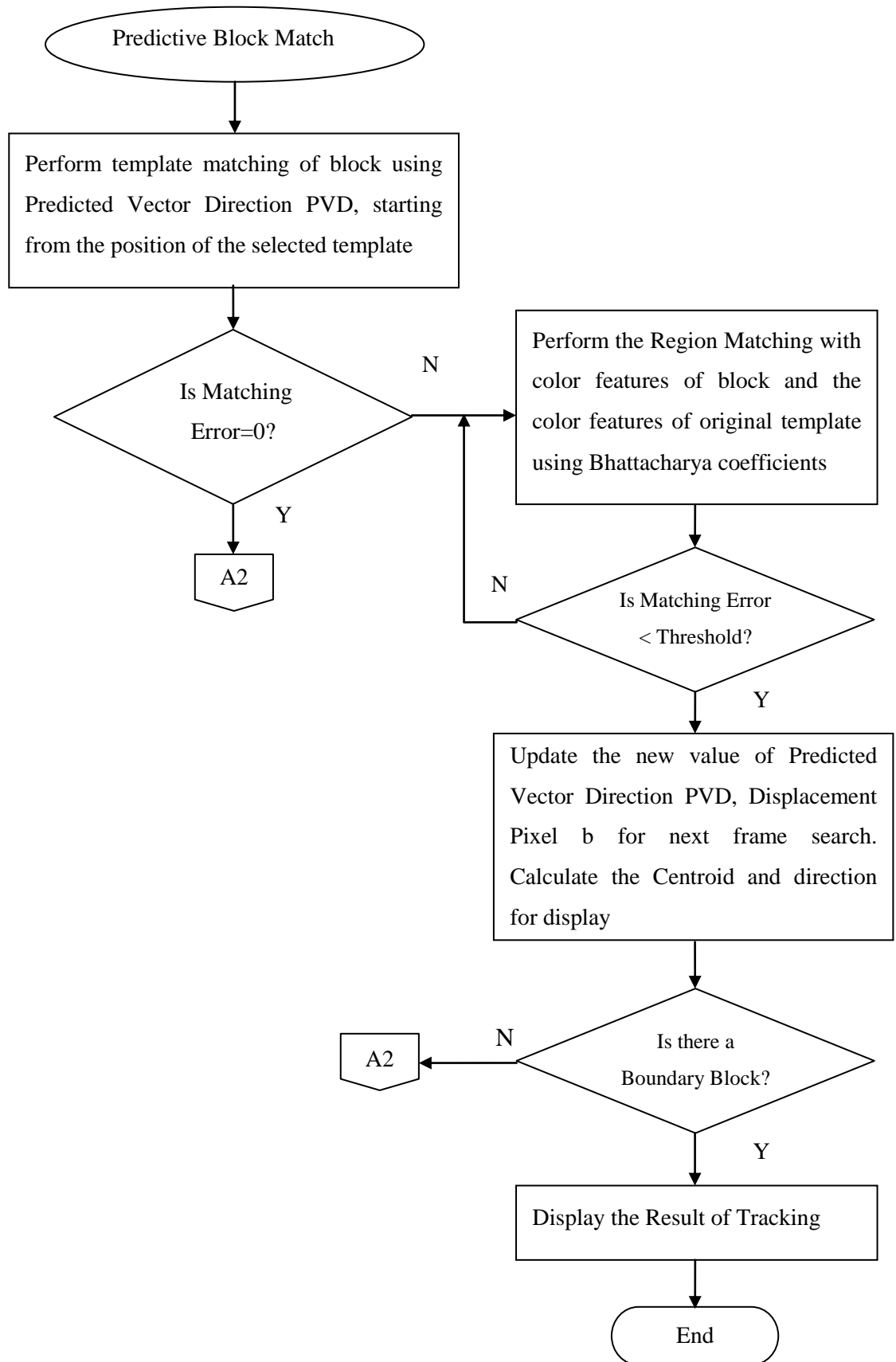


Figure 3.20 : Predictive Block Matching Algorithm

Block Matching Algorithm

The steps in the Block matching algorithm can be summarised as follows $n_1 \times m_1$ is the Bounding box size of the object, that is considered as a Block Size and the left corner pixel of the object is $B_o(i, j)$.

- Initialize the Predicted Vector (PV) and displacement value of pixel
- Initialize the Predicted Vector direction of search for the first frame as shown in the Figure 3.21.

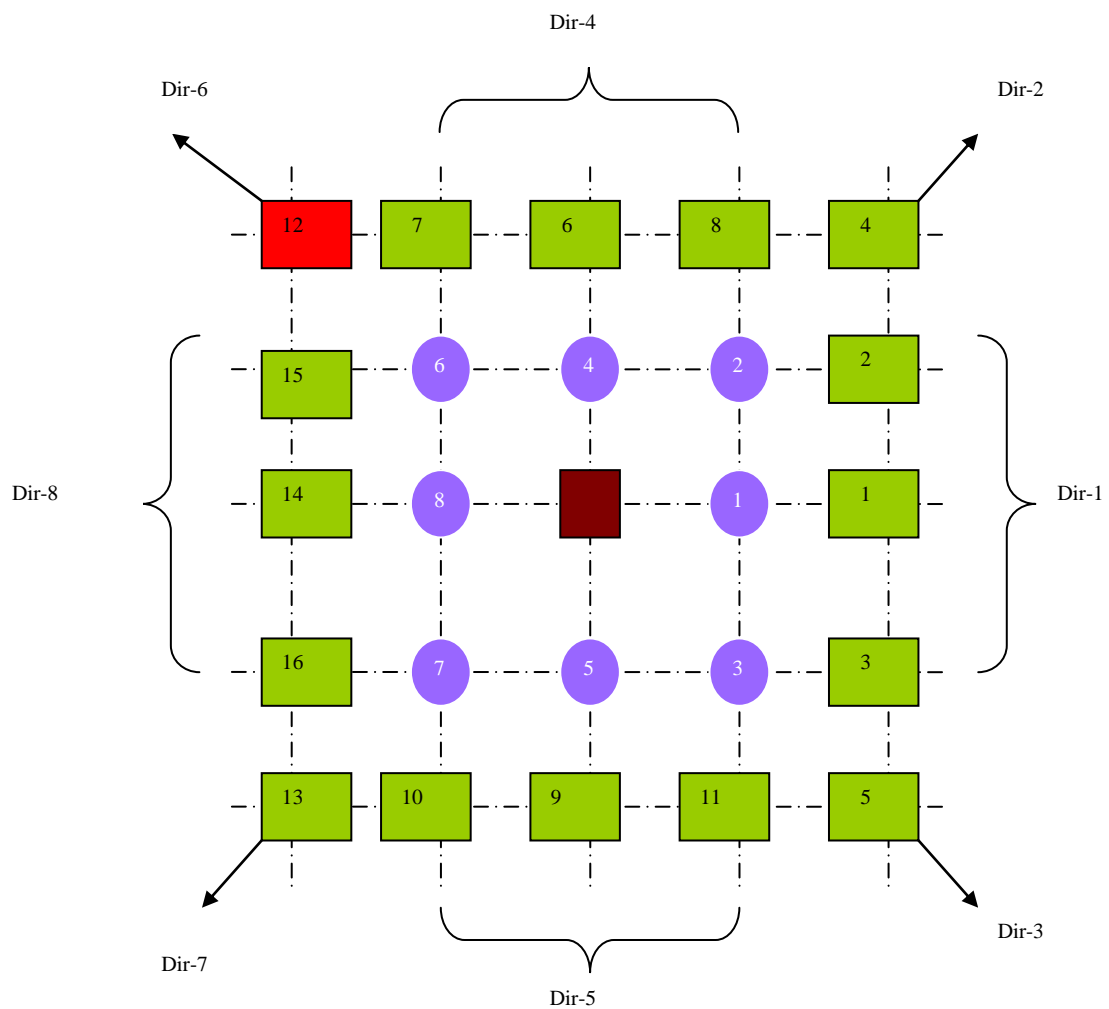


Figure 3.21 : Block Matching Direction and Search for Frame no. 1.

Best Block Match found in Direction 6

- b. Compute the matching error (E) between the current block and block which appeared at the same location of the object in the reference frame.

If matching error (E) = 0

Motion vector (MV) \leftarrow 0

Exit and go to step 4;

else

Go to step 3.

- c. Check nine search points of the Search window according to the direction set using predicted vector directions. Repeat the search as shown in the Figure 3.22 in all 8 directions until matching block is found.

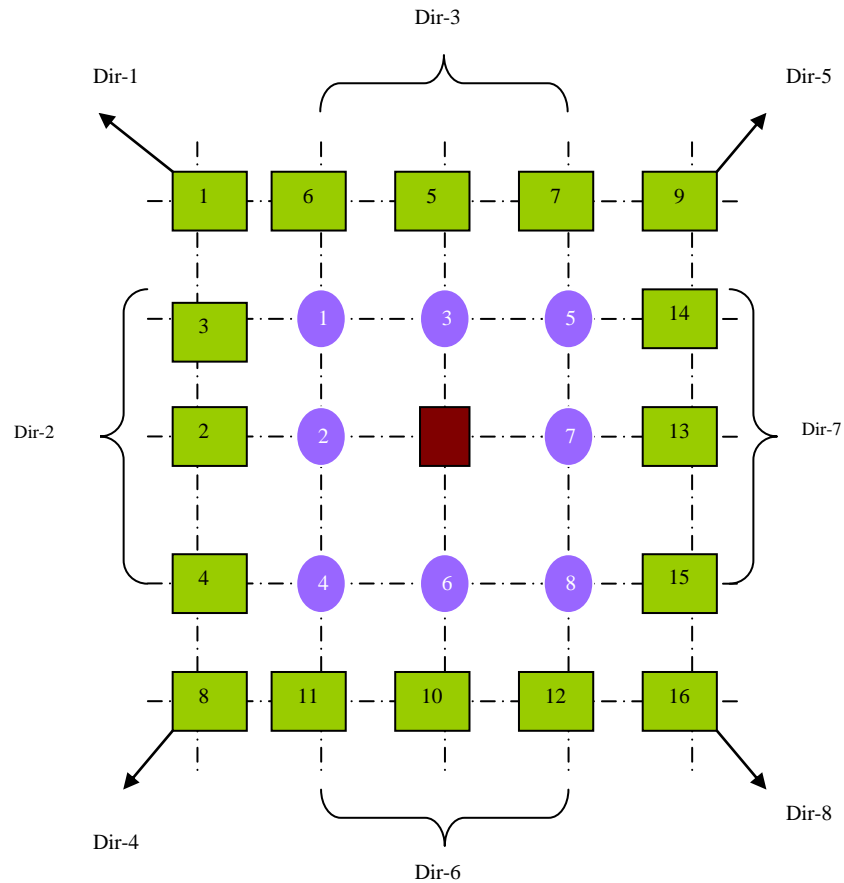


Figure 3.22 : Block Matching and Search Predicted Vector Direction for Predictor
Vector found in the Direction 6

If matching Criteria > Threshold Value

Motion Vector (MV) \leftarrow Block pixel value $B_j(x,y)$

$B_{11} \leftarrow$ x Coordinate of Match block – x Coordinate of previous Match Block

$B_{12} \leftarrow$ y Coordinate of Match block – y Coordinate of previous Match Block

Predicted Vector PV \leftarrow i

Displacement pixel b \leftarrow max (B11, B12)/2

Next direction nd \leftarrow PV

Predicted Vector Direction PVD \leftarrow [nd, nd+1, nd-1, nd+2, nd-2, nd+3, nd-3, n+4]

If Match Block = Boundary Block

Exit from the main program.

else go to step3

else go to step 2.

3.3.1.3 Analysis of PBM algorithm

Zero motion prejudgment

To distinguish the static and moving blocks, a technique called Zero motion prejudgment [45] is implemented. The prejudgment is made by computing the matching error between the current block & block at the same location in the reference frame that corresponds to zero motion vector. If matching error is zero, than object is considered as a static object and algorithm jumps to the next frame without performing the remaining search.

Early termination

If the object in the frame found moving towards the boundary, the early termination of the object is considered by minimum matching difference at all the boundary points. If matching found, search process will be immediately terminated without checking next frames.

3.3.2 Multiple Objects Tracking

For multiple objects tracking, Kalman Filter tracking works well, only when an accurate model of the problem is available. Particle Filter Tracking supports multi-object tracking without requiring any modeling of the object but at the cost of higher computational speed. Mean Shift tracking is fast but not robust for extremely fast moving object and illumination changes. To overcome the above problem, Blob Tracking algorithm is used as tracking that established by temporal relationships between Blobs without the use of domain-specific information. For further improvement in the conventional Blob tracking, color segmentation was applied to retrieve color statistics of the object. To overcome the problem of same color object, features were extracted using Contourlet transform. Hybrid tracker implementation is the suggested efficient algorithm for visual tracking. The hybrid tracker has been implemented using color features and discrete Contourlet Transform.

A distinctive feature of the proposed algorithm is that the method operates on region descriptors instead of region themselves. This means that instead of projecting the entire region into the next frame, only region descriptors need to be processed. Therefore, there is no need for computationally expensive models.

To show the validity of the algorithm, traffic monitoring system is considered at present. Different statistical conditions are incorporated for making efficient algorithm. Vehicle classifier is also incorporated with the visual tracking task which identifies and displays the class of vehicle indicating car; bus etc in the visualization of tracking. The Contourlet transform feature extraction used in the classifier and also

used to track the object of the same color. Thus Algorithm calculations serve multipurpose and also increase the efficiency.

The main algorithm for visual tracking is summarized as follows:

- Perform video pre-processing on each frame.
- Do Video Segmentation using background subtraction.
- Perform Thresholding to convert the image into binary image.
- Apply Opening and Closing function to remove unwanted small data.
- Calculate the object statistics using Blob Analysis system object.
- Do the Color Segmentation using the object statistics derived from above steps.
- Track the object based on the color histogram and set of 2D moment Invariants. Same color featured object is to be differentiated using Contourlet transform and PCA. Centroid Statistics is to be used to measure the distance and direction of the object with respect to the previous frame.
- Visualize the result in the movie frame.

The overall system flow chart for multiple objects tracking has been explained in the Figure 3.23 and Figure 3.24. Figure 3.24 explains the flow chart for visual object tracker task which has been called in the main algorithm of visual tracking explained in the Figure 3.23.

Brief description about each step implemented in the algorithms is given in the sub sections.

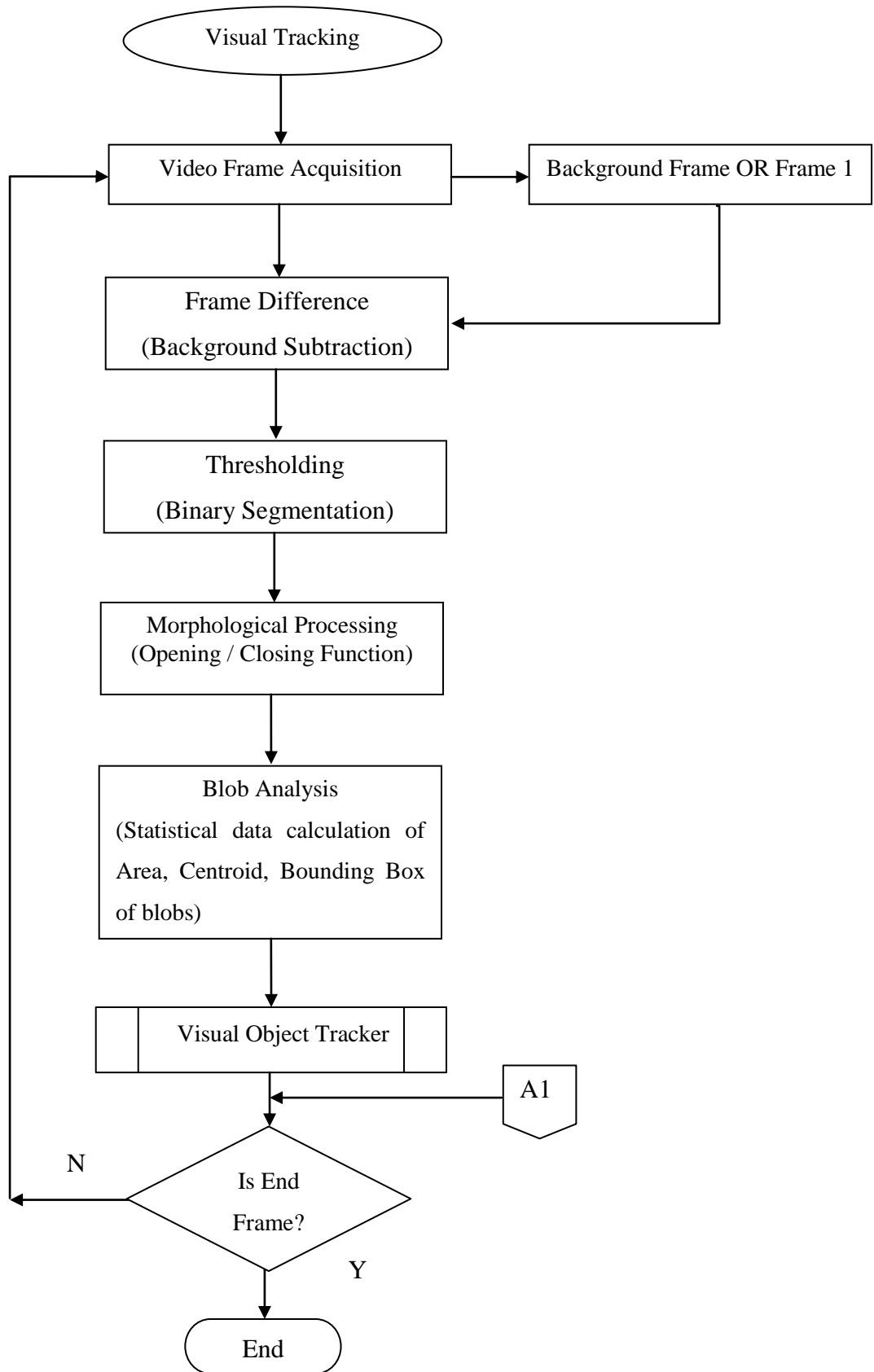


Figure 3.23 : Visual Tracking Algorithm for Multiple Objects

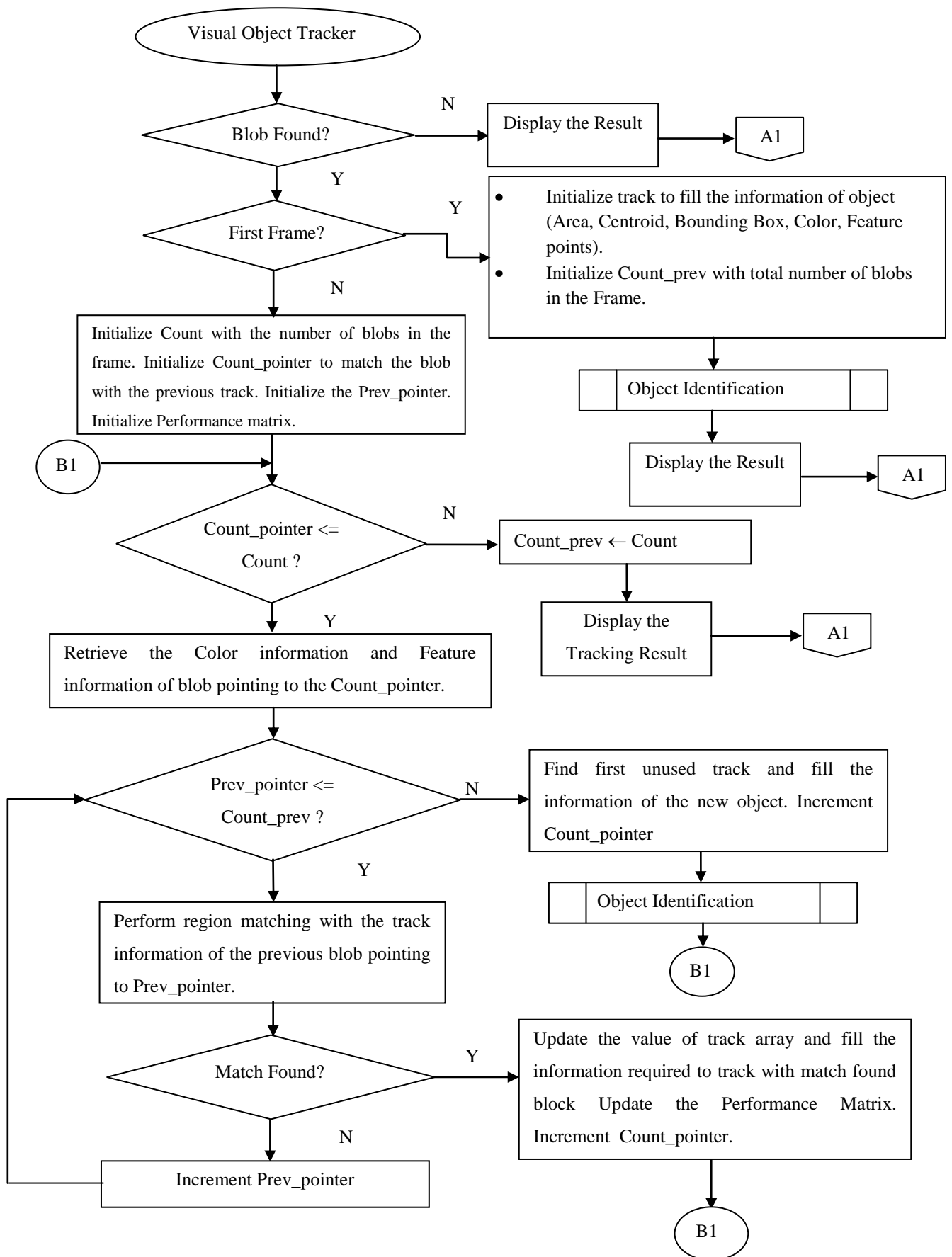


Figure 3.24 : Visual Object Tracker for performing Region Matching and Tracking

3.3.2.1 Foreground Object Extraction

For extraction of the foreground object; background subtraction, thresholding and morphological operations are performed on video frames.

➤ Background Subtraction and Thresholding

Segmentation approach to extract the foreground and background scene must be robust to shadow and changing light conditions. The method should be sensitive enough to detect actual changes in the background scene while not identifying false variations. Also the speed of execution must be at high rate which can be implemented for real time processing. To eliminate the effect of the shadow and lighting effect, all RGB frame are converted to YC_bC_r color space which is widely used for video processing [76]. In this format, luminance information is stored as a single component (y), and chrominance information is stored as two color-difference components (c_b and c_r). C_b represents the difference between the blue component and a reference value. C_r represents the difference between the red component and a reference value. They are expressed by the equations as

$$y = 0.299 R_0 + 0.587 G_0 + 0.114 B_0 \quad (3.22)$$

$$c_b = -0.169 R_0 - 0.331 G_0 + 0.500 B_0 \quad (3.23)$$

$$c_r = 0.500 R_0 - 0.419 G_0 - 0.081 B_0 \quad (3.24)$$

Luminance information that is affected by the shadow and lighting conditions in the background is removed by equation (3.25) and (3.26). Foreground object is detected by the difference between current frame and image of the static background of scene which is normally considered as a first frame of the sequence or selected by the user as a background frame.

$$|Frame_i(y) - background(y)| > Th_1 \quad (3.25)$$

$$|Frame_i(c_b c_r) - background(c_b c_r)| > Th_2 \quad (3.26)$$

Where Th_1 and Th_2 are threshold values used for detection of foreground objects calculated with Otsu's threshold method.

In the proposed system, static background scenes are assumed, so a general threshold value is chosen that applies to all pixels. In addition, a number of post processing stages are used to clean up the resultant image. Images produced by background subtraction techniques have a lot of noise due to threshold selection. The second step of this algorithm is to make a morphological opening and closing in the binary image (the segmented image). The morphological operations remove small objects created by noise.

➤ **Morphological operation**

The morphological open and close operations, using circular structuring elements of radius 5 pixels, are applied to assist in the noise removal process. The morphological opening is composed of two basic operators with the same structure element Erosion and Dilation. Erosion and Dilation can be expressed in terms of Minkowski addition and subtraction of two set A and B as [31]:

$$D(A, B) = A \oplus B = A + B = \cup_{b \in B} (A + B) \quad (3.27)$$

$$E(A, B) = A \ominus B = A - (-B) = \cap_{b \in B} (A - B) \quad (3.28)$$

The mathematical opening and closing can be represented as

$$O(A, B) = A \circ B = (A \ominus B) \oplus B \quad (3.29)$$

$$C(A, B) = A \bullet B = (A \oplus B) \ominus B \quad (3.30)$$

Where A is an input image and B is a structuring element.

3.2.2.2 Blob Analysis

Each object is processed separately and decomposed into a set of non-overlapping regions to produce the region partition. This step takes into account the spatio-temporal properties of the pixels in the computed object partition and extracts homogeneous regions. After performing morphological operations, blob analysis can be done to identify the objects in the scene and calculate their features to track the objects in the binary image frame. Typically the blob features usually calculated are area, number of pixels which compose the blob, perimeter, location and blob shape. Color segmentation is performed by extracting the color descriptor from the original frame using the blob statistics. Two different ways of connection can be defined in the blob analysis algorithm depending on the application. One consists to take the adjacent pixels along the vertical and the horizontal as touching pixels and the other by including diagonally adjacent pixels as shown in the Figure 3.25.

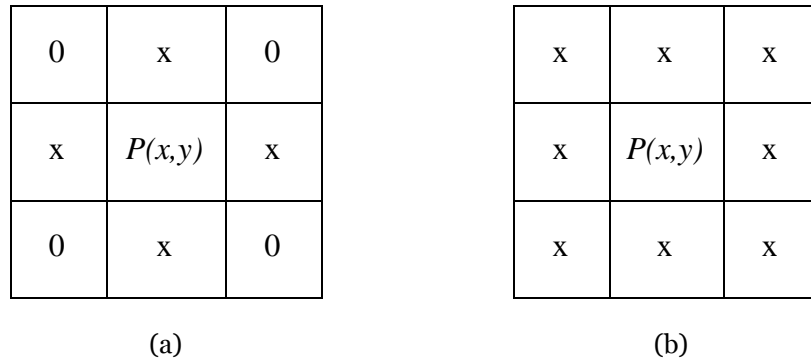


Figure 3.25 : (a) 4-Neighbourhood Pixels (b) 8-Neighbourhood Pixels

Figure 3.26 describes the overall steps for performing the blob segmentation.

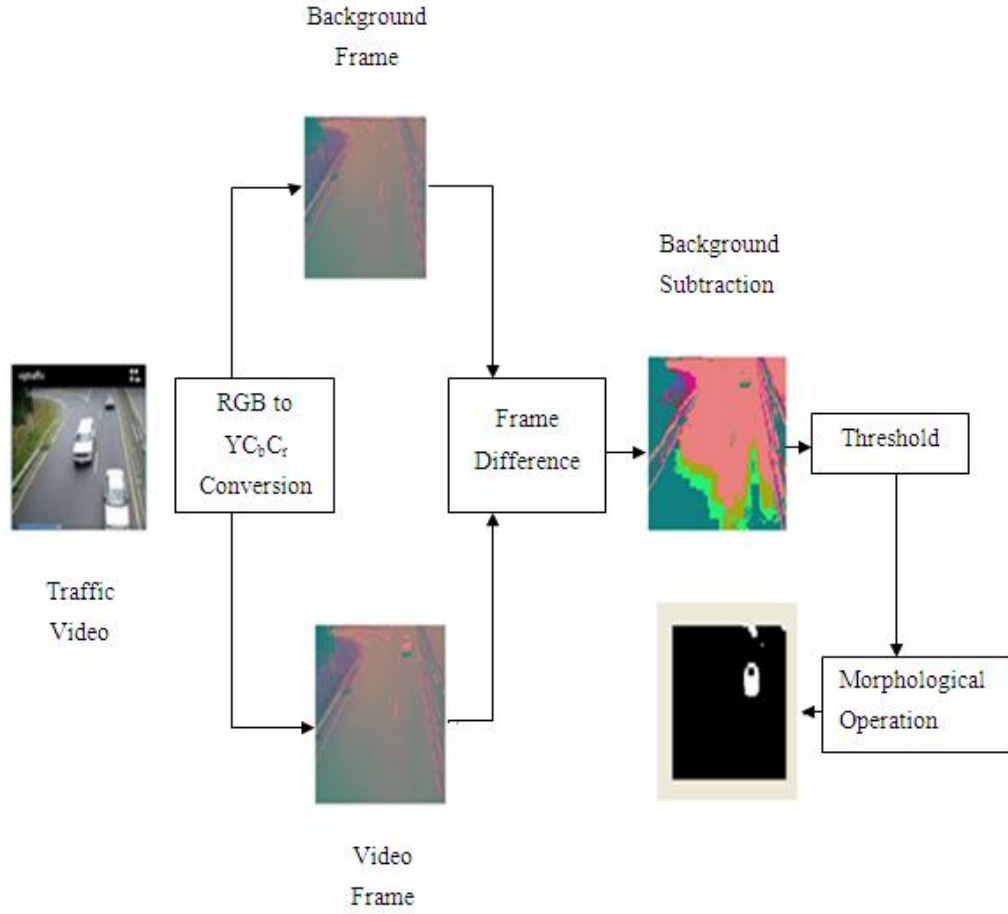


Figure 3.26 : Blob Segmentation Module

Blob statistics

For target instance called blob having the area a_t^l for l number of objects at time t for frame number n includes its Area $A(a_t^l)$. Its Centroid $c(a_t^l)$ is computed as:

$$c(a_t^l) = \frac{1}{A(a_t^l)} \sum_{i=0}^{A(a_t^l)-1} p_i \quad (3.31)$$

Where p_i are the blob pixels

The similarity s and the Centroid distance $D(a_t^l, b_{t-\Delta t}^l)$ between two target instances a_t and $b_{t-\Delta t}$ in two consecutive time slices t and $t-\Delta t$ are defined as:

$$s(a_t^l, b_{t-\Delta t}^l) = \left| \frac{A(a_t^l) - A(b_{t-\Delta t}^l)}{A(a_t^l) + A(b_{t-\Delta t}^l)} \right| \quad (3.32)$$

$$D(a_t^l, b_{t-\Delta t}^l) = \sqrt{\left(c(a_t^l) - c(b_{t-\Delta t}^l) \right)^2} \quad (3.33)$$

3.2.2.3 Region Descriptor and Region Matching

The performance of the blob analysis algorithm depends totally on the quality of the segmentation. With a bad segmentation, the blob analysis can detect some not interesting blobs or can merge some different blobs due to lighting condition or noise in the image. Blob analysis finds the entire bounding boxes which are created and puts a label for each one to memorize the different regions present in the scene. For each region a set of features are extracted as region descriptors. The feature space used is composed of spatial and temporal features. Color descriptor is derived for each region using blob statistics from the current frame. 3- D histogram is calculated for each region. Color descriptor is converted into seven invariant moments for dimensionality reduction.

Hu (1962) derived a set of two dimensional invariant moments in shape recognition using algebraic invariant. Two dimensional moments of a digitally sampled $M \times M$ image having gray function $f(x, y)$ for $(x, y=0, \dots, M-1)$ can be expressed by

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x)^p \cdot (y)^q f(x, y) \quad (3.34)$$

Where $p, q = 0, 1, 2, 3, \dots$

The moments $f(x, y)$ translated by an amount (a, b) are defined as

$$\mu_{pq} = \sum_x \sum_y (x + a)^p \cdot (y + b)^q f(x, y) \quad (3.35)$$

The central moments m'_{pq} or μ_{pq} can be computed using equation (3.34) and (3.35) by substituting $a = -\bar{x}$ and $b = -\bar{y}$ as

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{01}}{m_{00}} \quad (3.36)$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p \cdot (y - \bar{y})^q f(x, y) \quad (3.37)$$

The shape of an image can be represented in terms of seven invariant moments ($\phi_1 - \phi_7$) expressed by equation (3.38) to (3.44). The first six functions ($\phi_1 - \phi_6$) are invariant under rotation and last ϕ_7 is both skew and rotation invariant.

The seven moments can be defined as

$$\phi_1 = \eta_{20} + \eta_{02} \quad (3.38)$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \quad (3.39)$$

$$\phi_3 = (\eta_{30} - 3\eta_{31})^2 + (3\eta_{21} - \eta_{03})^2 \quad (3.40)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (3.41)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{31})(\eta_{30} \eta_{12}) \cdot [3(\eta_{30} + \eta_{12})^2 \\ & - (\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) (3\eta_{21} \\ & - \eta_{03}) \cdot [3(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] \end{aligned} \quad (3.42)$$

$$\begin{aligned} \phi_6 = & (\eta_{20} - \eta_{02}) [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12}) (\eta_{21} + \eta_{03}) \end{aligned} \quad (3.43)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03}) (\eta_{30} \\ & + \eta_{12}) \cdot [(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] (\eta_{21} \\ & + \eta_{03}) \cdot [3(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] \end{aligned} \quad (3.44)$$

Where the normalized central moments are denoted as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (3.45)$$

Where $\gamma = \frac{p+q}{2} + 1$

The ϕ values make a seven entries feature vector that is used for measuring the similarity between images.

Region Matching is the projection of the information of the current frame n into the next frame $n+1$. Regions of frame n and frame $n+1$ with most similarity are

considered as the correspondent objects and receive same labels. Seven 2D invariant moment values are used for region matching. Two way matching has been performed for accurate matching purpose. Color histograms with 2D invariant moments have been used for region matching. Color histogram handles shape variation property of the object well. If more objects having similar color features in the frame exists, than it gives more than one correspondence between the target frames. For eliminating the problem, frequency feature transform is used. Contourlet transform with Principal component analysis handles the scale deformation of the object well.

3.2.2.4 Region Tracking

After finding the corresponding regions between two successive frames, through a top-down and a bottom-up interaction with the region partition step, objects of current frame are validated and are given same labels as previous frame. Motion analysis is performed to find the direction of motion and speed of the motion using the blob statistics as shown in the Figure 3.27.

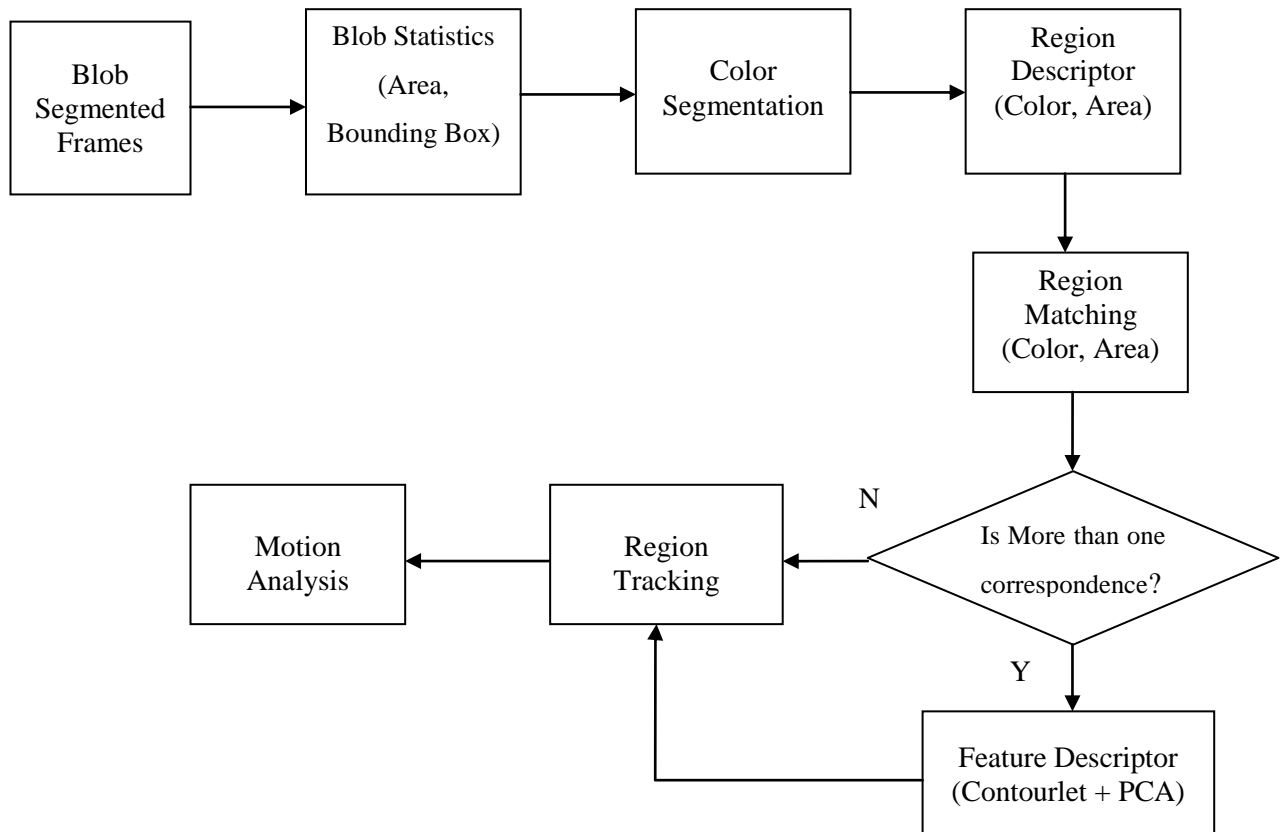


Figure 3.27 : Visual Tracking System

Summary: Object Identification and Visual Tracking algorithm for single object tracking and multiple objects tracking are discussed. For object identification, classifier is designed with the discrete Contourlet transform and fast discrete Curvelet transform via wrapping. For efficient design of Classifier, feature extraction is performed by discrete Contourlet transform and fast discrete Curvelet transform via wrapping followed by pre-processing stages used for image enhancement. For fast execution speed of feature extraction, Eigenvalues are calculated which are discriminant features of the image plays important role in the dimensionality reduction of feature matrix created for training dataset. Efficiency of the classifier is compared with the discrete Contourlet transform and discrete Curvelet transform with and without pre processing. Considering Visual Surveillance applications, Vehicle Classifier is designed with the 3-class structure for more improvement in the object identification task. Finally, discrete Contourlet transform with the pre processing is incorporated for object identification with the visual tracking task as it is more efficient and fast compared to other methods.

For single Visual tracking, novel block matching algorithm has been proposed with efficient termination conditions while tracking. Multiple objects tracking algorithm has been implemented using hybrid tracker. Hybrid tracker is more efficient for visual surveillance applications. Hybrid tracker is implemented with color features and texture features using discrete Contourlet transform, that are calculated for object identification purpose also. Visual tracking task calculates speed in terms of pixels, direction and object tracking number with object identification.

For actual speed conversion from image space to object space camera parameters are calculated. The camera parameters calculations with the experimental results of object identifications and visual tracking have been discussed in the next chapter.

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 Dataset⁷⁷

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 Dataset⁷⁸

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)

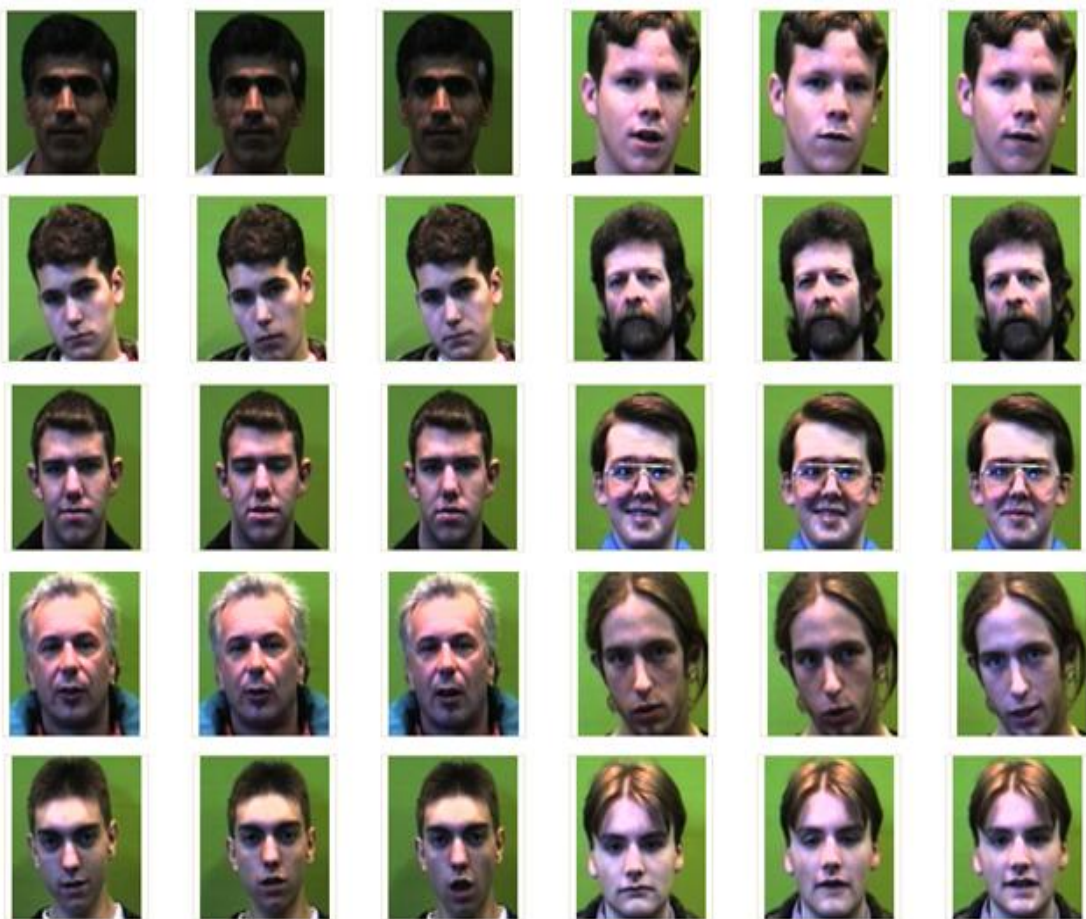


(b)

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

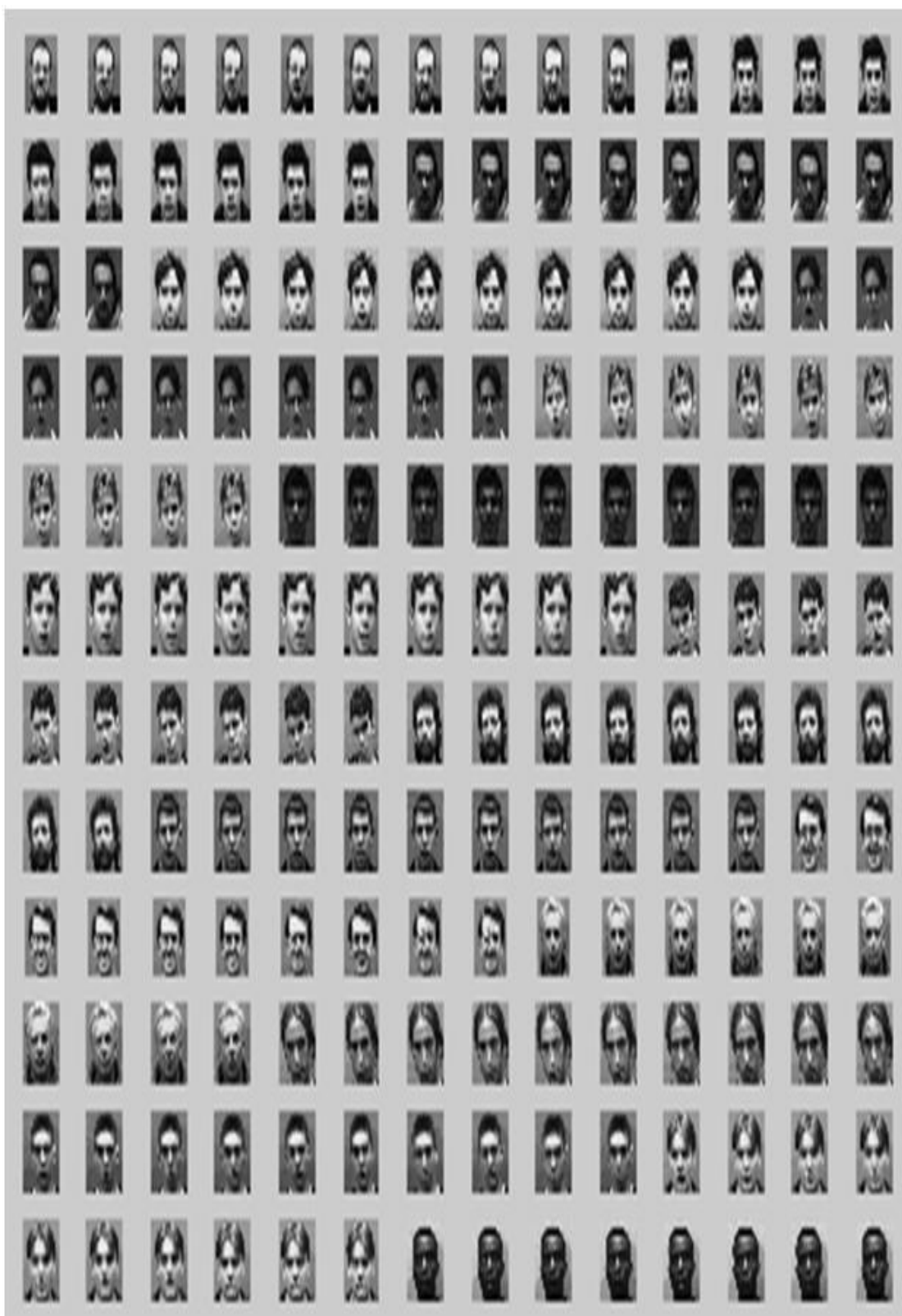


Figure 4.3 : Some of the Gray Scale Images of Face94 Dataset used for Testing

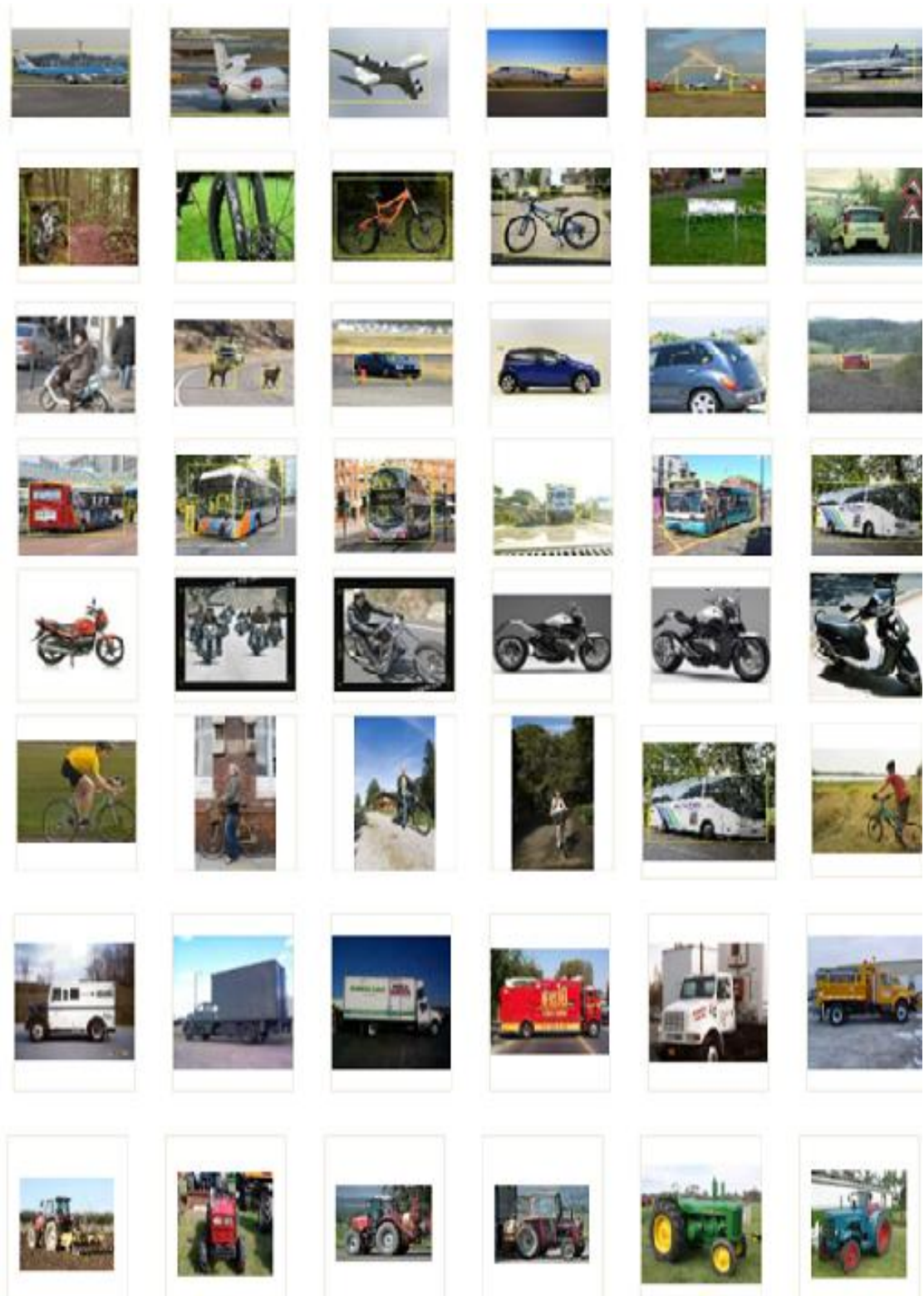


Figure 4.4 : Vehicle Dataset from PASCAL VOC 2006

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.

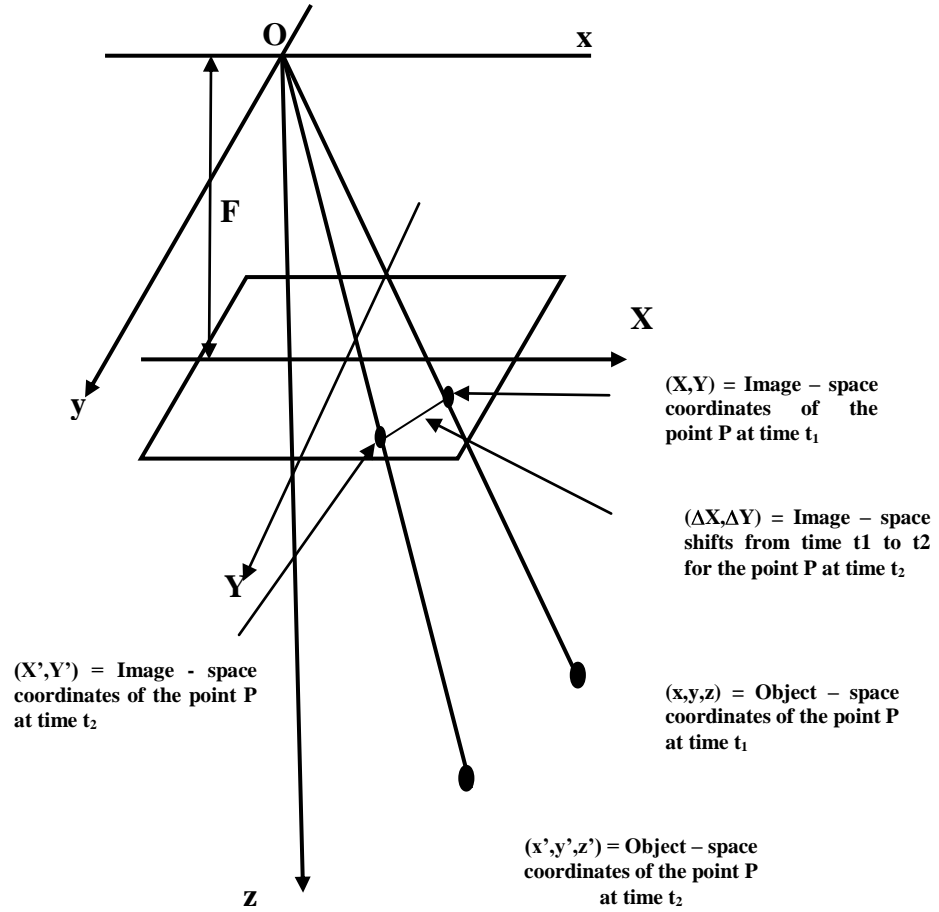


Figure 4.6 : Basic Geometry Model of the Object in 3- D Space

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.

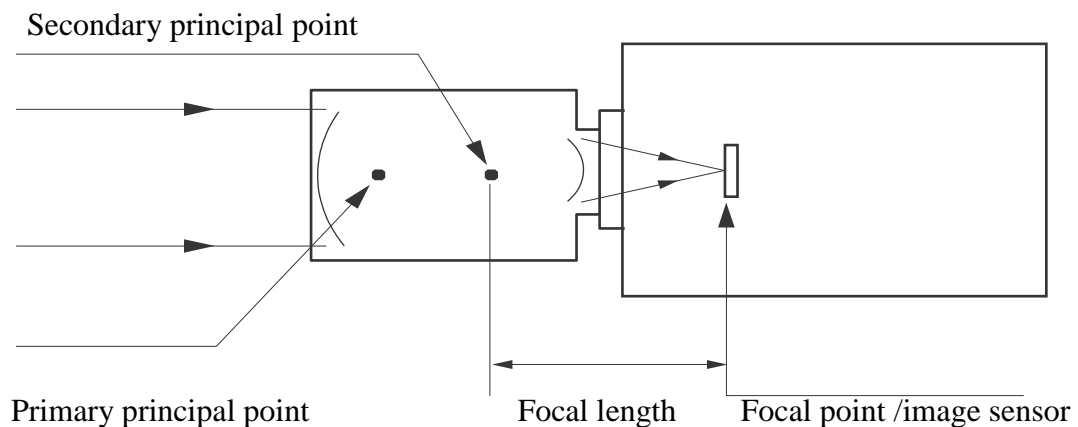


Figure 4.7 : Focal Length in Camera model

- **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.

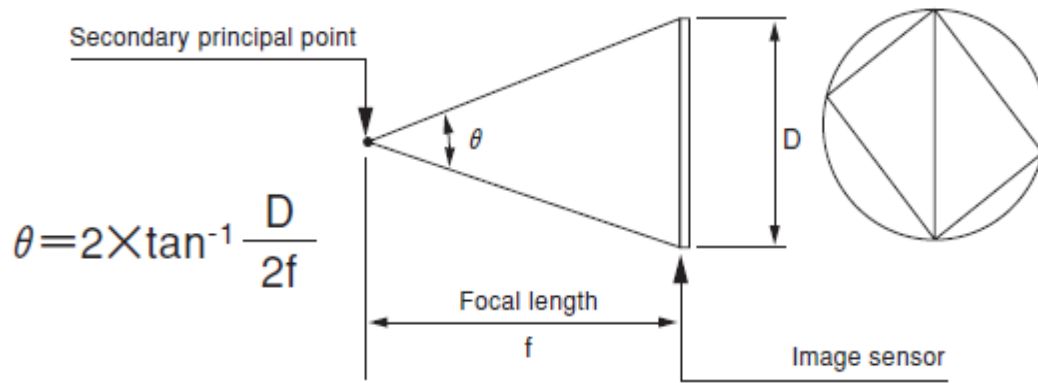


Figure 4.8 : Angle of View in image sensor

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

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

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

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	$\frac{2}{3}$ inch	$\frac{1}{2}$ inch	$\frac{1}{3}$ inch	$\frac{1}{4}$ inch	$\frac{1}{8}$ 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

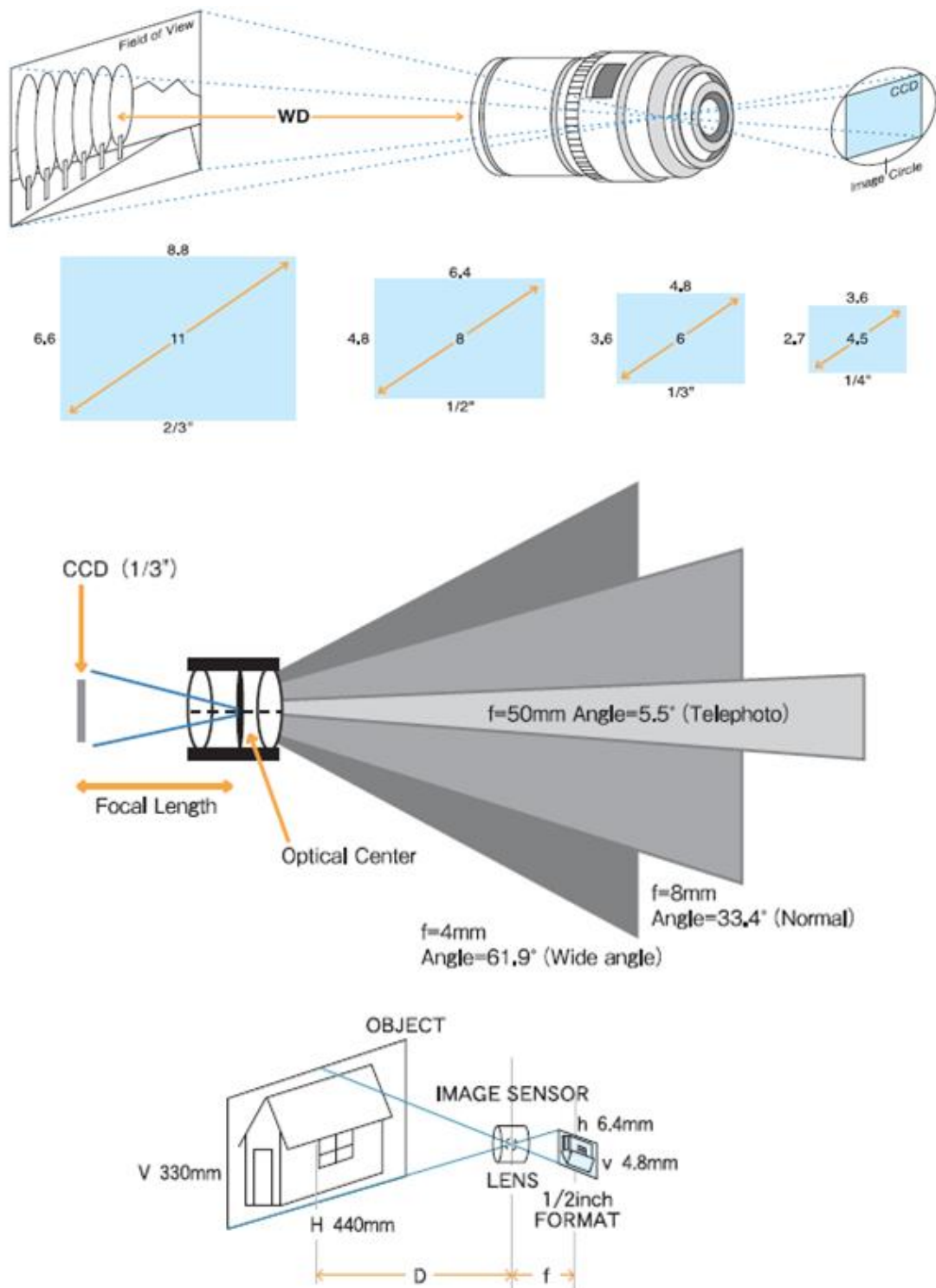


Figure 4.9 : Angle of View in CCTV Camera

- **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:

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

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.

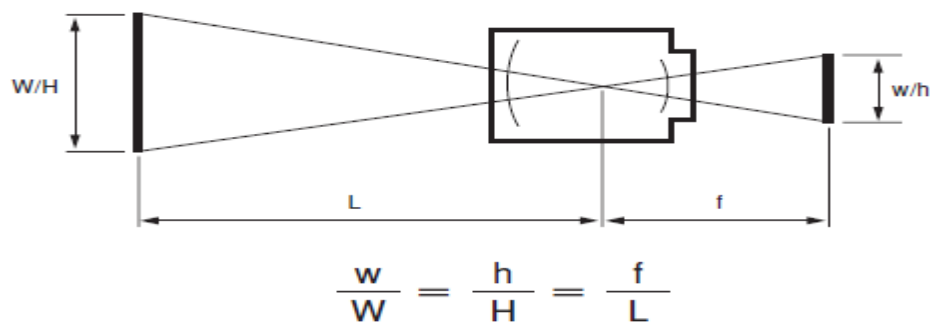


Figure 4.10 : Field of View

W : width of the object

H : height of the object

w : width of the format

$\frac{1}{2}$ format = 6.4 mm, $\frac{1}{3}$ format = 4.8 mm, $\frac{1}{4}$ format = 3.6 mm

h : height of format

$\frac{1}{2}$ format = 4.8 mm, $\frac{1}{3}$ format = 3.6mm, $\frac{1}{4}$ 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.

$$H = \frac{f^2}{C \times F} \quad (4.4)$$

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

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

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 $\frac{1}{2}$ format = 0.015 mm, $\frac{1}{3}$ format = 0.011 mm,
 $\frac{1}{4}$ 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.

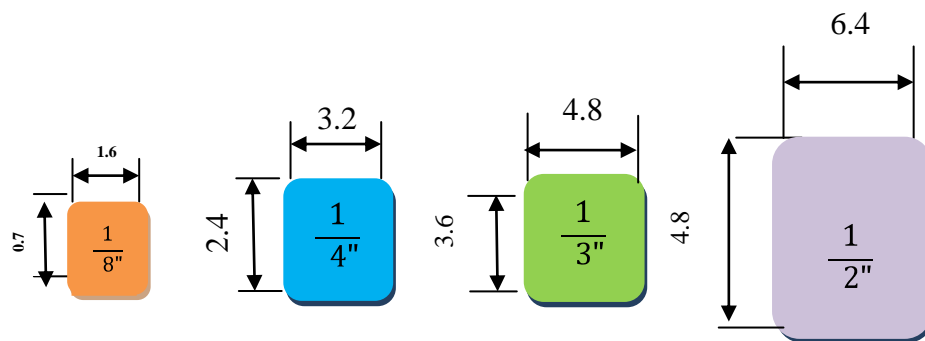


Figure 4.11 : Image Sensor Size

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]:

$$v = \frac{\Delta p}{\Delta t} \quad (4.7)$$

where v is instantaneous velocity vector of a point projected on 2D image space and Δp 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 Δt . The time interval Δt 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 Δt to be used in equation (4.7) is 33.3 milliseconds.

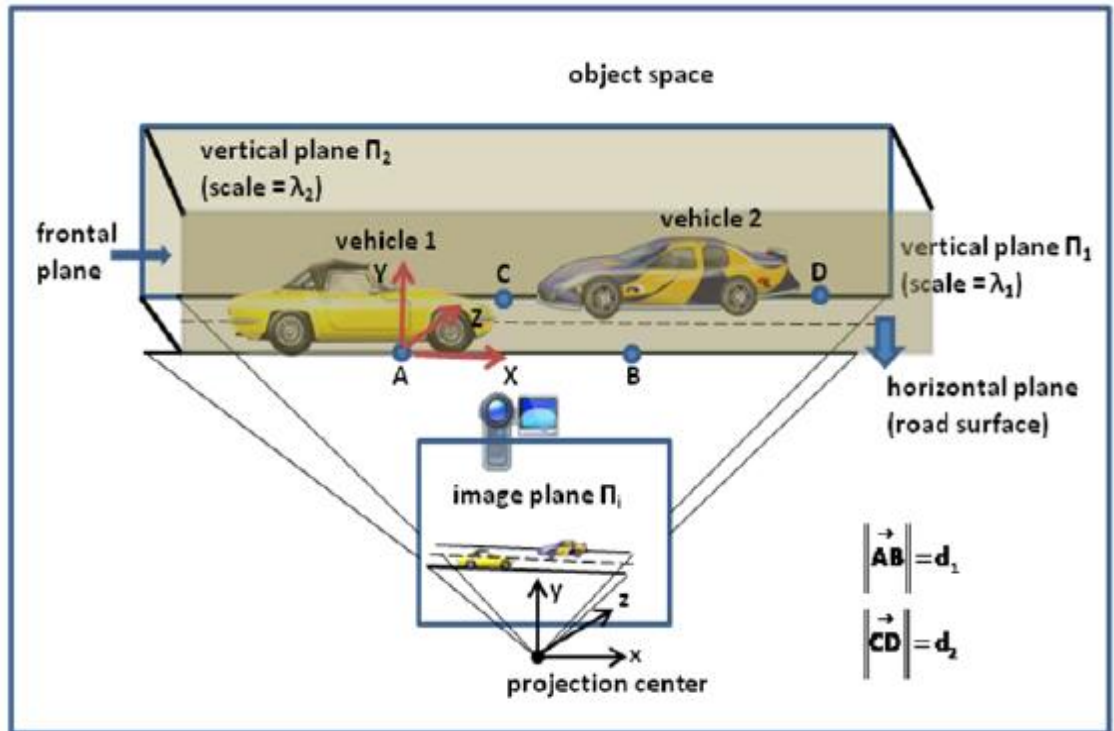


Figure 4.12 : View of Image Acquisition Plane [72]

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 Π_1 with the scale λ_1 . Scales of the vertical planes Π_1 and Π_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 $\lambda_2 = \frac{d_2'}{d_2}$ 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

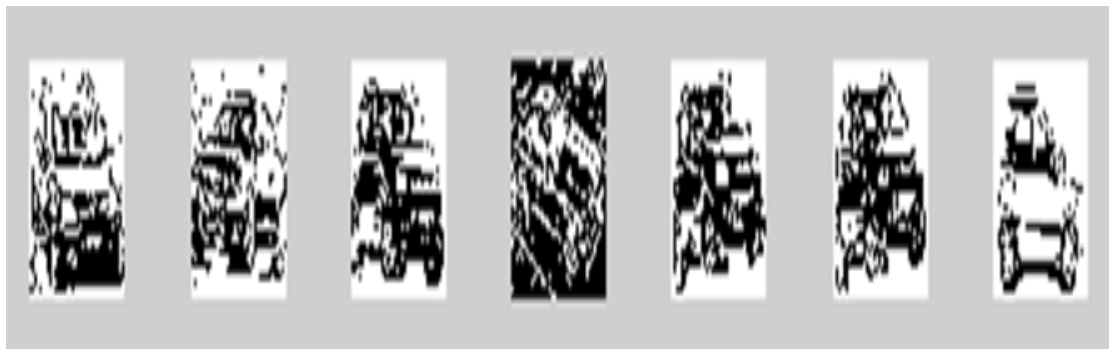


Figure 4.16 : Enhanced Images after performing Pre-processing on VOC 2006 Dataset

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.

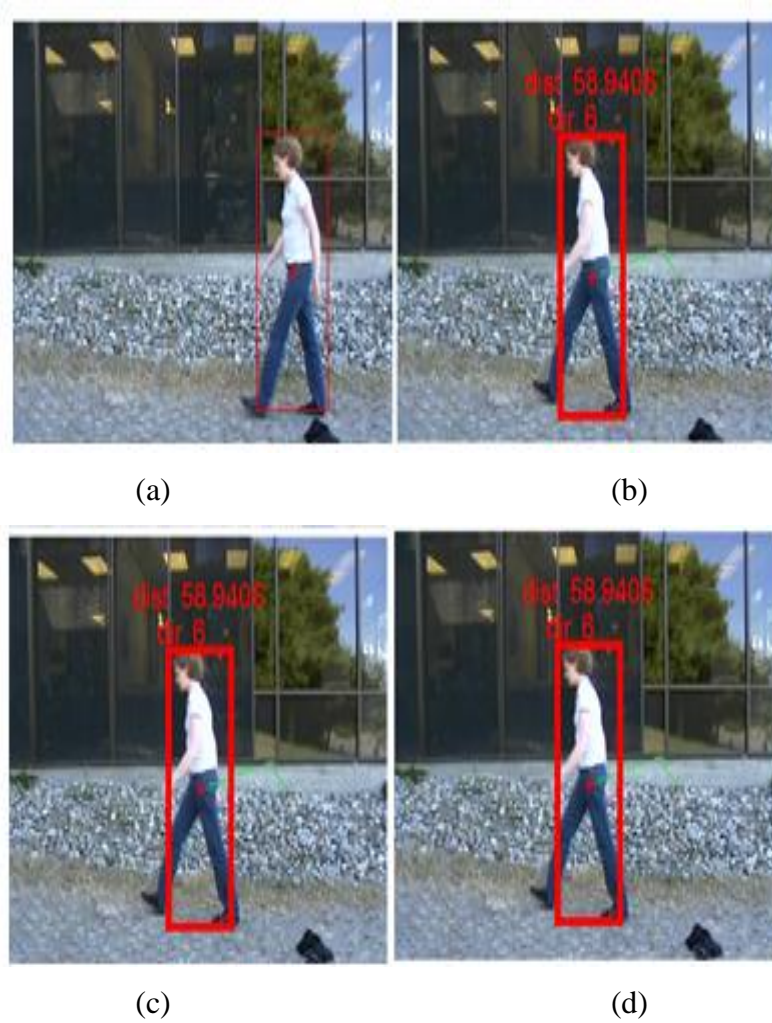
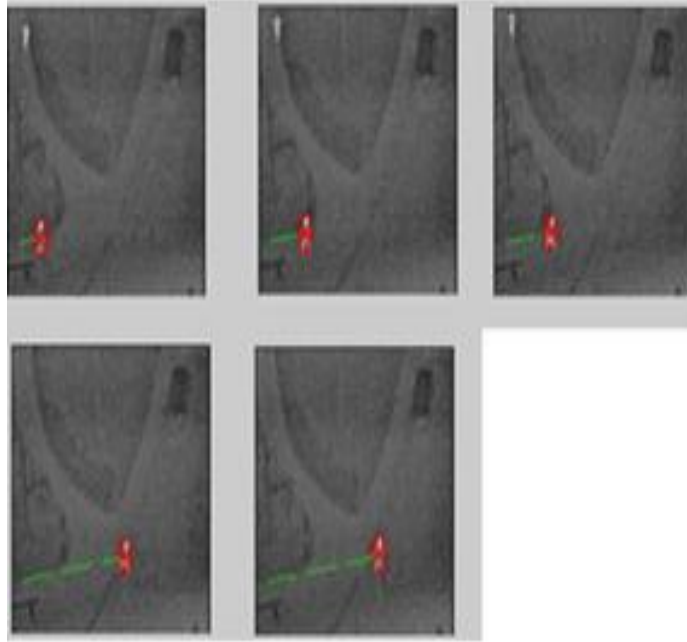
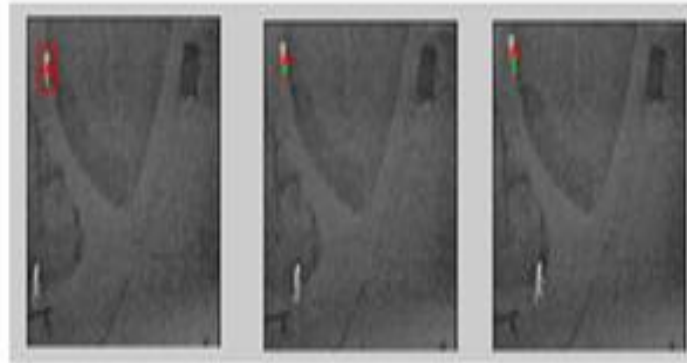


Figure 4.17 : ‘Girl_walking’ Sequence and Tracking Results in Different Frames

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

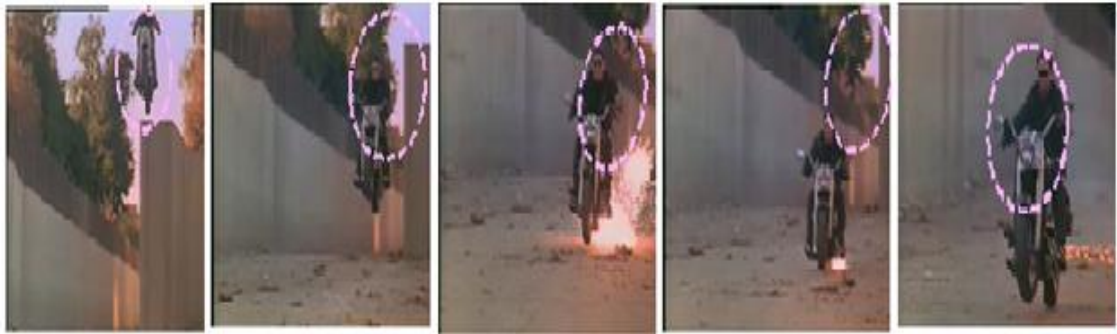


(a)



(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



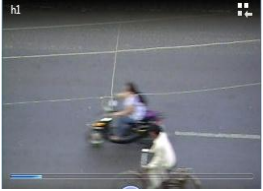


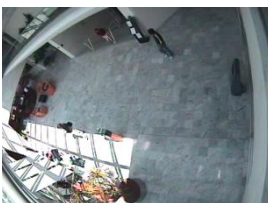

(a)





(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.

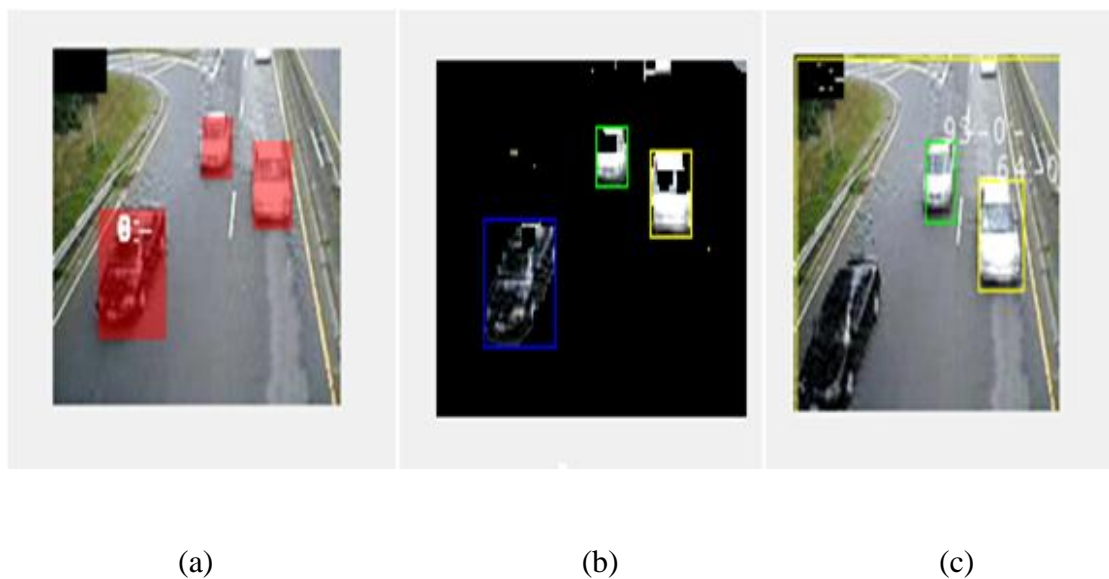


Figure 4.22 : Vehicle Tracking in the viptraffic Sequence (a) Tracking Vehicles
(b) Blob Extraction (c) Region Tracking with Motion Parameters

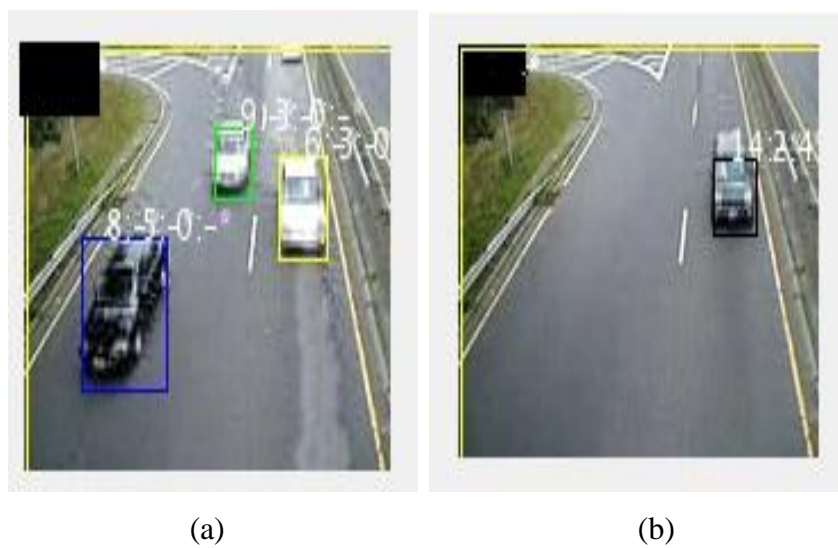
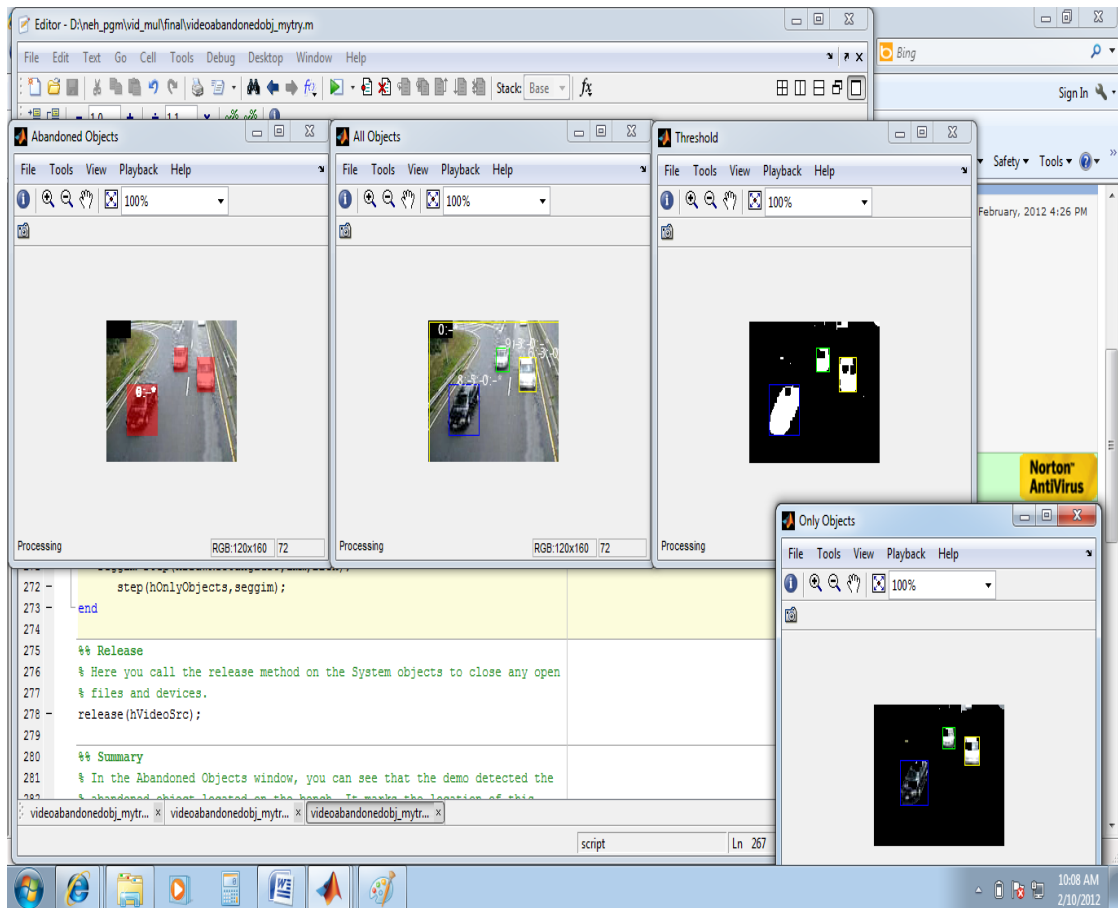


Figure 4.24 : Visualized Results in the Format [Object Number: - Speed: - Direction]
(a) Frame Number 72 (b) Frame Number 113

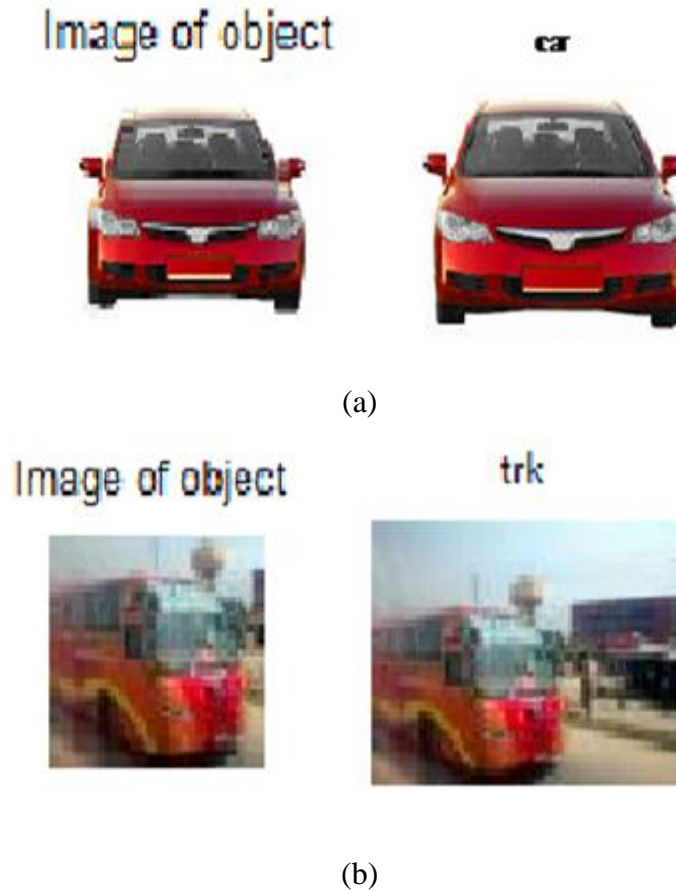








Figure 4.25 : Object Classifier (a) Correct Identification (b) False Identification

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.

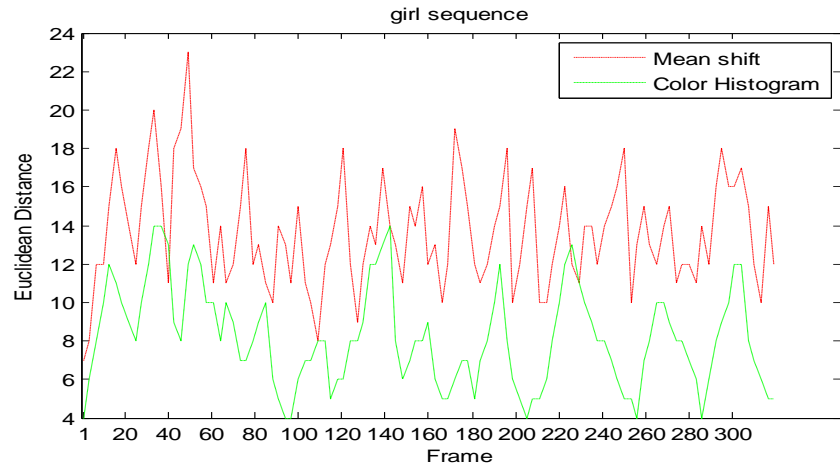
For multiple objects tracking, a procedure has been proposed based on the following principles:

- 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.

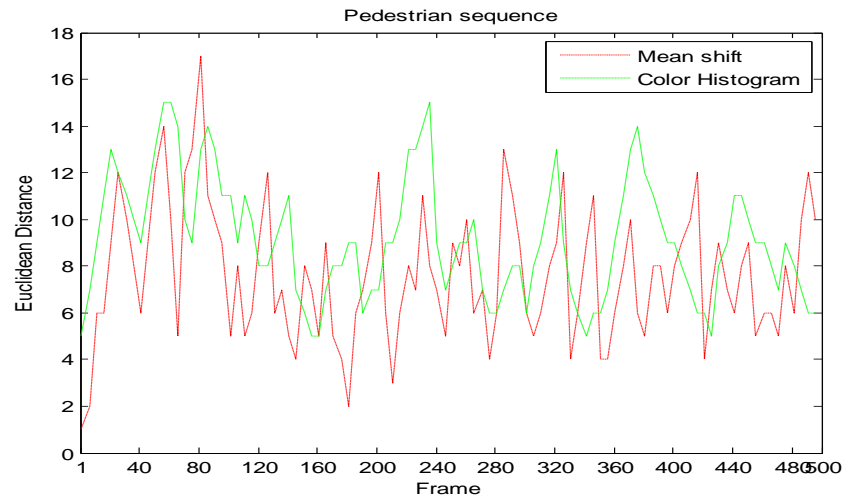
In order to compare the output of the algorithm with the ground truth segmentation, a region matching procedure is adopted which allows to establish a correspondence between the detected objects and the ground truth. Several cases are considered as follows:

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.

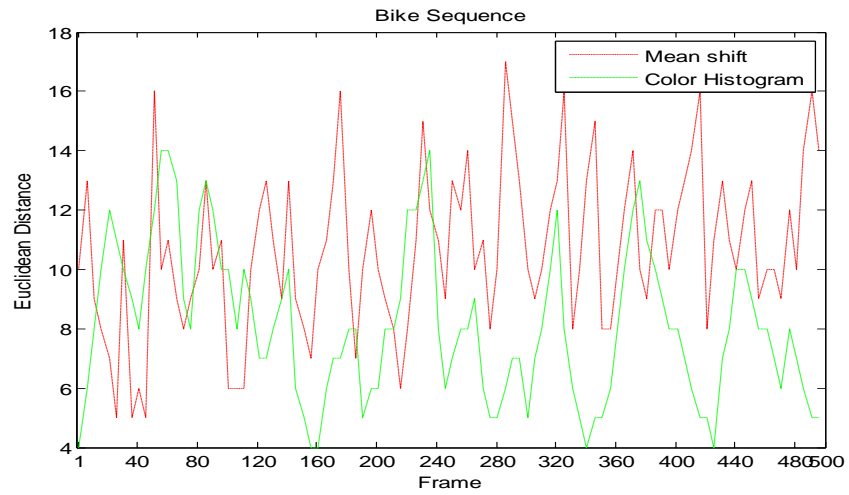
The Figure 4.27 shows the different class target between the actual ground truth and tracked target. The performance matrix for evaluation can be generated using the above conditions. The matrix generated for the Figure 4.27 is shown in the Table 4.8.



(a)



(b)



(c)

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

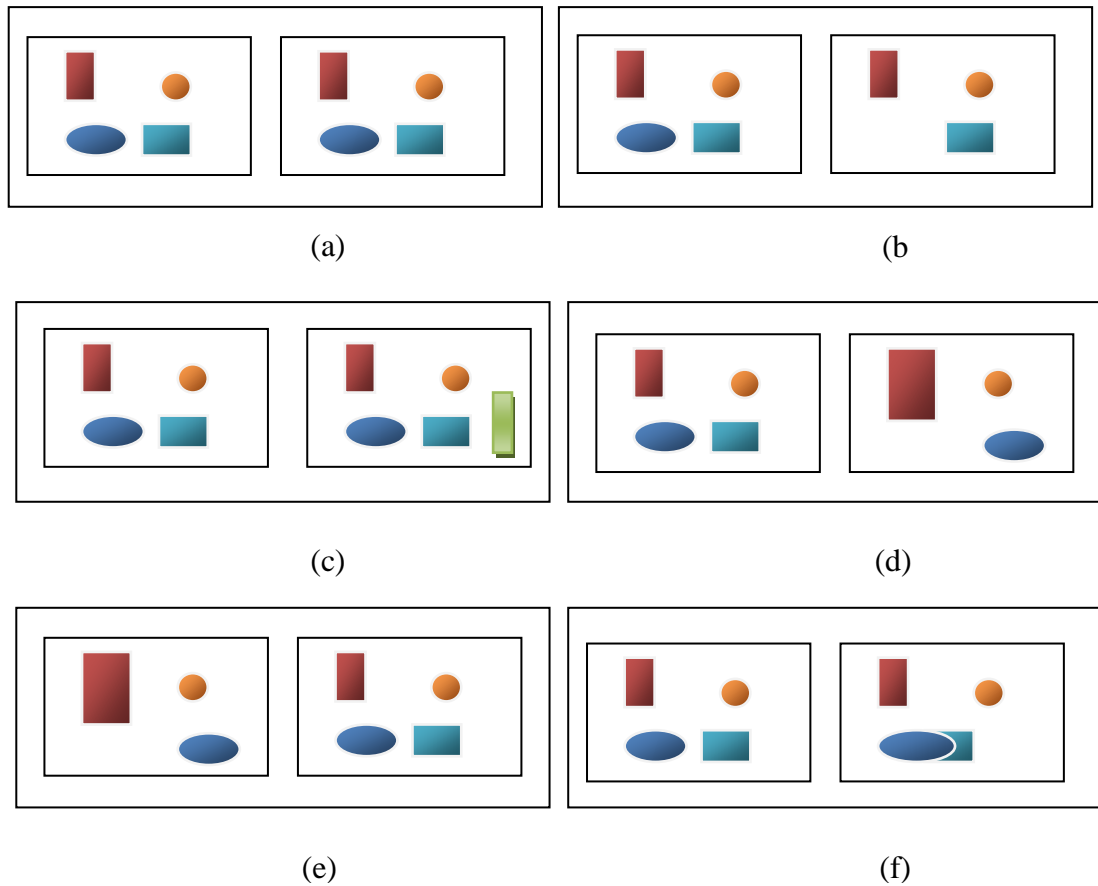


Figure 4.27 : Region Matching Cases: (a) Perfect Match (b) Detection Failure
(c) False Match (d) Merge (One Correspondence for More than One Target)
(e) Split (More than One Correspondence for One Target)
(f) Split and Merge (Conditions (d) and (e) Together)

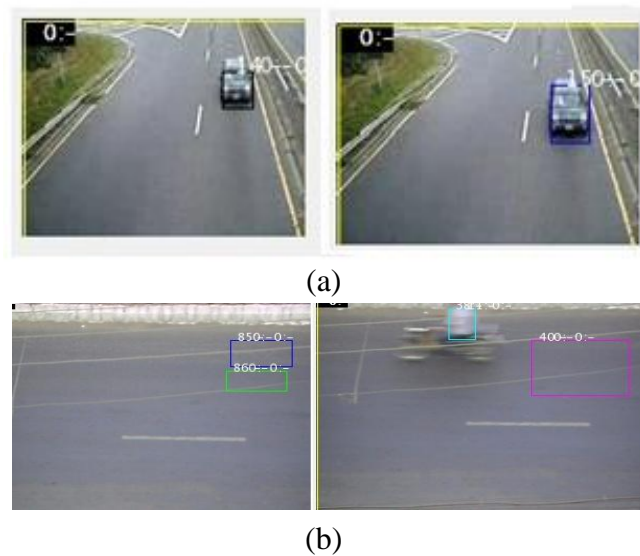


Figure 4.28 : Region Matching Failure: (a) Detection Failure of the Same Object in the Frame Number 72 and Frame Number 73 (Tracked as a new object) (b) False Match in Two Different Frames

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)

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

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.

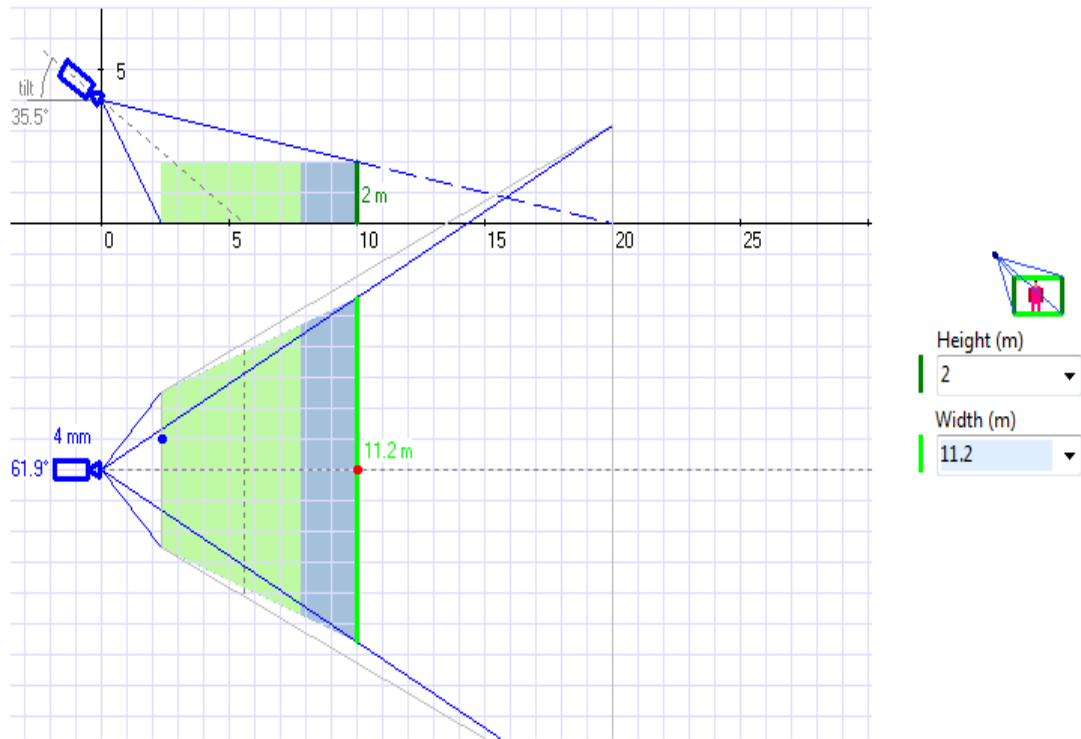


Figure 4.29 : Camera Parameters Calculations using Trigonometry Functions

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

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

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

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

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 $3 \times$ 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

$$s = \frac{d}{f} = \frac{h}{H} = \frac{w}{W} \quad (4.9)$$

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

$$V = m * s * 3.6 \quad (4.10)$$

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 / $3 \times \text{zoom}$
5.8mm / $3 \times \text{zoom} = 17.4 \text{ mm}$	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.

Chapter 5

5 Conclusions and Future Scope

This thesis has been proposed with the aim of developing a low cost software model for object identification and to carry out the estimation of motion analysis of the object. Because of the fact that there is not yet an outperforming algorithm, even though the literature on object tracking is very rich, various approaches are merged together for achieving better results. Different approaches have been tried for different tasks during the development of the new proposed algorithms implemented and discussed in the previous chapters.

Discrete Wavelet Transform was implemented for object identification initially, but during literature survey it was found to be superseded by Discrete Curvelet Transform and Discrete Contourlet Transform. For Visual tracking, Digital Signature algorithm has been tried for region matching. But the approach has not been extended further as it was not able to handle the shape variations. Scale Invariable Feature Transform (SIFT) could not implemented, as it required heavy texture of the object. Gradient Vector Force (GVF) snake algorithm also have been tried, but could not handle the tracking, as it needs good initialization parameters close to the object boundaries in order to segment the objects. Another disadvantage of GVF tracking is the risk of error propagation, due to wrong detected segments which is used for initialization in the next frame. Kalman Filtering is not implemented due to the requirement of model

parameters for each moving object. Finally Hybrid tracker based on Color histogram and discrete Contourlet Transform has been proposed.

To conclude this section, it is interesting to enumerate the advantages and disadvantages of the proposed algorithm.

5.1 Conclusions

The visual tracking algorithm for multiple object tracking based on Contourlet transform works more efficiently than the standard blob tracking method which is based on area and Centroid of the object. We introduced tracking method based on the 3D color histogram for color feature extraction and tracking the region. Region matching has been carried out using 2D seven invariant moments calculated from the histogram, which needs to match only seven descriptors of each region. So the execution time taken by the algorithm is less than the conventional matching methods. Also to overcome the problem of same color descriptor region, feature extraction using Contourlet transform has been introduced effectively. Algorithm uses multiple methods for tracking the object in efficient way, which can handle the color features as well as edge point features.

The visual tracking algorithm for multiple object tracking based on the color features and Contourlet transform are more efficient than the conventional methods. The proposed algorithm has been implemented embedding more challenges. The algorithm can handle the object tracking of varying size. General aperture problems which occur due to the motion of camera or light reflection from the surface can be handled by pre processing techniques. The method has no restrictions such as prior object shape or motion model assumptions. Execution speed of the proposed approach is sufficient enough to be used for real time applications. It can handle partial occlusion very well. Feature extractions using Contourlet Transform can be used for object identification as well as region matching that serves the dual purpose as it saves the time for execution as well as increases the efficiency for tracking along with

identification of an object. Algorithm can well handle the shadow, variation and illumination changes due to the change in lighting conditions.

5.2 Limitations and Future Scope

Although the visual tracking algorithm proposed here is robust in many of the conditions, it can be made more robust by eliminating some of the limitations as listed below:

- In the Single Visual tracking, the size of the template remains fixed for tracking. If the size of the object reduces with the time, the background becomes more dominant than the object being tracked. In this case the object may not be tracked.
- Fully occluded object cannot be tracked and considered as a new object in the next frame.
- Foreground object extraction depends on the binary segmentation which is carried out by applying threshold techniques. So blob extraction and tracking depends on the threshold value.
- Splitting and merging cannot be handled very well in all conditions using the single camera due to the loss of information of a 3D object projection in 2D images.
- For Night time visual tracking, night vision mode should be available as an inbuilt feature in the CCTV camera.

To make the system fully automatic and also to overcome the above limitations, in future, multi- view tracking can be implemented using multiple cameras. Multi view tracking has the obvious advantage over single view tracking because of wide coverage range with different viewing angles for the objects to be tracked.

In this thesis, an effort has been made to develop an algorithm to provide the base for future applications such as listed below.

- In this research work, the object Identification and Visual Tracking has been done through the use of ordinary camera. The concept is well extendable in applications like Intelligent Robots, Automatic Guided Vehicles, Enhancement of Security Systems to detect the suspicious behaviour along with detection of weapons, identify the suspicious movements of enemies on borders with the help of night vision cameras and many such applications.
- In the proposed method, background subtraction technique has been used that is simple and fast. This technique is applicable where there is no movement of camera. For robotic application or automated vehicle assistance system, due to the movement of camera, backgrounds are continuously changing leading to implementation of some different segmentation techniques like single Gaussian mixture or multiple Gaussian mixture models.
- Object identification task with motion estimation needs to be fast enough to be implemented for the real time system. Still there is a scope for developing faster algorithms for object identification. Such algorithms can be implemented using FPGA or CPLD for fast execution.

Publications

- [1] N.G.Chitaliya and A.I.Trivedi, "Vehicle Detection and pose estimation of Vehicle using Eigenspaces," in *in proceedings of IEEE Sponsored National Conference on Innovation and Applications of Mathematical Modelling Technique in Engineering and Mathematics*, VallabhVidhyanagar, 2008, pp. 54-57.
- [2] N.G.Chitaliya and A.I.Trivedi, "Image Segmentation using Watershed Transformation for Object Identification for Machine Learning," in *Recent Developments and Applications of Probability theory, Random process and Random Variables in Computer Science*, Tiruvalla, 2008, pp. 103-107.
- [3] N.G.Chitaliya and A.I.Trivedi, "Feature Extraction and Classification using Wavelet-PCA and Neural Network for Appearance based Object Classification," in *International Conference on signals, systems and automation*, VallabhVidhyanagar, 2009.
- [4] N.G.Chitaliya and A.I.Trivedi, "Feature Extraction using Wavelet-PCA and Neural network for application of Object Classification and Face Recognition," in *proceeding of IEEE International Association of Computer Science & Technology(ICCEA)*, vol. 1, Bali,Indonesia, 2010, pp. 510-514.
- [5] N.G.Chitaliya and A.I.Trivedi, "An Efficient Method for Face Feature Extraction and Recognition based on Contourlet Transform and Principal Component Analysis using Neural Network ," *International Journal of Computer Application*,

vol. 6, no. 4, pp. 28-34, September 2010.

- [6] N.G.Chitaliya and A.I.Trivedi, "An efficient method for Face Feature Point Extraction & Recognition based on Contourlet Transform and Principal Component Analysis," in *International Conference and Exhibition on Biometric Technology (ICEBT)*, vol. 2, Coimbatore, September 2010, pp. 52-61.
- [7] N.G.Chitaliya and A.I.Trivedi, "Novel Block matching algorithm using Predictive motion vector for Video Object Tracking based on colour histogram," in *IEEE International Conference on Electronics Computer Technology (ICECT 2011)*, Kanyakumari, India, 2011, pp. 81-85.
- [8] N.G.Chitaliya and A.I.Trivedi, "Comparative analysis using fast discrete Contourlet Transform and Curvelet Transform via wrapping for feature extraction and recognition", *Communicated to Springer Multimedia tools and Applications*.
- [9] N.G.Chitaliya and A.I.Trivedi, "Automated Vehicle Identification System based on Discrete Curvelet Transform for Visual Surveillance and Traffic Monitoring System", *To appear in International Journal of Computer Application (IJCA)*.

Bibliography

- [1] R.Szeiliski, *Computer Vision: Algorithms and Applications*. New York: Springer, 2010.
- [2] M.T.Jones, *Artificial Intelligence : A Systems Approach*. Hingham: Infinity Science Press, 2008.
- [3] L.Wang, W.M.Hu, and T.N.Tan, "Recent Developments in Human Motion Analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 588-601, 2003.
- [4] R.Polikar, *Pattern Recognition in Bioengineering*. New York: Wiley Encyclopedia of Biomedical Engineering, 2006, vol. 4, pp. 2695-2716.
- [5] L.K.Jones, "Constructive Approximations for Neural Networks by Sigmoid Functions," in *Proceedings of IEEE*, vol. 78, 1990, pp. 1586-1589.
- [6] S.M.Weiss and C.A.Kulikowski, *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. San Mateo, CA: Morgan Kaufmann, 1991.
- [7] S.Haykin, *Neural Networks: A Comprehensive Foundation.*: Prentice Hall, 1999.
- [8] C.R.Jung and J.Scharcansk, "Robust Watershed Segmentation using Wavelets,"

Image and Vision Computing, vol. 23, pp. 661-669, 2005.

- [9] J. Barron, D. Fleet, and S. Beauchemi, "Performance of Optical Flow Techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 42-77, 1994.
- [10] J.Mundy, "Object Recognition in the Geometric Era: A Retrospective," *Springer-Verlog*, 2006, pp. 3-29.
- [11] C.Harris and M.Stephens, "A Combined Corner and Edge Detector," in *Proceedings of the Fourth Alvey.Vision Conference*, Manchester,UK, 1988, pp. 147-151.
- [12] D.Lowe, "Distinctive Image Features from Scale-Invariant Key Points," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [14] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.
- [15] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine* , vol. 2, no. 6, pp. 559-572, 1901.
- [16] M.Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.
- [17] M.Turk and A.Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuro Science*, vol. 3, no. 1, pp. 71-86, 1991.

- [18] T. H.Thi, K. Robert, S. Lu, and J. Zhang, "Vehicle Classification at Night Time using Eigenspace and Support Vector Machine," in *Congress on Image and Signal Processing*, Sanya,China, 2008, pp. 424-426.
- [19] H.S.Sahambi and K.Khorasani, "A Neural network Appearance Based 3D Object Recognition using Independent Component Analysis," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 138-149, 2003.
- [20] C.Zhang, X.Chen, and W.B.Chen, "A PCA-based vehicle classification framework," in *Proceedings of IEEE International Conference on Data Engineering*, 2006, pp. 17-27.
- [21] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961-1005, 1990.
- [22] B.J. Woodford and N.K.Kasabov, "A Wavelet Based Neural Network Classifier for Temporal Data," in *Proceedings of 5th Austrailia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, Dunedin, New Zealand, 2001, pp. 79-85.
- [23] M. N. Do and M. Vetterli, "The Contourlet Transform: An Efficient Directional Multiresolution Image Representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091-2106, 2005.
- [24] J. Zhou, A.L. Cunha, and M.N.Do, "Nonsubsampled Contourlet transform: Construction and Application in Enhancement," in *Proceedings of International Conference on Image Processing*, vol. 1, 2005, pp. 469-472.
- [25] Y.Yan, R. Muraleedharan, X. Ye, and L.A. Osadciw, "Contourlet Based Image Compression for Wireless Communication in Face Recognition System," in *Proceedings of IEEE International Conference on Communications*, Beijing, China, 2008, pp. 505-509.

- [26] B. Yang, S.T. Li , and F.M.Sun, "Image Fusion using Nonsubsampled Contourlet Transform," in *Proceedings of 4th International Conference on Image and Graphics*, Chengdu,China, 2007, pp. 719-724.
- [27] Ch. Srinivasan Rao, S. Srinivas Kumar, and B. N. Chatterji, "Content Based Image Retrieval using Contourlet Transform," *International Journal on Graphics, Vision and Image Processing*, vol. 7, no. 3, pp. 9-15, 2007.
- [28] D.L.Donoho and M.R.Duncan, "Digital Curvelet Transform: Strategy, Implementation and Experiments," Stanford University, California, Technical Report 1999.
- [29] J.L. Starack, E.J. Candes, and D.L. Donoho, "The Curvelet Transform for Image Denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670-684, 2002.
- [30] L. Dettori and L. Semler, "A Comparison of Wavelet, Ridgelet and Curvelet-Based Texture Classification Algorithms in Computed Tomography," *Computers in Biology and Medicine*, vol. 37, no. 4, pp. 486-493, 2007.
- [31] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D Object Recognition from Range Images using Local Feature Histograms ," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition* , vol. 2, Kauai,HI,USA, 2001, pp. 394-399.
- [32] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Reognition Letters*, vol. 27, pp. 861-874, 2006.
- [33] G. Giacinto and F. Roli, "Methods for Dynamic Classifier Selection," in *Proceedings of 10th International Conference on Image Analysis and Processing*, Venice,Italy, 1999, pp. 659-665.

- [34] M. Piccardi, "Background Subtraction Techniques: A Review," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 2004, pp. 3099-3104.
- [35] A.Mittal and N. Paragios , "Motion-Based Background Subtraction using Adaptive Kernel Density Estimation," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, pp. 302-309.
- [36] S.Cheung and C. Kamath, "Robust Techniques for Background Subtraction in Urban Traffic Video," in *Proceedings of 16th Annual Symposium on Electronic Imaging, Visual Communications Image Processing*, San Jose,USA, 2004, pp. 881-892.
- [37] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.
- [38] C. Stauffer, W. Eric, and L. Grimson, "Learning Patterns of Activity using Real-Time Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, 2000.
- [39] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, 2000.
- [40] T. Boult, R. Micheals, X. Gao, and M. Eckmann, "Into the Woods: Visual Surveillance of Non-Cooperative Camouflaged Targets in Complex Outdoor Settings," in *Proceedings of IEEE*, Bethlehem,PA, 2001, pp. 1382-1402.
- [41] A. Yilmaz, "Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection," in *Proceedings of IEEE Computer Society*

- Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, 2007, pp. 1-6.
- [42] R.T. Collins, "Mean-Shift Blob Tracking Through Scale Space," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 234-240.
 - [43] B. Zhang, W.Tian, and Z. Jin, "Joint Tracking Algorithm using Particle Filter and Mean Shift with Target Model Updating," *Chinese Optics Letters*, vol. 4, no. 10, pp. 569-572, 2006.
 - [44] C.E.Erdem, "Video Object Segmentation and Tracking using Region-Based Statistics," *Image and Vision Computing*, vol. 25, no. 8, pp. 1205-1216, 2007.
 - [45] J. B. Xu, L. M. Po, and C. K. Cheung, "Adaptive Motion Tracking Block Matching Algorithms for Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1025-1029, 1999.
 - [46] K. Fukunaga and L. D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32-40, 1975.
 - [47] Y. Cheng, "Mean Shift, Mode Seeking, and Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790-799, 1995.
 - [48] H.Yu, J. Wei, and J. Li, "Object Tracking by Mean Shift Based on Colour Distribution and Simulated Annealing," in *Proceedings of International Seminar on Future Information Technology and Management Engineering*, Sanya, China, 2009, pp. 128-131.
 - [49] R. Venkatesh Babu., P. Pérez, and P. Bouthemy, "Robust Tracking with Motion

- Estimation and Local Kernel-Based Colour Modeling," *Image and Vision Computing*, vol. 25, no. 8, pp. 1205-1216, 2007.
- [50] Z. Zivkovic and B. Krose, "An EM-like Algorithm for Colour-Histogram-Based Object Tracking," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, vol.1, Washington, DC, USA, 2004, pp. 798-803.
 - [51] H. Zhou , Y. Yuan , and C. Shi, "Object Tracking using SIFT Features and Mean Shift," *International journal of Computer Vision and Image Understanding*, pp. 345-352, 2009.
 - [52] C. Yang, R. Duraiswam, and L. Davis, "Efficient Mean-Shift Tracking via a New Similarity Measure," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA,USA, 2005, pp. 176-183.
 - [53] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.
 - [54] T. S. Ling, L K. Meng, L. M. Kuan, Z. Kadim, and A. A. B. Al-Deen, "Colour-based Object Tracking in Surveillance Application," in *Proceedings of International MultiConference of Engineers and Computer Scientist*, Hong Kong, 2009, pp. 1-6.
 - [55] D. Lowe, "Robust Model-Based Motion Tracking Through the Integration of Search and Estimation," *International Journal of Computer Vision*, vol. 8, pp. 113-122, 1992.
 - [56] S. T. Birchfield and S. Rangarajan, "Spatial Histograms for Region-Based Tracking," *ETRI Journal*, vol. 29, no. 5, pp. 697-699, 2007.

- [57] M. A. Zaveri, S. N. Merchant, and U. B. Desai, "Robust Neural Net Based Data Association and Multiple Model Based Tracking of Multiple Point Targets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 37, pp. 337-351, 2007.
- [58] M. Ezhilarasan and P. Thambidurai, "Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression," *Journal of Computer Science*, vol. 4, no. 4, pp. 282-289, 2008.
- [59] J. R. Jain and A. K. Jain, "Displacement Measurement and its Application in Inter Frame Coding," *IEEE Transactions on Communications*, vol. 29, pp. 1799-1808, 1981.
- [60] H. Sidenbladh and M. Black, "Learning the Statistics of People in Images and Video," *International Journal of Computer Vision*, vol. 54, no. 1, pp. 181-207, 2003.
- [61] X. Jing and L. Chau, "An efficient three-step search algorithm for Block Motion Estimation," *IEEE Transactions on Multimedia*, vol. 6, pp. 435-438, 2004.
- [62] L.M. Po and W. C. Ma, "A Novel Four Step Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 313-317, 1996.
- [63] L.K. Liu and E. Feig, "A Block Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, 1996.
- [64] S. Zhu and K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287-290, 2000.

- [65] C. Cheung and L. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 1168-1177, 2002.
- [66] C. Zhu., X. Lin., L. Chau, and L. Po, "Enhanced Hexagonal Search for Fast Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 1210-1214, 2004.
- [67] Y. Nie and K.K. Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEE Transactions on Image Processing*, vol. 11, pp. 1442-1449, 2002.
- [68] R. Cutler and L. S. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 781-796, 2000.
- [69] A. J. Lipton, "Local Application of Optic Flow to Analyse Rigid Versus Non-Rigid Motion," in *Proceedings of International Conference on Computer Vision Workshop on Frame-Rate Applications*, Kerkyra, Greece, 1999.
- [70] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532-540, 1983.
- [71] E.J.Candes, L.Demanet, D. L. Donoho, and L.Ying, "Fast Discrete Curvelet Transforms," *Multiscale Modelling and Simulation*, vol. 5, no. 3, pp. 861-899, 2005.
- [72] "Technical information on CCTV camera modelling,"
www.videosec.com/education/lens-glossary.pdf.
- [73] S. Doğan, M. S. Temiz, and S.Külür, "Real Time Speed Estimation of Moving

- Vehicles from Side View Images from an Uncalibrated Video Camera," *Sensors*, vol. 10, pp. 4805-4824, 2010.
- [74] J. C. Nascimento and J. S. Marques, "Performance Evaluation of Object Detection algorithms for video Surveillance," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 761-774, 2006.
- [75] J. Ma and G. Plonka, "The Curvelet Transform," *IEEE Signal Processing Magazine*, vol. 27, no. 2, pp. 118-133, 2010.
- [76] MSU Video Group. (2004)
http://compression.ru/download/articles/color_space/ch03.pdf.
 [Online]. HYPERLINK www.compression.ru
- [77] Vidit Jain and Amitabha Mukherjee. (2002) The Indian Face Database. [Online].
 HYPERLINK
<http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/>
- [78] Essex Face94 database.
 [Online]. HYPERLINK <http://dces.essex.ac.uk/mv/allfaces/faces94.zip>
- [79] Girl Sequence.
 [Online]. HYPERLINK <http://www.csc.kth.se/~hedvig/>
- [80] Cow sequence.
 [Online]. HYPERLINK <http://www.robots.ox.ac.uk/~vgg/data/mosegobicut>
- [81] The PASCAL Visual Object Classes 2006 dataset.
 [Online]. HYPERLINK
pascallin.ecs.soton.ac.uk/challenges/VOC/voc2006/

- [82] CAVIAR Test Sequence.
[Online]. HYPERLINK
www.hitech-projects.com/euprojects/canata/datasets/dataset.html
- [83] W. Hu, N. Xie, L. Li, X. Zeng, and S.J. Maybank, "A Survey on Visual Content-Based Video Indexing and Retrieval," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 797-819, 2011.
- [84] M. J. Swain and D.H.Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.
- [85] S. Chand, "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," *International Journal of Mathematical Models and Applied Sciences*, vol. 4, no. 1, pp. 300-307, 2007.
- [86] H. F. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, pp. 1644-1649, 2006.
- [87] R. C. Gonzalez, R. E. Woods and S.L.Eddins, *Digital Image Processing Using MATLAB*, 2nd ed. Knoxville,TN: Gatesmark Publishing, 2009.