Multiple Time Spatial Images for Video-Based Automatic Tracking of Vehicles

by

Niluthpol Chowdhury Mithun

M.Sc. Engg. (EEE)

Department of Electrical and Electronic Engineering

BANGLADESH UNIVERSITY OF ENGINEEING AND TECHNOLOGY

June 2014

The thesis titled „**Multiple Time Spatial Images for Video-Based Automatic Tracking of Vehicles**‟ submitted by **Niluthpol Chowdhury Mithun** Roll No: **0411062262 P** Session **April/2011** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **Master of Science in Electrical and Electronic Engineering** on 18th June 2014.

BOARD OF EXAMINERS

1. ________________________
Dr. S. M. Mahbubur Rahman

Associate Professor Chairman
Department of Electrical and Electronic Engineering (Supervisor)
Bangladesh University of Engineering and Technology
Dhaka-1205, Bangladesh.

2. ________________________
Dr. Taifur Ahmed Chowdhury

Professor and Head Member
Department of Electrical and Electronic Engineering (Ex-officio)
Bangladesh University of Engineering and Technology
Dhaka-1205, Bangladesh.

3. ________________________
Dr. Mohammed Imamul Hassan Bhuiyan

Professor Member
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
Dhaka-1205, Bangladesh.

4. ________________________
Dr. Md. Hasanul Kabir

Assistant Professor Member
Department of Computer Science and Engineering (External)
Islamic University of Technology
Board Bazar, Gazipur 1704, Dhaka, Bangladesh

CANDIDATE’S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

____

Niluthpol Chowdhury Mithun

Roll. 0411062262P

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my humble gratitude to my benevolent advisor, Dr. S. M. Mahbubur Rahman, for his invaluable advice and encouragement. This work would not have been possible without his help, direction and especially his patience, when I took longer than expected to finish things.

I would also like to acknowledge all the faculty members of the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology. The knowledge gathered from here, during my B.Sc. and M.Sc. level study, has helped me significantly in this thesis.

I would also like to express my love and gratitude to my parents, without them my life would not be possible. Special thanks to my wife, Suborna Bhattacharjee, whose patience and unlimited support have made the successful completion of this thesis. At last, I like to thank my friends and colleagues who appreciated me for my work and motivated me a lot.

ABSTRACT

Video-based vehicle tracking has become an active research area due to its numerous transportation related applications. Some common challenges in traditional video-based tracking methods include initialization of tracking, tracking an unknown number of targets, sensitivity to drift from true position due to the variations in lighting condition, scene conditions and camera position in long sequences, and absence of corrective mechanism. In this thesis, a novel approach for unsupervised vehicle tracking algorithms is developed by introducing multiple time-spatial images (MTSIs)-based detection in the Monte-Carlo Particle filter or Kalman filter based-tracking. Such a use of MTSIs in tracking algorithm provides the opportunity of reliable identification of a vehicular object automatically whenever it appears in a scene. Notably, the proposed tracking method employs the concept of multiple numbers of key vehicular frames (KVFs) for each of the vehicular-objects in the traffic. These KVFs allow an accurate estimate of the centroid position of a vehicle in the key frames, due to the fact that the relative sizes of the vehicles captured in the video are maintained in these KVFs. The spatial correspondence of a vehicle in KVFs is then integrated in Particle filter or Kalman filter-based tracking as a corrective measure to alleviate the common problem of drifting and thereby increasing the accuracy in tracking trajectory. Extensive experimentations are carried out in vehicular traffics of varying environments to evaluate the tracking performance of the proposed method as compared with the existing methods. Experimental results demonstrate that the proposed approach not only automates the initialization of tracking procedure, but also increases the accuracy of tracking trajectory evaluated by the closeness of centroids of a vehicular object both in the forward and backward tracking.

TABLE OF CONTENTS

LIST OF ACRONYMS

ANRDT	Average Normalized Relative Distance of Tracking
CCD	Charge-Coupled Device
CMOS	Complementary Metal–Oxide–Semiconductor
JPDAF	Joint Probability Data Association Filtering
KVB	Key Vehicular Blob
KVF	Key Vehicular Frame
LiDAR	Light Detection And Ranging
MHT	Multiple Hypothesis Tracking
MTSI	Mutiple TSI
MTSIIKF	MTSI Integrated Kalman Filter
MTSIIPF	MTSI Integrated Particle Filter
NRDT	Normalized Relative Distance of Tracking
PDF	Probablity Distribution Function
RADAR	Radio Detection And Ranging
ROI	Region of Interest
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
TOB	TSI Object Bolb
TSI	Time-Spatial image
VDL	Virtual Detection Line

LIST OF SYMBOLS

i	Current frame index
L_λ	Line object for morphological operation, with length λ
S_μ	Line object for morphological operation, with length μ
n	VDL or TSI index
N	Total number of VDL in the area of interest
m	Current vehicle index
Ḿ	Total number of vehicle in the sequence
$\mathcal{N}$	2-D Array with size NxḾ, indicates the merging condition of vehicles on TSIs
$\mathcal{M}$	1-D Array with size N, indicates the index of TOB to be considered
D_n	Length of VDL n
D_v	Selected length of VDL for TSI creation
$\mathcal{C}_n^x$	X-axis coordinate of the centroid of TSI object blob on nth TSI
$\mathcal{C}_n^y$	Y-axis coordinate of the centroid of TSI object blob on nth TSI
$\mathcal{W}_n^x$	Width of TSI object blob along X-axis on nth TSI
$\mathcal{W}_n^y$	Width of TSI object blob along Y-axis on nth TSI
$\mathcal{A}_n$	Area of TSI object blob on nth TSI
X_l	Lower limit of X-axis coordinate of TOB
X_u	Higher limit of X-axis coordinate of TOB
Δx	Difference in pixel position between two successive VDLs in a frame
R_p	Resolution of the video in meters per pixel
F_r	Frame rate of the video

N_f Number of frames required for a vehicle to pass two successive VDLs

V_{max} Maximum Speed Limit of traffic

V_{min} Minimum Speed Limit of traffic

T_{C}^{y} Threshold parameter, depends on centroid movement of TOB along Y-axis

$T_{\mathcal{W}}^{y}$ Threshold parameter, which depends on width change of TOB along Y-axis

$T_{\mathcal{W}}^{x}$ Threshold parameter, which depends on width change of TOB along X-axis

$T_{\mathcal{A}}$ Threshold parameter, which depends on area of TOB

$n_{\mathrm{TOB}l\,|\,\mathcal{W}^x}$ Number of merged TOB detected considering back to front occlusion

$n_{\mathrm{TOB}l\,|\,\mathcal{W}^y}$ Number of merged TOB detected considering side by side occlusion

$n_{\mathrm{TOB}l\,|\,\mathcal{A}}$ Number of merged TOB detected using area of TOBs

n_{TOB} Detected number of merged TOBs

X_{i+1} Target state in Particle filter

Z_{i+1} Observed image state in Particle Filter

$p(X_{i+1}|X_i)$ Stochastic state transition model used in Particle Filter at $(i+1)^{\text{th}}$ frame

$p(Z_{i+1}|X_i)$ Observation model used in Particle Filter at $(i+1)^{\text{th}}$ frame

$s_i^{(m)}$ Generated Sample m at i^{th} Frame

$s_i^{x(m)}$ X-axis coordinate of the sample m at i^{th} frame

$s_i^{y(m)}$ Y-axis coordinate of the sample m at i^{th} frame

$\pi_i^{(m)}$ Weight assigned to Sample m at i^{th} Frame

M The sample size in Particle filter algorithm

C_i^x X-axis coordinate of the vehicle center position at i^{th} frame

C_i^y Y-axis coordinate of the vehicle center position at i^{th} frame

$C_i^{\bar{x}}$ Predicted vehicle center coordinate along x-axis at i^{th} frame

$C_i^{\bar{y}}$ Predicted vehicle center coordinate along y-axis at i^{th} frame

$C_i^{\hat{x}}$ Estimated vehicle center coordinate along x-axis at i^{th} frame in Particle filter

$C_i^{\hat{y}}$ Estimated vehicle center coordinate along y-axis at i^{th} frame in Particle Filter

$\tilde{R}$ Random number with normal distribution $\mathcal{N}(\mu,\ ^2)$

Ũ Random number with uniform distribution on the interval $[0,2\pi]$

r_θ The distance between center and detected edge point of object estimated in Particle filter

σ_I Image intensity standard deviation inside the object estimated in Particle filter

$C_{i+1|i}^{\hat{x}}$ Acquired center coordinate along x-axis in Kalman filter at $(i + 1)$ frame

$C_{i+1|i}^{\hat{y}}$ Acquired center coordinate along y-axis in Kalman filter at $(i + 1)$ frame

δt Time between frames

$\bar{v}_{i+1|i}^{x}$ Predicted velocity along x-axis at $(i + 1)^{\text{th}}$ frame

$\bar{v}_{i+1|i}^{y}$ Predicted velocity along y-axis at $(i + 1)^{\text{th}}$ frame

α_i Gain determined by Kalman filter at i^{th} frame, used for filtered center estimate

β_i Gain determined by Kalman filter at i^{th} frame, used for velocity prediction

$C_M^{x(n)}$ Center coordinate along x-axis of a vehicular object in KVF corresponding to n^{th} VDL calculated using the MTSI method

$C_M^{y(n)}$ Center coordinate of a vehicular object along y axis in KVF corresponding to n^{th} VDL calculated using the MTSI method

$C_F^{x(n)}$ Center coordinates of a vehicular object along x-axis in KVF estimated using the Kalman filter-based tracking algorithm

$C_F^{y(n)}$ Center coordinates of a vehicular object in KVF estimated using the Particle filter-based tracking algorithm

$\mathbb{T}_C^x$ Threshold parameter of tracking correction, which depends on the difference between center coordinate estimated using filter and that of KVB

$\mathbb{T}_C^y$ Threshold parameter of tracking correction, which depends on the difference between center coordinate estimated using filter and that of KVB

$\mathbb{T}_W^x$ Threshold parameter of tracking correction mechanism, which depends on the difference between center coordinate estimated using filter and that of KVB and width of KVB along X-axis

$\mathbb{T}_W^y$ Threshold parameter of tracking correction mechanism, which depends on the difference between center coordinate estimated using filter and that of KVB and width of KVB along Y-axis

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

Fatalities and serious injuries caused by traffic-related accidents are globally recognized as a serious and growing problem due to increasing usage of automobiles. A recent study by the World Health Organization has shown that over 1.2 million fatalities and 20 million serious injuries occur worldwide due to traffic-related accidents per year [1]. It is also reported that the associated hospital bill, costs of damaged properties and others incur a loss of more than one percent of the gross domestic product of the world [2]. Thus, with an aim of reducing traffic-related accidents and enhancing the chance of safe and smooth driving of an automobile in traffic, vision-based sensing system for vehicle has been recognized as an active area of research by the government agencies and automobile manufacturers over the past decade [2], [3]. Automatic tracking of vehicles can be very effective in this regard due to its numerous transportation related applications including the safety-aware instructions for driver assistance, monitoring a scene to detect suspicious activities for automatic surveillance or even development of driverless intelligent vehicles.

1.2 Vehicle Tracking: A Review

At present, there are mainly three approaches of vehicle tracking, viz., wireless distributed sensor network-based [4], [5], remote-sensing-based, e.g., radar sensing [6], [7] and lidar sensing [8], [9], and video-based tracking [10], [11]. Vehicle tracking in a sensor network is very similar to tracking a mobile phone in a cell, wherein the vehicles are monitored using radio frequency signals and associated controllers. Since the performance of such tracking is highly dependent on the availability of controlling signals, network-based tracking fails to provide any assistance when the sensor-mounted vehicles are out of range of the network or any surrounding vehicle without having such a sensor within the network.

Further, network-based tracking often shows significant position errors, and hence, cannot be used for instantaneous driver assistance. Remote sensing-based tracking transmits radar or laser signals and estimate the position of objects using the corresponding received signals [12]. In general, the costly radar or lidar sensors cannot provide sufficient field-of-view for vehicle tracking and due to high sensitivity to environmental noise often fails to classify the moving objects [11]. Video-based tracking algorithms, on the other hand, capture video frames from the wide field-of-view from the vehicle using high performance and noise-robust CCD or CMOS cameras. This is why these algorithms very often provide very accurate estimate of the relative positions and classification of surrounding vehicles even in the remote areas.

The techniques for video-based vehicle tracking usually follow two major strategies - one approach treats vehicular object identification in each frame and finding correspondence of the object in the neighboring frames independently [13], [14] and the other treats both these issues jointly [15]- [19]. The first approach is mainly based on recognizing and tracking objects by applying feature extraction and matching algorithms. An approach to detect keypoints with features more suitable for object tracking applications is showed in [20]. The SURF (Speeded Up Robust Features) algorithm claims to be a robust algorithm for object detection, which may be also suitable or tracking applications [21]. Another interesting approach is SIFT (Scale Invariant Feature Transform), which presents considerable potential, due to its promised robustness for different scales and under low level of affine transformations [22], [23].

Since the first approach considers the object identification and estimating object correspondence as two processes in serial, the real time implementation of tracking fails in many cases using this approach; especially for simultaneous tracking of multiple vehicles. Further, this approach in many cases assumes that the vehicle features remain invariant over the frames, and thus tracking performance of the methods using such strategy is also not satisfactory. In general, the second approach uses standard statistical techniques for joint recognition of vehicular objects in the frames and their correspondence over the successive frames. In these methods, moving vehicles in a video traffic are modeled in the standard state space frame work, in which the vehicular objects are assumed to follow certain random processes. Measurements for the random processes include the position/co-ordinates, velocity, and acceleration of vehicular objects in the frames. In order to find the solution of the state space model, the density function of the random processes can be chosen as parametric or non-parametric. The Kalman filter-based vehicle tracking is the most popular among the parametric approaches, which obtains an analytical solution for tracking by assuming a linear dynamics of vehicular movements and a Gaussian distributed intensity of vehicular objects [15], [18]. The Kalman filter based algorithm performs well in most of the cases. However, due to the variations of traffic, weather, and viewing conditions, the intensities of the vehicular objects may follow a non-Gaussian statistics as well as the movements of vehicular objects may follow a non-linear dynamics. Variations of the Kalman filter, such as the extended Kalman Filter has been proposed [24], [25] to apply the constrained Kalman filter to nonlinear systems and nonlinear state constraints and as a result performance improved in complex traffic environments. However, in such cases of non-gaussian intensity statistics and non-linear vehicle movement, the non-parametric approach that uses the Gaussian mixture density function [16] or the Particle filter [17], [19] for describing the nonlinear and non-Gaussian random processes show a significant success over the traditional parametric approaches. Among Particle filtering approaches, Condensation algorithm is commonly used in object tracking [17], [26]. Moreover, a color based Particle filter is proposed in many works [27], [28] for object tracking. In their case, target model of the Particle filter is defined by the color information of the tracked object. According to [27], an adaptive mixture color model is used for updating reference color model to make more robust visual tracking. Mainly color and texture cues are integrated in Particle filter based implementation. Moving edge features is used in [29]. The Particle filtering approach can simultaneously represent multiple hypotheses about a target‟s state. These Particle filter based methods also have the ability to keep a memory about the evolving state of an object over subsequent images, which can be very useful for resolving targets through situations of occlusion. However, Particle filtering approach is computationally more costly than Kalman filtering approach. Additionally, in [30], some other statistical tracking approaches have been reported, which include Joint Probability Data Association Filtering (JPDAF) and Multiple Hypothesis Tracking (MHT) [31], [32]. These approaches address some problems of tracking, but none of them provide a complete solution. Moreover, in most of the cases, they suffer from high computational complexity.

1.3 Scope of the Work

The inherent challenge of the most popular Kalman or Particle filter-based vehicle tracking algorithms is their high sensitivity to drift from the true object position, especially in long sequences for frequent occurrences of occlusion, for change in lighting conditions, for sudden turning or bumping of vehicles. Another problem of such algorithms is that, they are semi-supervised in a sense that the initial co-ordinate of a vehicular object in a video is necessary as an input to start the tracking. Since, they require relatively accurate vehicular-target initialization, it makes automatic tracking of an unknown number of vehicle in a complex scene difficult. Recently, time-spatial image (TSI) using a virtual detection line (VDL) has been shown to be very effective in identifying the vehicular objects from captured video traffics [17], [18]. Thus, this VDL-based identification of vehicles may be used for developing unsupervised tracking algorithms. Further, this method employs a concept of key vehicular frame (KVF) from which the position of a vehicle may be estimated accurately in this frame. In this context, the accurate positions obtained from a suitable set of KVFs may also be used as corrective measures for addressing the drifting problems of the Kalman filter or Particle filter-based vehicle tracking.

1.4 Objective

The main objective of this thesis is to develop a fully unsupervised tracking algorithm that identifies a vehicular object automatically whenever it appears in a scene and to provide an accurate trajectory of the object using the spatial correspondence of the concerned vehicle obtained from MTSI and a Kalman filter or Monte Carlo-based Particle filter. The performance of the proposed video-based tracking algorithm will be investigated and compared with that of the existing methods both in the case of forward and backward tracking.

1.5 Organization

The rest of the thesis is organized as follows.

In Chapter 2, a brief description of TSI based object identification, and how it can be used for tracking initialization are given. The concept of MTSI based vehicular-object identification is also presented in this chapter.

In Chapter 3, a novel vehicle tracking approach is described, which not only automates the Particle filter or Kalman filter-based vehicle tracking algortihms, but also provides a corrective measure, utilizing the co-ordinate position of centroid of that vehicle in the KVFs obtained from multiple TSIs.

In Chapter 4, experimental results are given to show the significance of using MTSI in the existing Kalman or Particle-filre based tarcking. Both the forward and backward tracking trajectories and the normalized relative distance (NRD) of these two trajectories are also examined, to compare the performance.

Finally, in Chapter 5, a general concluding discussion about the contribution and potential impact of the proposed method is presented. Scopes for future work are also given here.

2. MULTIPLE TIME-SPATIAL IMAGES FOR VEHICLE DETECTION

2.1 Introduction

The initial task to start tracking a vehicle from video is to identify the existence of the vehicle, and then locate its coordinates in the frame. The problem with most of the tracking algorithms is that the initial coordinate of a vehicular object in a video is required to feed to start the tracking. Thus, they are not suitable for real-time implementation when vehicle appears suddenly in the video scene.

In order to carry out automated tracking procedure, the tracking method requires an object detection mechanism when the object first appears in the video. Given the object regions in the image, it is then the tracker‟s task to perform object correspondence from one frame to the next to generate the trajectory. A common approach for object detection is to use temporal information computed from frames of video. This temporal information is usually in the form of frame differencing [33], background subtraction [34], [35], optical flow estimation method [37], [35], Gaussian scale mixture model method [38], [39], which highlights changing regions in consecutive frames. It is well known that frame differencing and optical flow estimation methods suffer significantly from accuracy [33], [37] and background subtraction and Gaussian scale mixture model based methods suffer from computational efficiency [34], [35], [38]. However, the major concern is that, the detection method works independently to only detect the initial coordinate and don‟t provide any further assistance to the tracker. Moreover, this auxiliary detection mechanism needs to be performed on every frame to identify the entrance of new vehicle, which introduces substantial computational complexity.

In [40], it has been shown that multiple virtual line based time-spatial image detection method may be used as a reliable vehicle detection mechanism because of its accuracy in detecting vehicles and notable computational efficiency. The MTSI based method utilizes a concept of key vehicular frame, which provides a comparable estimate of the centroid position of vehicles on a selected virtual line in the key frames, due to the fact that the relative sizes of the vehicles captured in the video are maintained in these KVFs. As centroid coordinates of a vehicular object is known in key frames, these mechanism will not only provide a detection mechanism for initialization of tracking, but also it will provide further assistance to the tracker. In this regard, the remainder of this chapter will firstly present a brief review of TSI based detection method. Then, the multiple TSI will be introduced to explain how it overcomes the limitation of single TSI based detection approach. The coordinates of vehicular object in KVFs obtained using MTSI method will be used not only to initialize the tracking in a video but also to correct the tracking trajectories of the Kalman or Particle filter-based algorithms shown in Chapter 3.

2.2 TSI for Vehicle Detection

Time Spatial Image is generated by placing the pixel strips of the frames on the virtual detection line (VDL) in a chronological order. VDL is in fact a set of indexes whose position is usually perpendicular to the motion of the vehicles and is independent of the frames. An example of a VDL on a few consecutive frames of a video-sequence and the corresponding TSI for the sequence are shown in Figures 2.1 and 2.2, respectively. It can be seen from these figures that each of the vehicle passing the VDL corresponds to an object in TSI. These objects are referred to as TSI object blobs (TOBs). Hence, the number of vehicles in traffic may be determined by counting these TOBs. It is to be noted that the horizontal length or width of an object in TSI corresponds to the number of frames in which the corresponding vehicle was on the VDL. Thus, the width of TOB is related to the length and speed of vehicle. On the other hand, the vertical length or height of the object in TSI corresponds to the breadth of the vehicle. In this way, TOBs include the characteristics of both the temporal and spatial features of the vehicles passing the VDL.

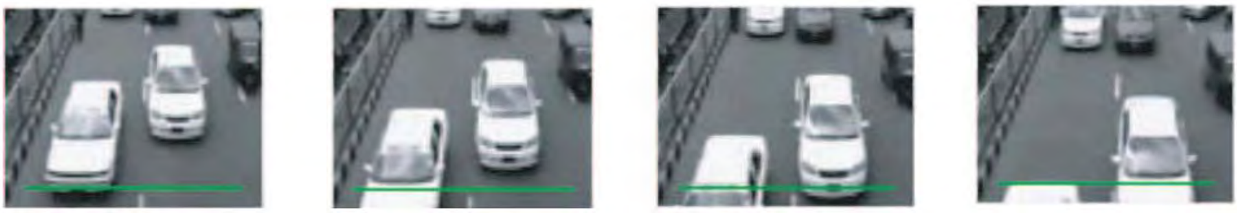


Figure 2.1: VDL on a few consecutive frame of a typical video sequence

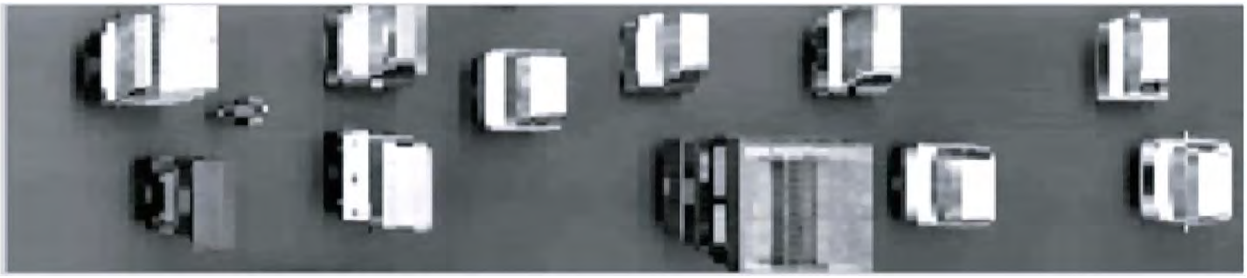


Figure 2.2: TSI generated from a VDL wherein each object corresponds to a vehicle.

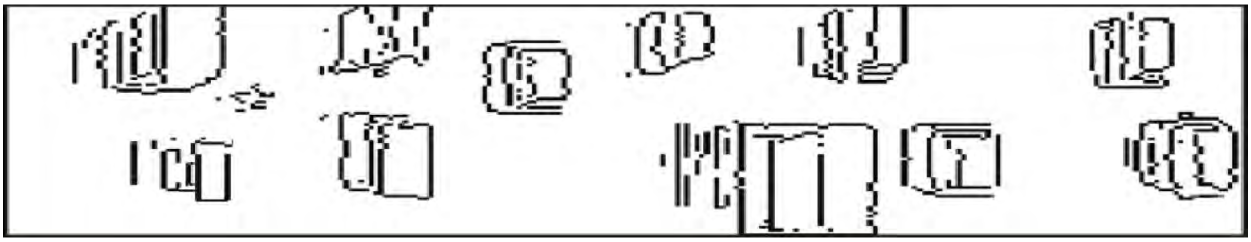


Figure 2.3: The edge of the objects obtained from Canny edge detector.

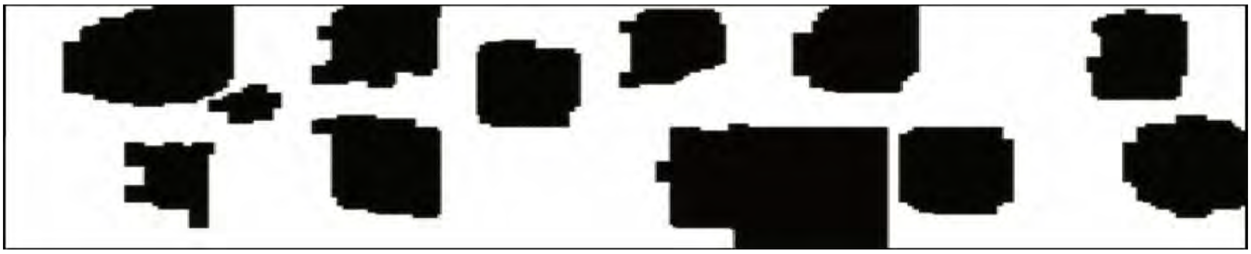


Figure 2.4: Binary masks of the TOBs are obtained using two consecutive morphological closing operations, first with *line* object L_5 and then with *square* object S_7.

In order to extract the features of TOB, a binary image mask is created from the TSI. For this purpose, Canny edge detector is used at first, for detecting fine details of the objects in TSI. Then, morphological erosion and dilation operations are performed with suitable objects to obtain a segmented region correspond to each of the objects in a TSI. Figures 2.3 and 2.4 show the edge detected and region of binary mask obtained, respectively, from the

TSI shown in Figure 2.2. A number of shape-features may be obtained from the objects of TSI by using this kind of binary mask for the purpose of vehicle classification.

To obtain the coordinates of a vehicle in a frame, it is necessary to first find the KVF wherein each of the vehicles captured in the video should have the same distance from the camera. In this method, the key frame KVF is selected as the midpoint of the width of a TOB, which corresponds to the frame in which the midpoint of the vehicle is approximately on the VDL. The appearances of some of the vehicles in the KVFs, as shown in Figure 2.5, are such that the shape and texture of the vehicles are comparable on the same scale and do not depend on the speed of the vehicle. This is why only the KVFs are used to obtain the features of the vehicular object. Since the TSI-based vehicle method uses only the KVFs for detecting vehicle, the computational complexity of the algorithm is significantly lower compared to other methods.

To obtain the region of interest (ROI) from the KVF, the values of the center and width of the corresponding TOB in TSI is used. The vertical strip of width $(d_2 - d_1)$ is cropped using the width of TOB from the KVF. To find the width of horizontal strip, the absolute difference of the vertical strips of KVF and KVF $+$ 2 is used. An example showing the process of obtaining the silhouette of a vehicle „car" is shown in Figure 2.6.

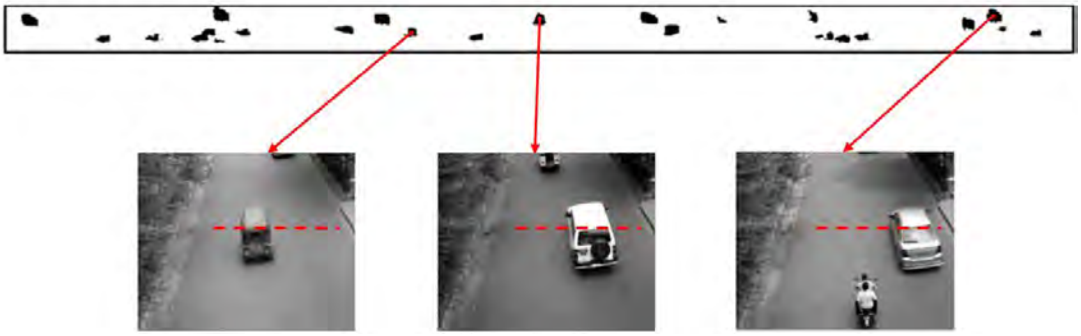
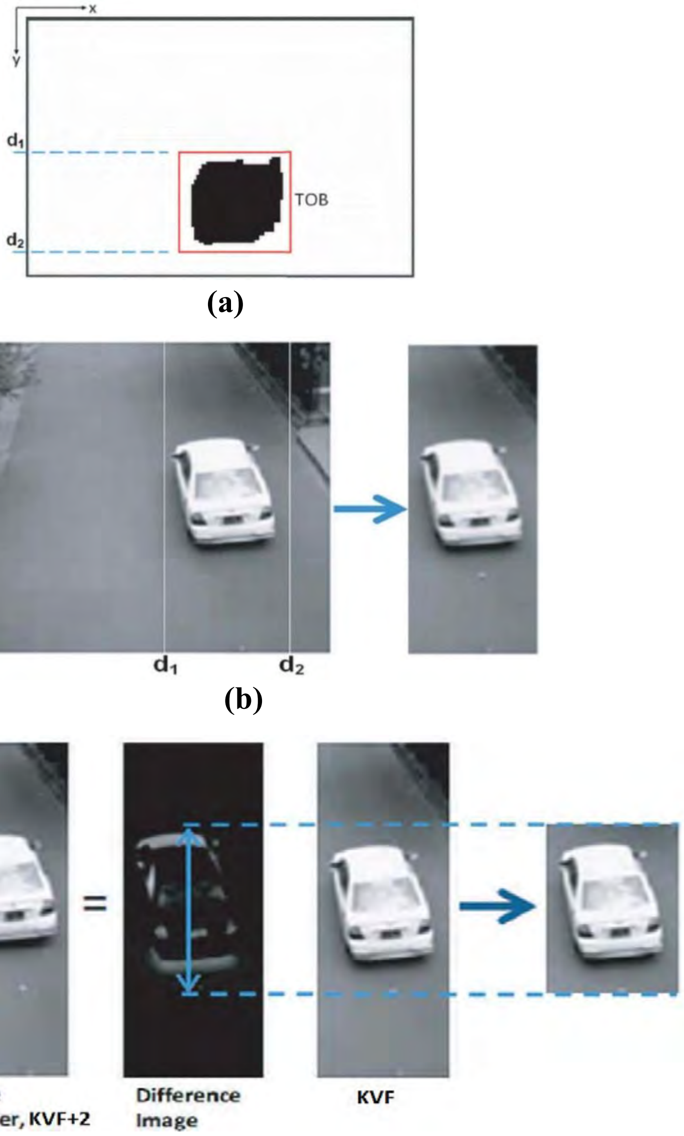


Figure 2.5: Example of Vehicle on KVFs



(a)

(b)

(c)

Figure 2.6: Obtaining the ROI from the KVF. (a) Center and Width of TOB in TSI provide the y-axis coordinates $\boldsymbol{d_1}$and $\boldsymbol{d_2}$ $(\boldsymbol{d_2} > \boldsymbol{d_1})$, between which the vehicle passes. (b) A vertical strip of width $(\boldsymbol{d_2} > \boldsymbol{d_1})$is cropped. (c) The frame difference provides the width to be cropped for the horizontal strip.

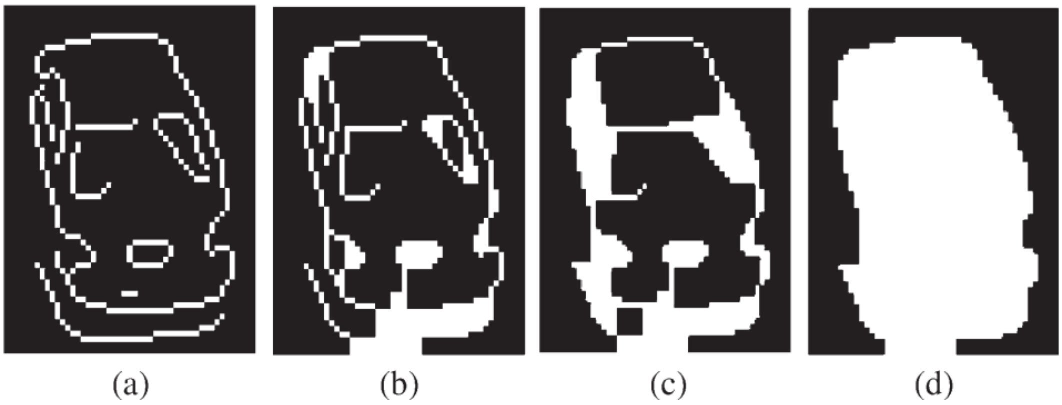


Figure 2.7: Obtaining the silhouette from the ROI of the vehicle. (a) Edge detected. (b) Image obtained from the closing operation on (a) with object $\boldsymbol{L_4}$. (c) Image obtained from the closing operation on (b) with object $\boldsymbol{S_5}$ (d) KVB obtained after removing the unwanted region inside ROI.

A segmentation process is necessary to obtain the silhouette from the ROI that describes the shape and texture of the vehicle concerned. This segmentation process is similar to the generation of binary masks of TOBs in a TSI. The edges of the ROI found from the edge detector are often very discontinuous. To connect the disjoint edges, two successive closing operations using the line object L_λ and square object S_μ are done. The resulting binary mask is referred to as the key vehicular blob (KVB), using which the features are extracted for the purpose of classification. The values of parameters λ and μ need to be chosen based on environment. The KVB obtained from a typical silhouette of a vehicle "car" using $\lambda = 4$ and $\mu = 5$ is shown in Figure 2.7. From this figure, it is evident that the KVB maintains the apparent shape of the vehicle considered in the example.

2.3 Multiple TSI for Vehicle Detection

The primary reason for introducing Multiple TSI based detection is to minimize the occurrence of misdetection and misclassification, mainly due to merging between TOBs. Merging between TOBs are found when occlusion is seen between two or a higher number of neighboring vehicles because of viewing angle or the closely spaced TOBs become connected because of morphological dilation operation. Hence, in the MTSI method, a number of VDLs are used to generate TSI with the consideration that each of the vehicles on a road successively passes all the VDLs in a uniflow direction, as shown in Figure 2.8. This method improves detection performance by successfully identifying the vehicles, whose TOBs are merged in certain TSIs and remain disjoint in others. The improvement in classification is also achieved by using multiple TOBs in classification for a single vehicle.

The VDLs on uniflow traffic on roads need to be chosen at such distances that the vehicles of the traffic fully remain in the scene, the chance of occlusion among vehicles is low, and the lateral movements of the vehicles to the flow direction are approximately zero. In general, the chance of occlusion of a vehicle by another vehicle increases with the increasing distance between the view line on the road defined by a VDL and the camera position. For each of the VDLs to cover almost the same breadth of the road through which the vehicles pass, the length of a VDL decreases as the distance between the view lines on the road defined by the VDL and the camera position increases.

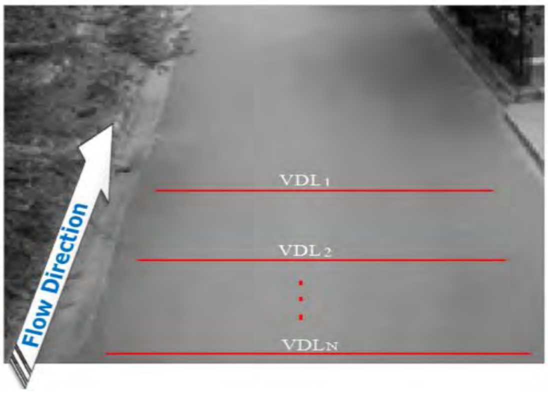
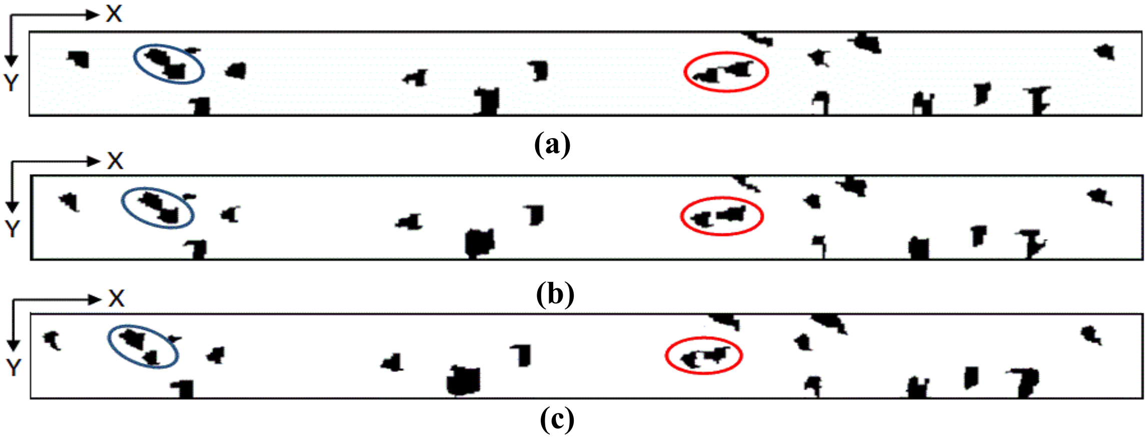


Figure 2.8: Frame showing a few number of VDLs perpendicular to the traffic flow



(a)

(b)

(c)

Figure 2.9: Section of the scaled version of multiple TSIs generated from MVDLs. (a)TSI$_1$, (b)TSI$_2$, (c)TSI$_3$

Let D_n $(n = 1,2,...,N)$ be the lengths of the VDL_{n} $(n = 1,2,...,N)$ in terms of pixels. To process each of the TSIs on the same scale, the TSIs are generated by placing the up- or downsampled version of the pixel strips of the frames on the VDLs, so that the height of the TSIs remains same. The scaling factors for up- or downsampling are determined from the ratio between D_n and D_v, where D_v is the height selected for the TSIs. Let the scaled version of the TSIs denoted as TSI_n $(n = 1,2,...,N)$ be generated from the corresponding VDL_n $(n = 1,2,...,N)$. The processing of the scaled version of the TSIs reduces the effects of environmental factors such as the size of vehicles on the road, elevation of the camera, and video resolution on the choice of position of VDLs. A section of the binary masks generated for the objects in such a scaled version of the TSIs for $N = 3$, where the selected height is D_1, is shown in Figure 8. It may be observed from this figure that the TSIs have similar heights and that a vehicle leaves a delayed version of similar TOBs on the successive TSIs. It is also observed from this figure that some of the TOBs are merged in certain TSIs and remain disjoint in others. For example, the TOBs corresponding to the second and third vehicles from the left of the TSIs become merged in TSI_1 and TSI_2, whereas they are disjoint in TSI_3. The MTSI-based method shows improved detection performance than single TSI based method. In practice, the chance of merging of TOBs are reduced by increasing the elevation of the camera or reducing the size of the line and square objects in the morphological operations for segmentation of the objects in a TSI. In general, by identifying the merged TOBs and adjusting the total count of TOBs, detection performance may be significantly improved. To detect possible merging, the y-axis co-ordinate of the centroid $\mathcal{C}_n^y$, the width along both x- and y-axis $\mathcal{W}_n^x$, and $\mathcal{W}_n^y$, and the area $\mathcal{A}_n (n = 1,2,\ldots,N)$ of each of the TOBs of the TSIs are estimated. Since the MVDLs are close to each other in the direction of traffic flow, the value of $\mathcal{C}_n^y$, $\mathcal{W}_n^y$ and $\mathcal{A}_n$ for a distinct TOB should not significantly change with variation of n. A significant change in these parameters indicates that the TOBs have merged side by side in the direction of traffic flow. In a similar manner, change in parameters $\mathcal{W}_n^x$ and $\mathcal{A}_n$ indicates that the TOBs have merged back to front in the direction of traffic flow. Let $\mathcal{N}$ be a 2-D array of size $N \times \acute{\mathrm{M}}$ that keeps the trace of whether the TOBs corresponding to m number of distinct vehicles are disjoint or merged by placing the element as zero or one, respectively. Here, $\acute{\mathrm{M}}$ indicates the total number vehicle in the sequence. This array is generated from an iterative process by starting from the left of the TSIs and comparing the parameters previously mentioned. Let $\mathcal{M}$ be a 1-D array containing N rows to track the position of TOBs on different TSIs traced from the left. To estimate the elements of arrays $\mathcal{N}$ and $\mathcal{M}$, three major steps are followed in each iteration.

Step 1: In this step, a search region is found to compare a TOB in a TSI corresponding to a vehicle with that in other TSIs. It is to be noted that a TSI is formed using the pixels on a VDL of each of the frames; hence, thex-axis of aTSI corresponds to the frame number of the video. Let Δx represent the difference in pixel position between two successive VDLs in a frame of a video. Thus, if a vehicle in the video passes two successive VDLs, then the vehicle has actually passed the physical distance $\Delta x R_p$ meter, where R_p is the resolution of the video in meters per pixel. In such a case, the speed of the vehicle in meters per second may be estimated from TSIs corresponding to two successive VDLs as

$$V = \Delta x R_p F_r / N_f \tag{1}$$

Where N_f is the number of frames required for a vehicle to pass two successive VDLs, and F_r is the frame rate of the video. Let V_{max} and V_{min} be the maximum and minimum speed limits, respectively, of the traffic in video. Let a TOB corresponding to a vehicle exist on TSI_n $(n > 1)$ in such a position that the lower and upper limits of the x-axis coordinate of the TOB are X_l and X_u $(X_u > X_l)$, respectively. In such a case, the search region S along the x-axis of the TOB corresponding to the same vehicle in TSI_{n-1} $(n > 1)$ will be between $X_l + \delta_f^{min}$ and $X_u + \delta_f^{max}$, wherein

$$\delta_f^{min} = \Delta x R_p F_r / V_{max} \tag{2}$$

$$\delta_f^{max} = \Delta x R_p F_r / V_{min} \tag{3}$$

It is to be noted that the VDLs are chosen in such a way that the lateral movements of the vehicles to the traffic flow are considerably small. Hence, the changes in the position of a TOB along they-axis among the TSIs are very insignificant. As a consequence, the search region S along the y-axis for the TOB in TSI_{n-1} $(n > 1)$ will be between $\mathcal{C}_n^y - \gamma \mathcal{W}_n^y$ and $\mathcal{C}_n^y + \gamma \mathcal{W}_n^y$, where γ is chosen to be as small as 0.1. This way, search region S ensures that a given TOB in TSI_n has only one corresponding TOB in $TSI_{n-1}(n > 1)$.

Step 2: This step finds out the dissimilarity among N number of TOBs, each obtained from a TSI using the search region S defined in Step 1. The dissimilarity among the TOBs is estimated using the differences of four shape-based features, viz., $\mathcal{C}_n^y, \mathcal{W}_n^x, \mathcal{W}_n^y$, and $\mathcal{A}_n$. The logical value of the differences is found to be

$$\mathcal{T}_1 : \frac{|\mathcal{C}_l^y - \mathcal{C}_n^y|}{|\mathcal{C}_n^y|} > T_{\mathcal{C}}^y \qquad (4)$$

$$\mathcal{T}_2 : \frac{|\mathcal{W}_l^y - \mathcal{W}_n^y|}{|\mathcal{W}_n^y|} > T_{\mathcal{W}}^y \qquad (5)$$

$$\mathcal{T}_3 : \frac{|\mathcal{W}_l^x - \mathcal{W}_n^x|}{|\mathcal{W}_n^x|} > T_{\mathcal{W}}^x \qquad (6)$$

$$\mathcal{T}_4 : \frac{\mathcal{A}_l - \mathcal{A}_n}{\mathcal{A}_n} > T_{\mathcal{A}} \qquad (7)$$

Where, $(l = 1,2,\ldots,N)$ $(n = 1,2,\ldots,N)$ $(l \neq n)$; and $T_{\mathcal{C}}^y$, $T_{\mathcal{W}}^y$, $T_{\mathcal{W}}^x$, and $T_{\mathcal{A}}$ are the threshold parameters that may be dependent on the illumination, vehicle size, and video resolution. For example, a higher threshold value may be chosen when shadow in a highly illuminated environment affects the segmentation of the vehicle. In the experiments, it is found that the values of the threshold parameters lie between 5% and 15%. Since a merging of TOBs may occur either side by side or back to front, conditions $\mathcal{T}_2$ and $\mathcal{T}_3$ may not be true at the same time. Hence, the TOBs are considered merged when condition $\mathcal{T}$ is true, wherein

$$\mathcal{T} : \mathcal{T}_1 \wedge (\mathcal{T}_2 \vee \mathcal{T}_3) \wedge \mathcal{T}_4 \qquad (8)$$

Step 3: In this step, arrays $\mathcal{N}$ and $\mathcal{M}$ are updated depending on whether the TOBs are merged or not, the decision of which is obtained from Step 2. If, in a given iteration, the TOBs are found to be disjoint, then one column of zeros is appended to $\mathcal{N}$, and elements of $\mathcal{M}$ are incremented by one. On the other hand, when merging occurs on TOBs, it is crucial to detect the number of merged TOBs referred to as n_{TOB}. Since various shapes and sizes of

TOBs exist in a TSI, it is intuitive that n_{TOB} may be obtained as the number of disjoint TOBs for which the added values of the features, i.e., area and widths, are approximately equal to those of the merged TOB. Let the widths and area of a merged TOB in TSI_n be denoted as $\mathcal{W}^{x1}$, $\mathcal{W}^{y1}$, and $\mathcal{A}^1$, respectively, and those remaining disjoint in the search region S of TSI_l $(l \neq n)$ as $\mathcal{W}_n^{x0}$, $\mathcal{W}_n^{y0}$, and $\mathcal{A}_n^0$ respectively. In such a case, the number of TOBs that are merged in TSI$_n$ may be estimated from search region S of TSI$_l$ as

$$n_{\mathrm{TOB}l\,|\,\mathcal{W}^x} = \arg max_n \left(\left\lfloor \frac{\mathcal{W}^{x1}}{\sum_{n\in S} \mathcal{W}_n^{x0}} + \frac{1}{2} \right\rfloor = 1 \right) \tag{9}$$

$$n_{\mathrm{TOB}l\,|\,\mathcal{W}^y} = \arg max_n \left(\left\lfloor \frac{\mathcal{W}^{y1}}{\sum_{n\in S} \mathcal{W}_n^{y0}} + \frac{1}{2} \right\rfloor = 1 \right) \tag{10}$$

$$n_{\mathrm{TOB}l\,|\,\mathcal{A}} = \arg max_n \left(\left\lfloor \frac{\mathcal{A}^{1}}{\sum_{n\in S} \mathcal{A}_n^{0}} + \frac{1}{2} \right\rfloor = 1 \right) \tag{11}$$

$$n_{TOB} = mode\left(n_{\mathrm{TOB}l\,|\,\mathcal{W}^y},\, n_{\mathrm{TOB}l\,|\,\mathcal{W}^x}, n_{\mathrm{TOB}l\,|\,\mathcal{A}}\right) \tag{12}$$

where $\lfloor z \rfloor$ denotes the largest integer contained in z, and mode (•) indicates the integer number whose occurrence is maximum in (•). In the event of a tie, precedence is given to $n_{\mathrm{TOB}l\,|\,\mathcal{A}}$, $n_{\mathrm{TOB}l\,|\,\mathcal{W}^y}$, and $n_{\mathrm{TOB}l\,|\,\mathcal{W}^x}$ with consideration that the TOBs corresponding to vehicles are most likely to be close side by side, compared to back to front. When the TOBs become merged, the 2-D array $\mathcal{N}$ is updated by appending n_{TOB} number of columns, wherein the elements for the n^{th} row become ones and the rest become zeros. It is to be noted that a TOB corresponding to a vehicle in a given TSI may be merged and, at the same time, may remain disjoint in other TSIs. In such a case, the elements of array $\mathcal{M}$ are incremented only by the number of disjoint TOBs.

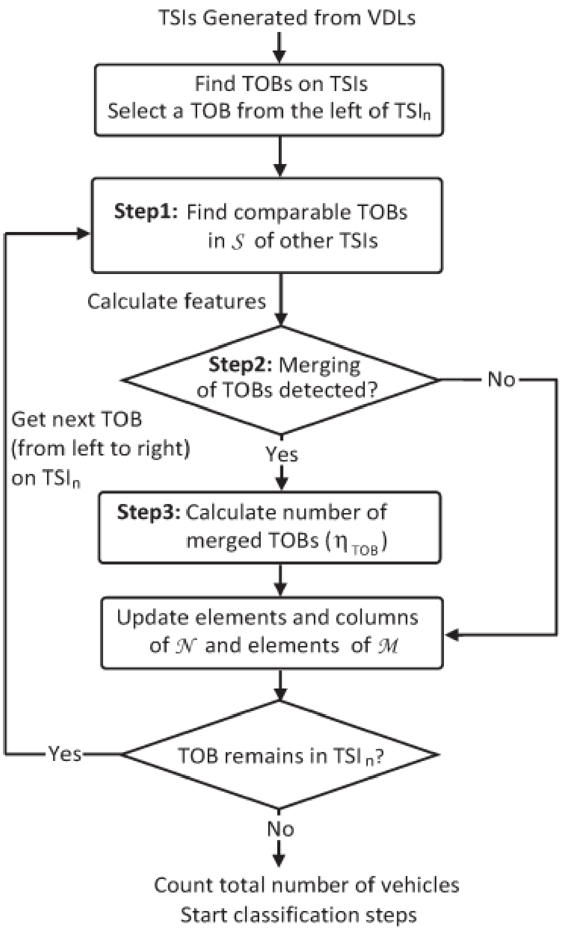
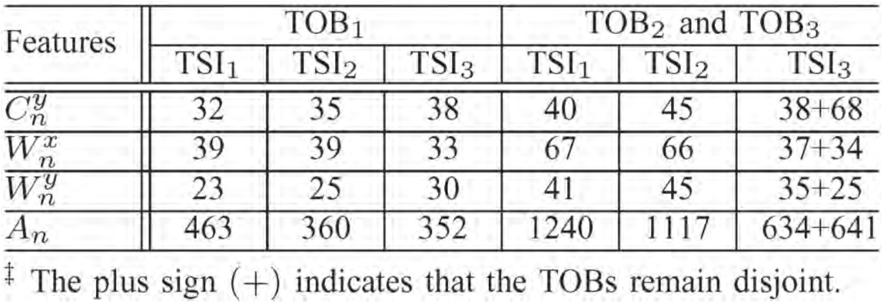

TSIs Generated from VDLs
Find TOBs on TSIs
Select a TOB from the left of TSIn
Step1: Find comparable TOBs in 𝒮 of other TSIs
Calculate features
Step2: Merging of TOBs detected?
No
Get next TOB (from left to right) on TSIn
Yes
Step3: Calculate number of merged TOBs (η TOB)
Update elements and columns of 𝒩 and elements of ℳ
Yes
TOB remains in TSI n?
No
Count total number of vehicles
Start classification steps

Figure 2.10: Block diagram of the proposed detector

Table 2.1: RESULTS CONCERNING THE CENTROIDS, WIDTHS, AND AREAS OF THE FIRST THREE OBJECTS FROM THE LEFT OF THREE TSIs ‡

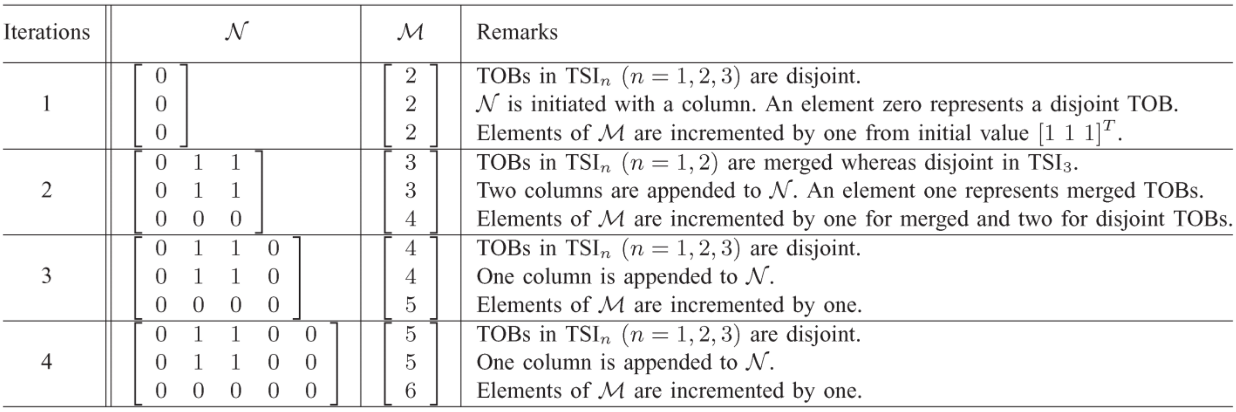


Features	TOB$_1$			TOB$_2$ and TOB$_3$		
	TSI$_1$	TSI$_2$	TSI$_3$	TSI$_1$	TSI$_2$	TSI$_3$
C_n^y	32	35	38	40	45	38+68
W_n^x	39	39	33	67	66	37+34
W_n^y	23	25	30	41	45	35+25
A_n	463	360	352	1240	1117	634+641

‡ The plus sign (+) indicates that the TOBs remain disjoint.

It is clear that the number of columns on the 2-D array 𝒩 provides the number of vehicles that pass through the VDLs. If, for any row, the elements of a given column are one, it indicates that the TOBs corresponding to the vehicles are merged, and zero elements represent the TOBs remaining disjoint. A simple block diagram of the proposed vehicle-counting process is shown in Figure 2.10. From this figure, it can be seen that an iterative process that follows Steps 1-3 and updates $\mathcal{N}$ and $\mathcal{M}$ accordingly in each iteration is sufficient for detection of vehicles. We would provide an example showing how arrays $\mathcal{N}$ and $\mathcal{M}$ are updated, provided that the TSIs have merged and disjoint TOBs. Let us consider only those TOBs from the left of the three TSIs shown in Figure 2.9 that correspond to the first five vehicles. It can be seen from the figure that the TOBs corresponding to the first, fourth, and fifth vehicles appeared in the TSIs that are disjoint, whereas the TOBs corresponding to the second and third vehicles are merged in TSI_1 and TSI_2 and remain disjoint in TSI_3. Since the example considers three TSIs, there are three rows in arrays $\mathcal{N}$ *and* $\mathcal{M}$ and are initialized by an empty matrix and $[1\ 1\ 1]^T$, respectively. The y-axis coordinate of the centroids, widths, and areas of the first three TOBs from the left (which are referred to as TOB_1, TOB_2, and TOB_3, with the last two being merged in the first two TSIs) are given in Table 2.1. It can be seen from this table that the y-axis coordinate of the centroids, widths, and areas are significantly deviated when individual objects such as TOB_2 and TOB_3 in TSI_3 are merged in the other two TSIs. On the other hand, these features have closeness in values, while the TOBs remain separated in the TSIs. It is also noted that the sum of the x-axis widths and areas of the two disjoint TOBs in TSI_3 are approximately equal to those of the merged TOB in TSI_1 or TSI_2, which indicates that the number of merged TOBs is two. The elements of the 2-D array $\mathcal{N}$ and 1-D array $\mathcal{M}$ estimated for the first four iterations by using the proposed detection algorithm are shown in Table 2.2. It can be seen from this table that the five columns of $\mathcal{N}$ represent the five vehicles that have passed the VDL, and the second and third columns of the first and second rows show that the second and third vehicles have been merged in TSI_1 and TSI_2.

Table 2.2: RESULTS CONCERNING THE ELEMENTS OF N AND M FOR THE FIRST FOUR ITERATIONS FOR THE CLIP CORRESPONDING TO FIGURE 2.6



Iterations	$\mathcal{N}$	$\mathcal{M}$	Remarks
1	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$	TOBs in TSI$_n$ $(n = 1, 2, 3)$ are disjoint. $\mathcal{N}$ is initiated with a column. An element zero represents a disjoint TOB. Elements of $\mathcal{M}$ are incremented by one from initial value $[1\ 1\ 1]^T$.
2	$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 3 \\ 4 \end{bmatrix}$	TOBs in TSI$_n$ $(n = 1, 2)$ are merged whereas disjoint in TSI$_3$. Two columns are appended to $\mathcal{N}$. An element one represents merged TOBs. Elements of $\mathcal{M}$ are incremented by one for merged and two for disjoint TOBs.
3	$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 4 \\ 4 \\ 5 \end{bmatrix}$	TOBs in TSI$_n$ $(n = 1, 2, 3)$ are disjoint. One column is appended to $\mathcal{N}$. Elements of $\mathcal{M}$ are incremented by one.
4	$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 5 \\ 5 \\ 6 \end{bmatrix}$	TOBs in TSI$_n$ $(n = 1, 2, 3)$ are disjoint. One column is appended to $\mathcal{N}$. Elements of $\mathcal{M}$ are incremented by one.

2.4 Conclusion

In this chapter, multiple VDL based MTSI detection approach has been discussed in detail. Using this method, the spatial correspondence in terms of coordinates of a vehicle is identified on KVFs corrensponding to the virtual detection lines. It is expected that, the center coordinate position obtained from detection mechanism will not only help to start tracking, but also will help to track vehicles in some key positions of road, which can later be used as corrective tool for other tracking methods.

3. MTSI-BASED TRACKING ALGORITHMS

3.1 Introduction

In Chapter 2, it has been mentioned that the vehicular object in KVFs obtained using MTSI method can be very helpful in obtaining a very accurate position of the vehicle in the key frames. In this chapter, first we present computationally efficient very fast tracking algorithm using the vehicular coordinate positions obtained from the MTSI described in Chapter 2. Then, we describe the Kalman and Particle filter-based tracking algorithms those are commonly referred to video-based tracking literature. Finally, we present a high performance tracking algorithm, which improves reliability of tracking by incorporating MTSI based detection method into the Kalman or Particle filter-based tracking methods.

3.2 Low-Complexity MTSI-Based Tracking

In Chapter 2, it has been shown that comparable TOBs for a vehicle can be identified on different TSI using that MTSI based detection method. Multiple TOBs for a single vehicle provides multiple KVFs, where the positions of the vehicle are known. As we accurately know the position of vehicular object on different key frames, the approximate trajectory of vehicular movement can be estimated by interpolating the coordinates. The accuracy of estimated trajectory will be increased with increasing the number of VDLs, as more data will then be used to estimate the trajectory. This trajectory will not give an accurate trajectory as the estimated trajectory between the VDLs depends on interpolator.

In this approach, at first we need to choose the interpolator for constructing new data points within the range of a discrete set of known center data points in KVFs. As there is usually very little lateral movement of vehicle between VDLs and also the speed is almost constant between VDLs, a linear interpolator or a polynomial interpolator with a lower degree may give a good estimate.

The simplest method is linear interpolation. Generally, linear interpolation takes two data points, say (x_a, y_a) and (x_b, y_b), in the Cartesian coordinate system and the interpolant is given by:

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a} \text{ at the point } (x, y) \qquad (3.1)$$

If there is no lateral movement of vehicle and speed is fully constant, then using a linear interpolator will give good result. Linear interpolation is quick and easy but the chance of getting a precise result is low in real world case. However, nonlinear trajectory estimation may improve the tracking accuracy, but at the expense of computational complexity. In such a case, the polynomial interpolator with degree n ($n \in \{2,3,4,5\}$) may be applied to obtain a sufficiently accurate trajectory. The process for constructing the interpolation polynomial of degree *n* is given below.

Given a set of N data points (x_n, y_n) corresponding to N TSIs, polynomial p of degree at most *N-1* can be formed with the property

$$p(x_n) = y_n, i = 0, \ldots.., N\text{-}1$$

Suppose that the interpolation polynomial of degree $N - 1$ is in the form

$$p(x) = a_{N-1}x^{N-1} + a_{N-2}x^{N-2} + \cdots + a_2x^2 + a_1x + a_0 \qquad (3.2)$$

The statement that p interpolates the data points means that

$$p(x_n) = y_n \quad \text{for all } i \in \{0,1, \ldots.., N - 1\}$$

If we substitute equation (3.2) in here, we get a system of linear equations in terms of the coefficients a_n. Now, the system in matrix-vector form

$$\begin{bmatrix} x_0^{N-1} & x_0^{N-2} & x_0^{N-3} & \cdots & x_0 & 1 \\ x_1^{N-1} & x_1^{N-2} & x_1^{N-3} & \cdots & x_1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{N-1}^{N-1} & x_{N-1}^{N-2} & x_{N-1}^{N-3} & \cdots & x_{N-1} & 1 \end{bmatrix} \begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \cdot \\ \cdot \\ \cdot \\ a_0 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_{N-1} \end{bmatrix}$$

We have to solve this system for a_n to construct the interpolant $p(x)$.

We used Lagrange polynomial to solve the problem. The polynomial may be written down immediately in terms of Lagrange polynomials:

$$p(x)=\frac{(x-x_1)(x-x_2)\ldots.(x-x_{N-1})}{(x_0-x_1)(x_0-x_2)\ldots.(x_0-x_{N-1})}.y_0+\frac{(x-x_0)(x-x_2)\ldots.(x-x_{N-1})}{(x_1-x_0)(x_1-x_2)\ldots.(x_1-x_{N-1})}.y_1$$

$$+\cdots+\frac{(x-x_0)(x-x_1)\ldots.(x-x_{N-2})}{(x_n-x_0)(x_n-x_1)\ldots.(x_N-x_{N-2})}.y_{N-1}.$$

That is,

$$p(x)=\sum_{n=0}^{N}y_n\prod_{0\leq m\leq N-1,m\neq n}\frac{x-x_m}{x_n-x_m} \qquad (3.3)$$

So, $p(x)$ polynomial can be formed using the known n number of given center points. From this $p(x)$, the coefficients of equation (3.2) can be calculated. Here, since the coordinate positions are estimated from the KVBs only, the forward and backward tracking trajectories will be the same.

3.3 High Performance Tracking: MTSI Integrated with Particle Filter or Kalman Filter

In this high performance approach, the tracking procedure for each vehicle will be started automatically using the coordinate of center position of the vehicle in the KVF corresponding to the first VDL [40]. Then the tracking will be continued in successive frames using a Monte-Carlo Particle filter tracker [17], [41] or Kalman filter tracker [18]. A tracking coordinate correction will go on in parallel, which will only be active in KVFs using vehicular coordinates obtained in KVFs using MTSI detection method. Here, the procedure for tracking a single vehicle in successive frames using Kalman filter tracker will be discussed in Section 3.3.1. The procedure for tracking a single vehicle in successive frames using Monte Carlo Particle filter tracker will be discussed in Section 3.3.2. Then, tracking correction in KVFs method for the vehicle will be described in Section 3.3.3.

3.3.1 Tracking in Succesive Frames Using Kalman Filter

In this section, Kalman filtering algorithm will be described for tracking vehicular object in successive frames. The Kalman filter is a combination of a predictor and a filter. Both the predictor and the filter are linear systems. A constant velocity model is assumed for our target. If we assume a constant velocity model for our target, the Kalman filter reduces to the alpha-beta filter. Here, the predictor estimates the location of the target at $(i+1)$th frame given ith previous frame observations. When observation $i+1$ arrives, the estimate is improved using an optimal filter to estimate the target position at frame $i+1$. The filtered estimate is the final estimate of the true location of the target att frame $i+1$.

Initially, the vehicle center is identified for tracking using TSI based detection. This center coordinate will be fed into the tracker as the first predicted coordinate estimate. Additionally, velocity in X-direction (v_x) and velocity in Y-direction (v_y) is also needed to be initialized to start the tracking. Since in vertical direction, Y-coordinate is decreasing in the direction of vehicular movement, the proposed tracking methods initialize v_y as -1 for forward tracking and v_y as 1 for backward tracking. On the other hand, v_x is initialized as 1 for both cases.

Here, for tracking, two Kalman filters are formed for prediction/estimation: one for the horizontal position and the other for the vertical position of the vehicular-object center. Let, C_i^x and C_i^y be the horizontal coordinate of center and vertical coordinate of center of a vehicle at video frame i.

Then, the predicted center position at $(i+1)$th frame $(C_{i+1|i}^{\bar{x}}, C_{i+1|i}^{\bar{y}})$ is estimated based on track history

$$C_{i+1|i}^{\bar{x}} = C_i^x + \delta t \bar{v}_{i+1|i}^x \,,$$

$$C_{i+1|i}^{\bar{y}} = C_i^y + \delta t \bar{v}_{i+1|i}^y \qquad (3.4)$$

Where δt is the time between frames, obtained from framerate and $\bar{v}^{x}_{i+1|i}$ and $\bar{v}^{y}_{i+1|i}$ is the predicted velocity in X-direction and Y-direction respectively, at $(i + 1)$th frame. Here, initially C^{x}_{0} and C^{y}_{0} is found from the vehicular-object identified using TSI based detection method. The goal of the prediction portion is to estimate the next position of the target, given the previously acquired image sequence.

Once we have an estimate of the next target position, we look at a small subimage for the target and avoid searching the whole image again. The Observed position at $(i + 1)$th frame $(C^{\hat{x}}_{i+1}, C^{\hat{y}}_{i+1})$ is found by calculating center of mass(intensity centroid) of a small sub-image region, centered at the predicted position $(C^{\bar{x}}_{i+1|i}, C^{\bar{y}}_{i+1|i})$. This region is termed as track gate, which depends on height and width of KVB identified using TSI based detection method.

Finally, the filtered center position is calculated as

$$C^{x}_{i+1} = \alpha_i C^{\hat{x}}_{i+1} + (1 - \alpha_i) C^{\bar{x}}_{i+1|i} \, ,$$

$$C^{y}_{i+1} = \alpha_i C^{\hat{y}}_{i+1} + (1 - \alpha_i) C^{\bar{y}}_{i+1|i} \tag{3.6}$$

Where, $\alpha_i (0 < \alpha_i < 1)$ a gain is determined by the Kalman filter.

The predicted velocity is given by

$$\bar{v}^{x}_{i+1|i} = \bar{v}^{x}_{i|i-1} + \beta_i (C^{\hat{x}}_{i} - C^{\bar{x}}_{i|i-1}) / \delta t, \tag{3.7}$$

$$\bar{v}^{y}_{i+1|i} = \bar{v}^{y}_{i|i-1} + \beta_i (C^{\hat{y}}_{i} - C^{\bar{y}}_{i|i-1}) / \delta t, \tag{3.7}$$

Where β_i $(0 < \beta_i < 1)$ is another Kalman gain.

The gains α_i and β_i depend upon the noise variances and the state vector error covariance matrix. Here, α_i determines the balance between the previous track history and the new observation. If α_i is large (near 1), it is believed that the observations are very reliable (this is essentially ignoring the track history). On the other hand, If it is small (near 0), it is believed that there is a lot of measurement noise (this is essentially ignoring the observation). So, the gain α_i in (3.6) is increased in order to give more weight to the observations and less weight to the predictions as the track commences.

The other Kalman gain β_i controls how new observation affects the predicted velocity. If it is near 0, it means that it is believed that the observations are unreliable and that the actual velocity is really a constant. In this case, the observation does not affect the predicted velocity. However, if β_i is near 1, then the observations are reliable. Here, a drift (acceleration) in velocity is expected.

3.3.2 Tracking in Succesive Frames Using Particle Filter

In this section, a modified Particle filtering algorithm for tracking vehicular object in successive frames is described. In frame $i+1$, let X_{i+1} be the target state and Z_{i+1} be the observed image state. In practice, the stochastic state transition model $p(X_{i+1}|X_i)$ is assumed to be a Markov chain. The observation model $p(Z_{i+1}|X_i)$ is defined to make the maxima in $p(Z_{i+1}|X_{i+1})$ correspond to the measured image intensity features. The density of $p(Z_{i+1}|X_{i+1})$ (where $Z_{1:i+1} = \{Z_1, Z_2, \dots, Z_{i+1}\}$) can be propagated according to Condensation algorithm developed by Isard and Blake [14] :

$$p(X_{i+1}|Z_{1:i+1}) = k_{i+1} p(Z_{i+1}|X_{i+1})\, p(X_{i+1}|Z_{1:i}) \qquad (3.8)$$

Where, $p(X_{i+1}|Z_{1:i}) = \int p(X_{i+1}|X_i)\, p(X_i|Z_{1:i}) \mathrm{d}X_i$, (3.9)
And k_{i+1} is a normalizing constant.

Based on the factored sampling [42], [43], [45], the Condensation algorithm approximates $p(X_{i+1}|Z_{1:i+1})$ by a sample set $\{s_{i+1}^{(m)}, \pi_{i+1}^{(m)}\}$, where $s_{i+1}^{(m)}$ is the sample m, $\pi_{i+1}^{(m)}$ is its weight, and $m = 1,2,\dots,M,$ where M is the sample size,. Suppose the sample set $\{s_i^{(m)}, \pi_i^{(m)}\}$ approximates $p(X_i|Z_{1:i})$. We generate $\{s_{i+1}^{(m)}, \pi_{i+1}^{(m)}\}$ to approximate $p(X_{i+1}|Z_{1:i+1})$ using following steps:

1. Resample the set $\{s_i^{(m)}, \pi_i^{(m)}\}$ from the observation by drawing a sample $s_{i+1}^{(m)}$ such that the samples have a prediected probability $\pi_i^{(m)}, m = 1,2,\ldots, M$;
2. Draw a sample $s_{i+1}^{(m)}$, from the state transition model $p(X_{i+1}|X_i = s_{i+1}^{(m)})$, m =1, 2,...,M;
3. Weight $s_{i+1}^{(m)}$ by the image intensity observation

$$\pi_{i+1}^{(m)} = \frac{p(Z_{i+1}|X_{i+1} = s_{i+1}^{(m)})}{\sum_{j=1}^{M} p(Z_{i+1}|X_{i+1} = s_{i+1}^{(j)})}, \qquad m = 1,2,\ldots,M \tag{3.10}$$

The target state is estimated by

$$\hat{X}_{i+1} = E_{p(X_{i+1}|Z_{1:i+1})}(X) \approx \sum_{j=1}^{M} p(Z_{i+1}|X_{i+1} = s_{i+1}^{(j)}) \tag{3.11}$$

3.3.2.1 Sample generation

In the Particle filter tracker algorithm, a sample set is used to approximate the posterior density $p(X_i|Z_{1:i})$ of the target state. The sample set contains $s_i^{(m)}$ samples and their corresponding weight $\pi_i^{(m)}$. Here, m= 1,2, …M, where M is the sample size.

The samples are generated from the state transition model $p(X_{i+1}|X_i)$, which is built through probabilistic learning. It is noted that deterministic learning from some training sequences cannot obtain a motion model representing the all the possible movements of the vehicles. In the Monte Carlo tracker, the vehicle position is predicted using the movement of previous steps, and then the samples $s_{i+1}^{(m)}$ are generated around the prediction randomly.

The vertical direction in the frame is set to be parallel with the vehicle flow direction in the road. The elapsed time between two consecutive frames in the video sequence is very small that depends on frame-rate of video. In this short time period, the vehicle horizontal movement is negligible. Further, its speed does not change dramatically. The vehicle center position is predicted by following equations:

$$C^{\bar{x}}_{i+1} = C^{\hat{x}}_{i}$$

$$C^{\bar{y}}_{i+1} = C^{\hat{y}}_{i} + \quad (C^{\hat{y}}_{i} - C^{\hat{y}}_{i-1}) + (1 - \quad)(C^{\hat{y}}_{i-1} - C^{\hat{y}}_{i-2}) \qquad (3.12)$$

Where $(C^{\bar{x}}_{i+1}, C^{\bar{y}}_{i+1})$ is the predicted vehicle center position at $i+1$ $(i \geq 2)$ frame after the start of tracking, $(C^{\hat{x}}_{i}, C^{\hat{y}}_{i})$ is the estimated position in frame i, and (< 1) is a constant. For $i = 2$, is chosen as 1. Here, to automatically start tracking of a vehicle, the co-ordinate of center position of key vehicular blob corresponding to the first VDL will be used as the initial input $(C^{\hat{x}}_{1}, C^{\hat{y}}_{1})$.

The sample $s^{(m)}_{i+1} = (s^{x(m)}_{i+1}, s^{y(m)}_{i+1})$ is generated around $(C^{\bar{x}}_{i+1}, C^{\bar{y}}_{i+1})$:

$$s^{x(m)}_{i+1} = C^{\bar{x}}_{i+1} + \tilde{R} cos\text{Ũ}, \quad m = 1, 2, \ldots, M$$

$$s^{y(m)}_{i+1} = C^{\bar{y}}_{i+1} + \tilde{R} sin\text{Ũ}, \quad m = 1, 2, \ldots, M \qquad (3.13)$$

Where $\tilde{R}$ is a random number with normal distribution Ɲ(μ, ^{2}), where μ is mean and ^{2} is variance and Ũ is a random number with uniform distribution on the interval $[0, 2\pi]$.

3.3.2.2 Local image intensity measurement

Vehicles are of different shapes and a vehicle sillouttes contain varying number of pixels having significant intensity and edge variations within. Here, for simplicity, we consider a circular object around estimated center. This radius of the circular object is chosen based on the image intensity property observation in such a way that image intensity remains constant inside the object varies significantly outside the boundary [47].

On the basis of these properties, a method is adopted to measure the local image intensity features Z for a given position of the vehicle center (C^x, C^y).

By performing radial edge detection around (C^x, C^y), the object boundary is detected. We construct several line segments extending radially from (C^x, C^y) with coordinates $(l_{\theta,x}(k), l_{\theta,y}(k))$ such that

$$l_{\theta,x}(k) = C^x + r(k) cos\theta$$

$$l_{\theta,y}(k) = C^y + r(k)sin\theta,\ \ \theta = \frac{k1}{\ _1}2\pi\ , \qquad k1 = 0,1,\ldots\ldots,\ {}_1 - 1$$

$$r(k) = r_1 + \frac{p}{K}(r_2 - r_1), \qquad r_1 < r_2,\ \ p = 0,1,2,\ldots.,K \tag{3.14}$$

Where ${}_1$ is the number of line segments, $K + 1$ is the number of points on each line, θ gives the orientation of the line segment, and r_1 and r_2 are pre-specified values delimiting the length of the line segments. Note that (3.14) defines line segments for detecting potential object boundaries, where (3.13) gives a locus of samples around the predicted center position. The one-dimensional edge detection operator is applied on each line segment [44], [48].

$$e_\theta(k) = \left|\acute{I}_\theta(k-2) + 2\acute{I}_\theta(k-1) - 2\acute{I}_\theta(k+1) - \ \acute{I}_\theta(k+2)\right| \tag{3.15}$$

Where $\acute{I}_\theta(k)$ is the image intensity at point$(l_{\theta,x}(k), l_{\theta,y}(k))$, which is obtained by bilinear interpolation. The corresponding coordinate, denoted $(e_{\theta,x}, e_{\theta,y})$, with the maximum $e_\theta(k)$ is the detected edge point for the orientation θ. The value of r_θ records the distance between (C^x, C^y) and $(e_{\theta,x}, e_{\theta,y})$.

The object shape characteristic is represented by $\bar{r}$ and ${}_r$, the mean and the variance of r_θ, respectively. The object image intensity characteristic σ_I is defined as the image intensity standard deviation inside the vehicular-object. In this method, σ_I is obtained by computing the image intensity standard deviation in the detected area centering at (C^x, C^y) with radius $\bar{r} - \delta$, where δ is a positive constant. Here, the use of the circular area to compute σ_I in an area as large as possible and the use of $\bar{r} - \delta$ are to lessen the effect of the boundary. The local image intensity measurement for the position (C^x, C^y) is defined as the following two-component vector:

$$Z = [r_\theta\ \sigma_I]^T \tag{3.16}$$

According to object image intensity properties, if (C^x, C^y) is near the center, the elements of r_θ should not differ much with each other and $\sigma_I$should not be large.

Radial edge detection requires choosing r_1 and r_2 appropriately according to the object size. The edge detection detects the boundary where r_1 and r_2 are chosen appropriately. If the radial edge detection can detect the object boundary on all orientations ($_1$), we regard the edge detection as successful.

3.3.2.3 Sample Weighting

Given a set of generated samples and the associated image measurements, the sample weighting is defined by comparing the local image intensity measurement of the sample with that of the target object center. Here, normalized weight $\pi_{i+1}^{(m)}$ for each sample is calculated using Intensity standard deviation and mean of radius.

Let, the target center in the first frame (C_1^x, C_1^y) is known to us. The local image intensity measurement of (C_1^x, C_1^y) is $Z_1 = [r_{\theta,1}\sigma_{I,1}]^T$ and that of $s_{i+1}^{(m)}$ is $Z_{i+1}^{(m)} = [r_{\theta,i+1}^{(m)}, \sigma_{I,i+1}^{(m)}]^T$. $\mathrm{q}_{1,i+1}^{(m)}$ is defined to measure the difference between $r_{\theta,i+1}^{(m)}$ and $r_{\theta,1}$, and $\mathrm{q}_{2,i+1}^{(m)}$ is defined to measure the difference between $\sigma_{I,i+1}^{(m)}$and $\sigma_{I,1}$:

$$\mathrm{q}_{1,i+1}^{(m)} = \left|\sum_{\theta} \left(r_{\theta,i+1}^{(m)} - \bar{r}\right)^2 - N_1\sigma_r^2\right|, \tag{3.17}$$

$$\mathrm{q}_{2,i+1}^{(m)} = \left|\sigma_{I,i+1}^{(m)} - \sigma_{I,1}\right| \tag{3.18}$$

Where $\bar{r}$ and σ_r are, respectively, the mean and the variance of $r_{\theta,1}$, and N_1 is the number of detected boundary points$(N_1 = 8)$.

The sample weighting should be defined to force the maxima in $p(Z_{i+1}|X_{i+1})$ correspond to the target object image intensity features. If $s_{i+1}^{(m)}$ is near the target center, the distribution of $r_{\theta,i+1}^{(m)}$will not differ much with that of $r_{\theta,1}$ and $\mathrm{q}_{1,i+1}^{(m)}$ will be small. The values of $\sigma_{I,i+1}^{(m)}$ and $\sigma_{I,1}$ will not be large since the image intensity is relatively constant inside the selected object and $d_{2,i+1}^{(m)}$will not be large. These imply $\pi_{i+1}^{(m)}$ should be large when $\text{ʠ}_{1,i+1}^{(m)}$ and $\text{ʠ}_{2,i+1}^{(m)}$ are small and can be assumed to follow Gaussian distribution.

The conditions that the samples $s_{i+1}^{(m)}$is near the target center require both $\text{ʠ}_{1,i+1}^{(m)}$ and $\text{ʠ}_{2,i+1}^{(m)}$ be small and they are two independent measurements. Hence, the weight criterion is defined as:

$$Z_{i+1}^{(m)} = Z_{1,i+1}^{(m)} Z_{2,i+1}^{(m)}, \qquad (3.19)$$

Where

$$Z_{1,i+1}^{(m)} = e^{-\left(\frac{\text{ʠ}_{1,i+1}^{(m)^2}}{2\sigma_2^2}\right),} \qquad (3.20)$$

$$Z_{2,i+1}^{(m)} = e^{-\left(\frac{\text{ʠ}_{2,i+1}^{(m)^2}}{2\sigma_2^2}\right),} \qquad (3.21)$$

And σ_1 and σ_2 are constants. We disregard the constant coefficients in the Eqs.(3.20) and (3.21) because $Z_{i+1}^{(m)}$ will be normalized. After normalization, the weight assigned to the sample $s_{i+1}^{(m)}$ is

$$\pi_{i+1}^{(m)} = \frac{Z_{i+1}^{(m)}}{\sum_{j=1}^{M} Z_{i+1}^{(j)}} \qquad (3.18)$$

The center coordinate in $(i + 1)$th frame is calculated finally from the multiplication of normalized weight vector π and position sample vector s.

The center horizontal coordinate, $C_{i+1}^{x} = \sum_{m=1}^{M} \pi_{i+1}^{(m)} s_{i+1}^{x(m)}$

The center horizontal coordinate, $C_{i+1}^{y} = \sum_{m=1}^{M} \pi_{i+1}^{(m)} s_{i+1}^{y(m)}$

These coordinates will be used as input in equation (3.12) for estimating predicted center position in next frame.

3.3.3 Trajectory Correction in KVFs

In section 3.3.1 and 3.3.2 the center coordinate has been estimated using Kalman filtering and Particle filtering approach. In many cases, Kalman or Particle filter misses the correspondence of the vehicular object center in the estimated trajectory. The error in trajectory may occur due to weak image intensity features, partial occlusion of the selected object by other edges or structures, and broken boundaries. Thus, a investigation is mandatory at a regular interval for finding out whether the tracking coordinates for a given vehicle obtained from Kalman or Particle filter have missed the correspondence or not.

Here, in order that the tracking method will not miss the correspondence in the trajectory over the frames, the center coordinates of vehicle on KVFs is used. It has been mentioned in Chapter 2 that the TOBs corresponding to each vehicle in the MTSIs can be used to identify accurate co-ordinate positions in KVFs. The center position estimated from the Kalman or Particle filter is compared in the selected number of KVFs of the proposed tracking method. The estimated trajectory co-ordinate is considered to be erroneous, if it significantly differs from the centroid co-ordinates of the vehicle in that frame estimated from the TOB.

Let $(C_M^{x(n)}, C_M^{y(n)})$ be the estimated center coordinate of a vehicular object in KVF corresponding to nth VDL calculated using the MTSI method presented in Section 2.2. Let $(C_F^{x(n)}, C_F^{y(n)})$ be the center coordinates of the same vehicular object in that same KVF estimated using the Kalman filtering approach presented in Section 3.3.1 or Particle filtering approach presented in Section 3.3.2, respectively. To determine whether the trajectory estimated from filter has been deviated significantly from the estimated positions using MTSI, the following criteria are used:

$$Y1 : \frac{|C_F^x - C_M^x|}{X_{Max}} > \mathbb{T}_C^x \qquad (3.23)$$

$$Y2 : \frac{|C_F^y - C_M^y|}{Y_{Max}} > \mathbb{T}_C^y \qquad (3.24)$$

$$Y3 : \frac{|C_F^x - C_M^x|}{W_M^{x(n)}} > \mathbb{T}_W^x \qquad (3.25)$$

$$Y4 : \frac{|C_F^y - C_M^y|}{W_M^{y(n)}} > \mathbb{T}_W^y \qquad (3.26)$$

Where, X_{Max} and Y_{Max} indicates the highest coordinate value along X-axis and Y-axis, respectively. $W_M^{x(n)}$ and $W_M^{y(n)}$are the width along x-axis and y-axis of KVB detected using nth VDL, respectively. Here,$\mathbb{T}_C^x$, $\mathbb{T}_C^y$, $\mathbb{T}_W^x$, $\mathbb{T}_W^y$ are the threshold parameters, which are chosen based on movement on the direction as well as average speed of the vehicular traffic. The factors of choosing these parameters due to the variations of illumination, vehicle size, and video resolution are also investigated on the tracking performance. For example, a higher threshold value may be chosen when shadow in a highly illuminated environment affects the segmentation of the vehicle. In the experiments, it is found that the value of the $\mathbb{T}_C^x$ lies between 1% and 5% whereas $\mathbb{T}_C^y$ lies between 5% and 15%. Here, the value of $\mathbb{T}_C^y$ is relatively greater than $\mathbb{T}_C^x$ because of the fact that the probablity of movement in flow direction (y-axis) is much higher than the probability of lateral movement in x-direction. On the other hand, $\mathbb{T}_W^x$ and $\mathbb{T}_W^y$ is selected to be 50% to ensure that the center doesnot lie outside the vehicular-object.

An error in tracking trajectory may occur in both x-direction and y-direction in any one direction. So, the decision on trajectory correction is true when

$$Y : Y1 \cup Y2 \cup Y3 \cup Y4 \qquad (3.27)$$

If Y is false, then the tracking will follow the trajectory porived by the filtering methods. On the other hand, if Y is true then error in estimate is confirmed. In case of such error in trajectory estimation, the co-ordinate estimated from the TOB $(C_M^{x(n)}, C_M^{y(n)})$ will be fed into the Kalman or Particle filter tracker to restart the tracking process from the next frame.

Here, increasing the number of VDL will increase the number of KVFs and as a result the frequency of tracking correction checking will also be increased. So, the number of VDLs required for the tracking algorithm is needed to be investigated in terms of number of vehicles in the scene, average speed of the vehicles, field-of-view, as well as the frame rate of the video. In our experiment, we found that using 5 VDLs on the road, gives an acceptable tracking performance.

3.4 Conclusion

In this chapter, two tracking approach has been proposed. Firstly, a low complexity tracking approach based on MTSI detection method is introduced, which provides an approximate estimate of trajectory in a significantly low computational time. Then, traditional Kalman filter based tracking and Particle filter based tracking approach that we adopted has been presented. Finally, the center coordinates of vehicle calculated from KVF obtained using the MTSI method, has been incorporated into the filter-based tracking approach to obtain a highly reliable trajectory. It is expected that the proposed high performance tracking method will not only provide an automatic tracking method with low complexity, but also will increase reliability of tracking.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Introduction

In order to evaluate the performance of the algorithms proposed in Chapter 3, experiments are carried out on a vehiclular traffic-video database. Both subjective and objective evaluations of the vehicular video sequences are done. The subjective evaluation is done through visualization of the tracking trajectory of the vehicles. On the other hand, the objective evaluation is done based on the closeness of estimated tracking trajectories from forward and backward direction. In ideal tracking situation, both these trajectories in terms of coordinates should be the same. Thus, the performance metric is defined based on the normalized relative distance between forward and backward tracking coordinate, referred to as NRDT. The NRDT value is used here to evaluate the competency of video-based tracking algorithms. In the experiments, five different video-based tracking algorithms, viz., Kalman-filter based tracking [18], [24], Particle-filter based tracking [49], [41] and three MTSI-based tracking methods proposed in Chapter 3 are considered. In the following sections of this chapter, details of the test data, mathematical definition of the performance metric, plot of forward and backward tracking trajectories in terms of the Cartesian coordinates for each vehicle for the proposed and the existing algorithms will be presented. Moreover, the performance of the proposed MTSI based tracking methods will be discussed based on performance metrics.

4.2 Vehiclular Traffic-Video Data and Description

Experimentations on the video-based vehicle tracking algorithms have been conducted on a database [39], [40] that contains road-traffic video of Dhaka, Bangladesh and Suwon, Korea, under various sunlight and traffic conditions using a fixed camera of model Sony W-110. In the database, both the incoming and outgoing traffic flows are considered, with variations in the elevation of camera, so that different appearances of similar vehicles are captured in the video. The traffic consists of vehicles of seven types, namely, motorbike (Type I), rickshaw (Type II), auto-rickshaw (Type III), car or station wagon (Type IV), jeep or minivan (Type V), covered-van (Type VI), and Bus (Type VII). The vehicles also can be grouped into four broad classes based on number of wheels, viz., 2Wh, 3Wh, 4Wh, and 6Wh. To process the sequences, the captured clips will be converted to gray-scale sequences having a suitable frame size (say, 176×144) and a suitable frame rate (say, 25 frame/sec). The minimum and maximum speeds of the vehicles in video traffic are 8 and 60 km/h, respectively.

4.3 Performance Metrics

Here, the evaluation of performance of tracking methods is done mainly by frame by frame relative distance between vehicle center coordinate in forward and backward tracking. The mathematical expressions of the performance metrics, viz., NRDT and ANRDT considered in this thesis are given next.

Let, $C_f^y(i)$ and $C_b^y(i)$ indicates the center coordinate in forward and backward tracking at ith frame. Moreover, Y_{Max} is the maximum pixel value in flow direction and F_c is the frame count of tracking. NRDT can be calculated using the following equation,

$$\text{NRDT} = \frac{\sum_{i=1}^{\mathbb{N}_F} |C_f^y(i) - C_b^y(\mathbb{N}_F - i + 1)|}{Y_{Max}F_c} \qquad (4.1)$$

Where, $\mathbb{N}_F$ is the maximum number of frames, for which tracking trajectory is continued.

Average NRDT for a class of vehicle is calculated using the following equation,

$$\text{ANRDT} = \frac{\sum_{j=1}^{\grave{\text{й}}} \text{NRDT}_j}{\grave{\text{й}}}$$

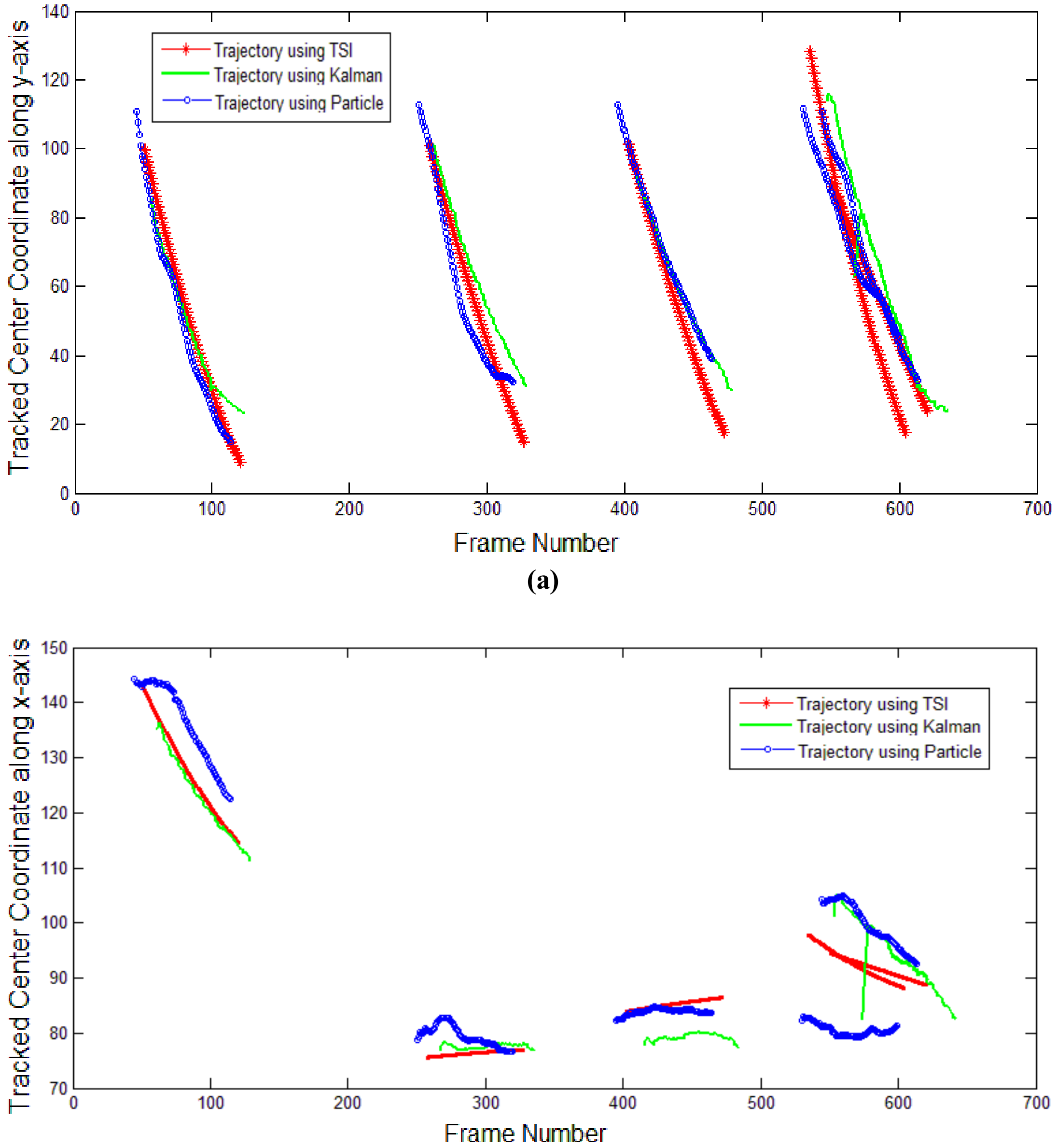
Where, $\grave{\text{й}}$ indicates the total number of vehicle in a selected vehicle-class and NRDT_j indicates the NRDT value of the j$^{\text{th}}$ object of that vehicle class.

4.4 Performance Analysis

At first, vehicle trajectory obtained from the low-complexity MTSI based tracking method is compared with the trajectory obtained using the filter-based tracking algorithms. Then, the performance of MTSI integrated Kalman filter(MTSIIKF) based tracking algorithm and MTSI integrated Particle filter(MTSIIPF) based tracking algorithm is compared with the existing Kalman and Particle filter based algorithms to find out which performs better on vehicle-video database. Average NRDT value for every vehicle class is investigated for comparing the performance of tracking algorithm. Here, a lower value of ANRDT indicates better tracking performance. Moreover, the correctness in estimating trajectory of vehicles is detected considering the NRDT values of the vehicles.

4.4.1 Low Complexity MTSI Tracking Algorithm

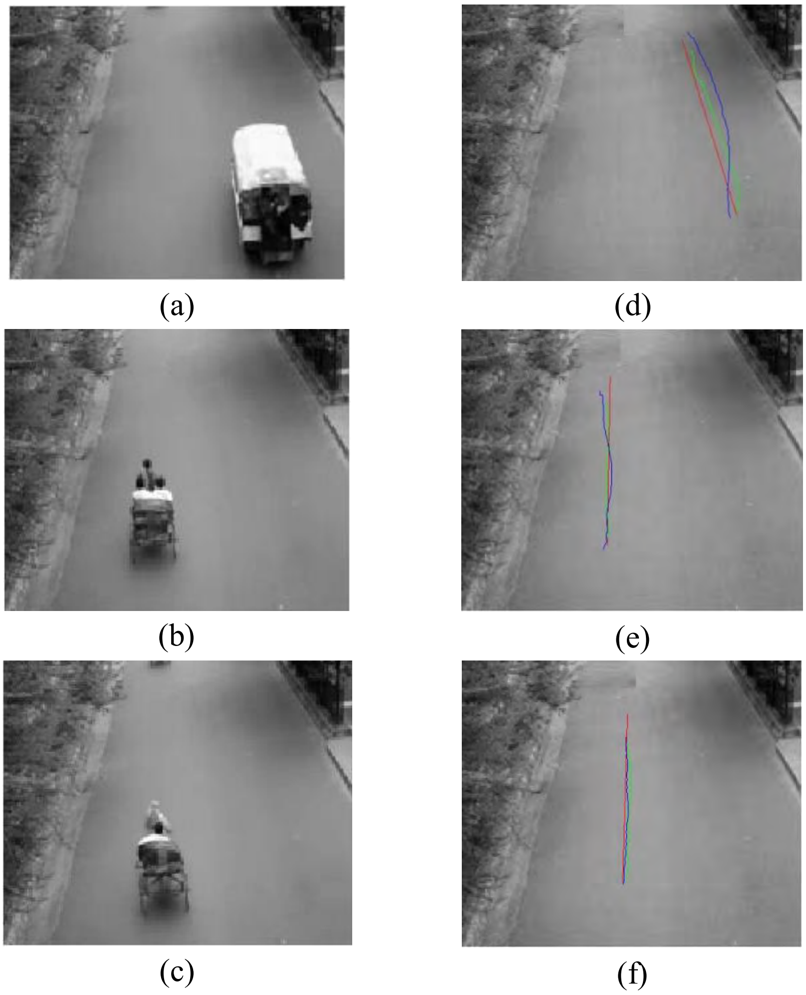
In this section, vehicle trajectory obtained from the low-complexity MTSI based tracking method is compared with the trajectory obtained using other popular methods like Particle filter [49] and Kalman filter tracking [24]. The tracking trajectory estimate of low complexity TSI based tracking method is very close to the tracking trajectory estimated using using the filter-based tracking algorithms, as shown in Figs 4.1 and 4.2. Thus, it may be useful in cases where very low computational complexity is expected, as the performance is close enough to the other popular methods.



(a)

(b)

Figure 4.1: Comparison of Low-Complexity MTSI based tracking approach with Kalman [18] and Particle filtering [41] approach in a typical video sequence in terms of tracking of (a) Vertical center coordinates; (b) Horizontal center coordinates


(a) (d)

(b) (e)

(c) (f)

Figure 4.2: Comparison of trajectory of Low-Complexity MTSI tracking method with Kalman[18] and Particle[41] filtering approach on real frame. Here, (a), (b) and (c) show three vehicles in a typical video sequence and (d), (e) and (f) show the corresponding trajectories, respectively.

4.4.2 Kalman and Particle Filter-Based Tracking Algorithms

In this section, the tracking coordinates obtained using existing filter-based algorithms are evaluated. For evaluation, the Cartesian coordinates in flow direction (y-axis) , obtained using Kalman filter based [24] and Particle filter based tracking alogorithm[49] are plotted in Figures 4.3 and 4.4 for two typical sequences of database. Here, only coordinate in y-direction is considered, as it is the vehicle flow direction. In all the following figures the line with red star will indicate the forward tracking trajectory and the line with blue bubble indicates backward tracking trajectory.

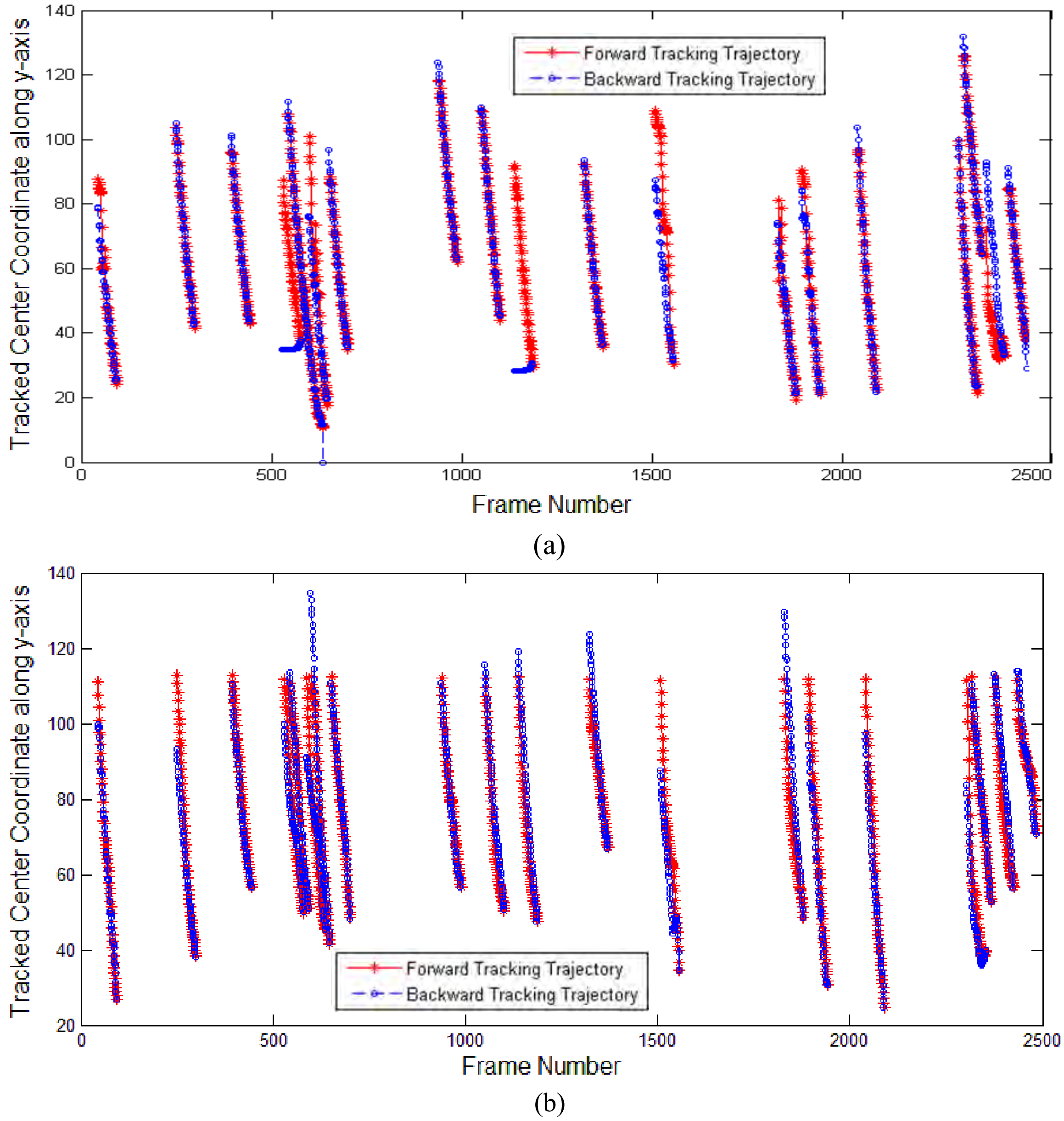
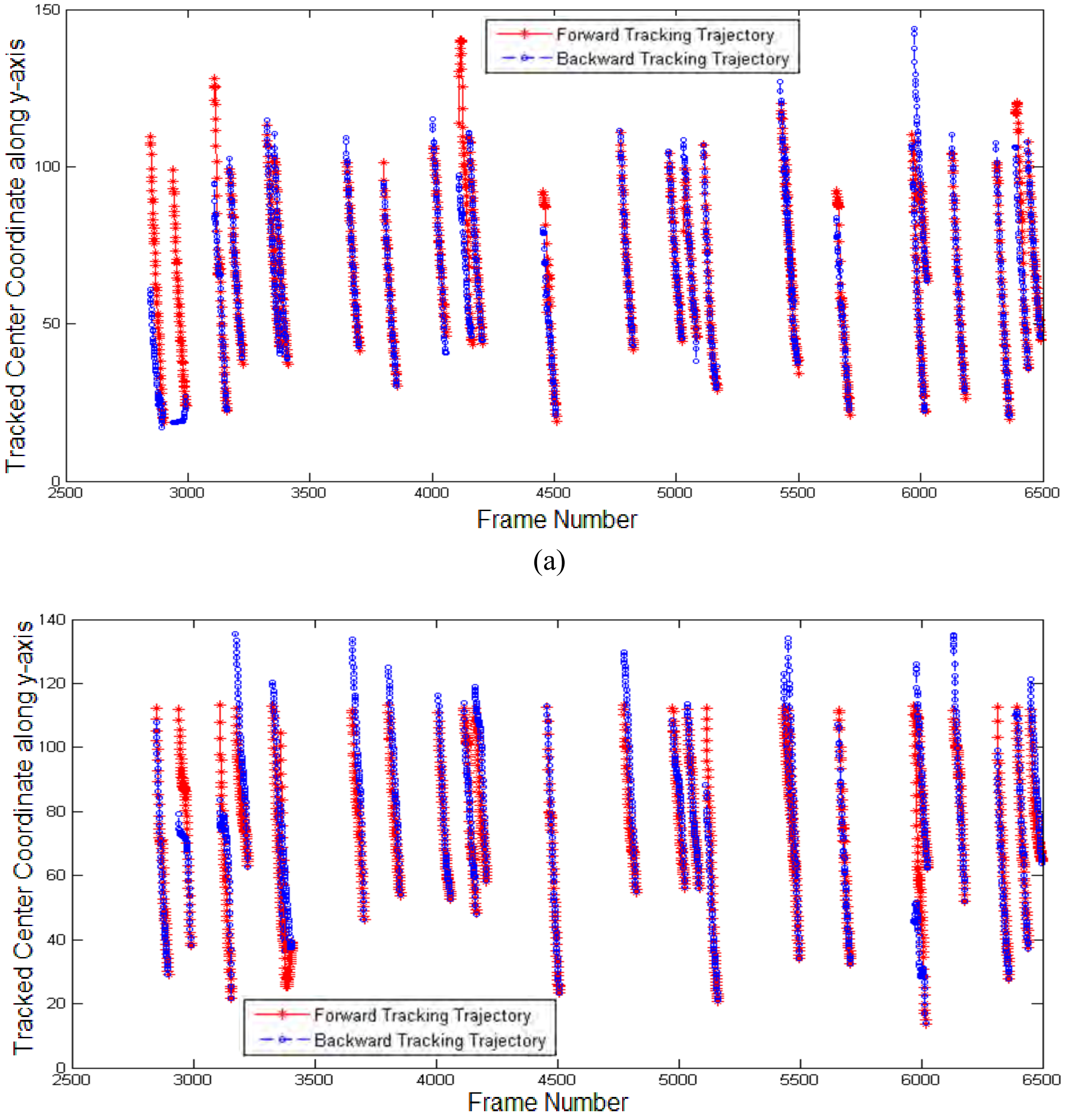


Figure 4.3: Tracking center coordinate along flow direction of first typical video sequence in forward and backward direction, using (a) Kalman filter (b) Particle Filter



(a)

(b)

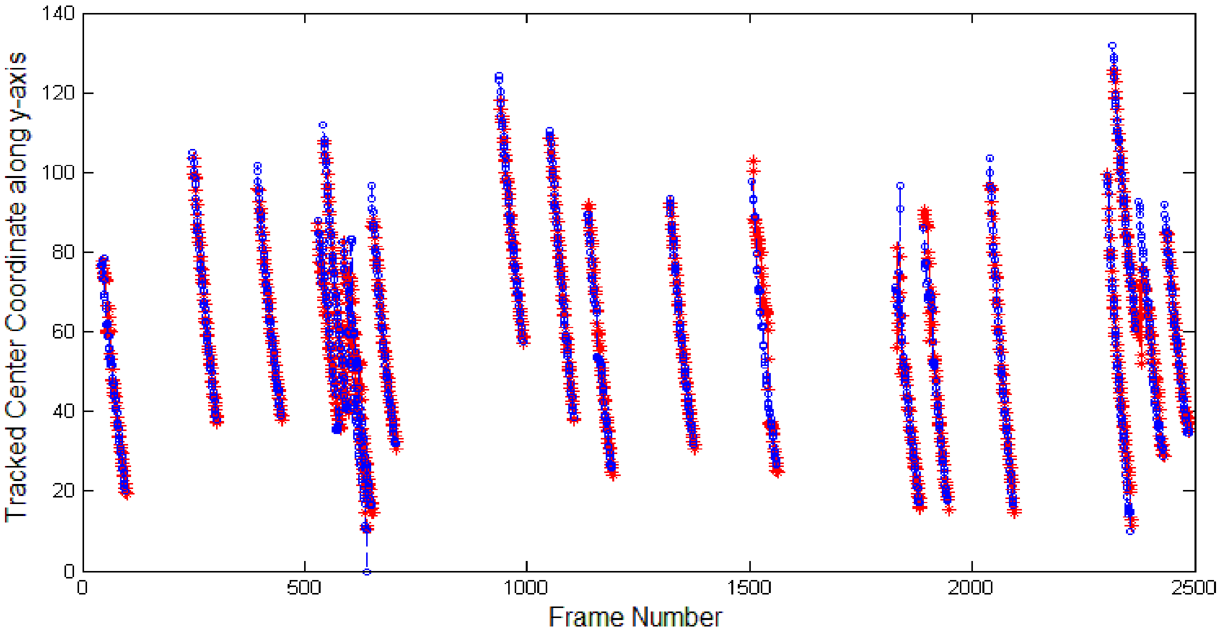
Figure 4.4: Tracking center coordinate along flow direction of second typical sequence in forward and backward direction, using (a) Kalman filter and (b) Particle Filter

From Figures 4.3 and 4.4, it is evident that Particle tracker misses trajectory in lesser cases than that of Kalman tracker. However, it can also be seen that, most of the vehicles those don"t lose trajectory, the Kalman tracker performs better in terms of closeness between forward and backward tracking coordinate.

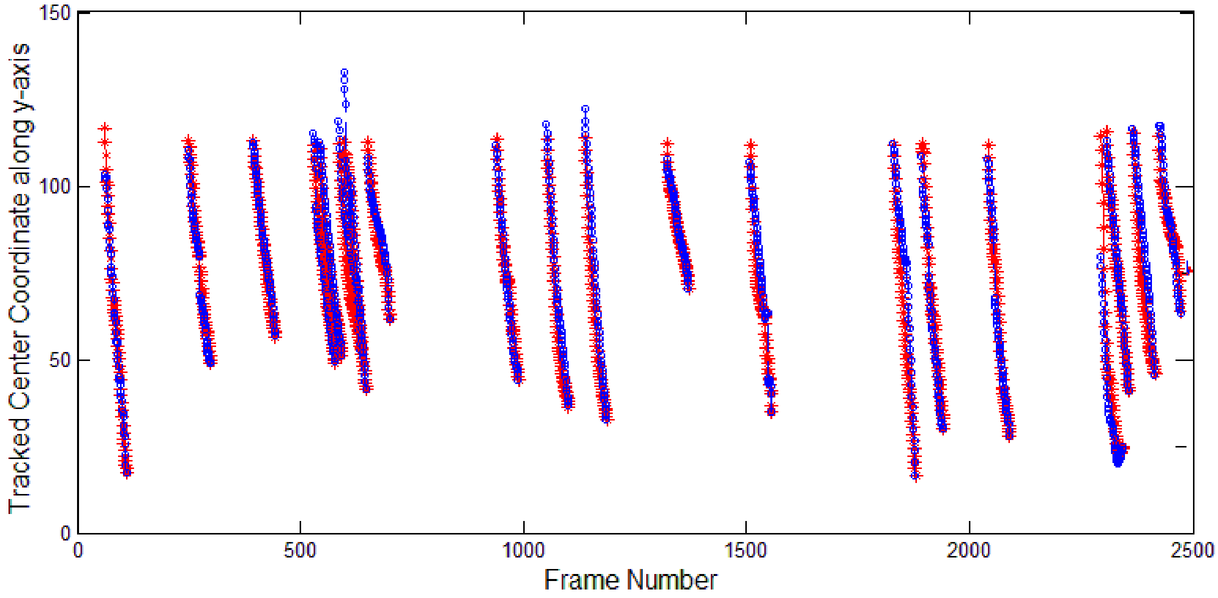
This result is expected, as Kalman filter estimates the predictor and filter as linear system. So, if the vehicle movement is linear, Kalman filtering performs very well and the tracking coordinate matches closely in forward and backward tracking. However, in real world scenario, there are cases where vehicluar movement may not be linear and Kalman filter fails in such cases. On the other hand, Particle filter doesn‟t have the linearity problem of Kalman filter based tracking and performs better in such scenarios. Thus Particle filter tracker performs better in terms of not missing trajectory.

4.4.3 TSI Integrated Filter-based Tracking Algorithm

Integrating MTSI based correction mechanism, improves the performance of both Kalman filter and Particle filter based tracking methods. However, the performance of Kalman is improved more, as MTSI correction mainly performs to correct the tracking coordinates in case of significant drift from the actual trajectory and doesn‟t have any impact on coordinates in case of little drift. Thus, as the chance of losing trajectory completely is significantly reduced using correction mechanism, the relative distance between forward and backward tracking coordinate is the dominating performance evaluation criterion now. As a result, it is expected that MTSIIKF algorithm will perform better than that of MTSIIPF algorithm. For evaluation, the Cartesian coordinates in flow direction (y-axis), obtained using MTSIIKF approach and MTSIIPF approach are plotted in Figures 4.5 and 4.6 for the same typical sequences of database used in Figures 4.3 and 4.4.

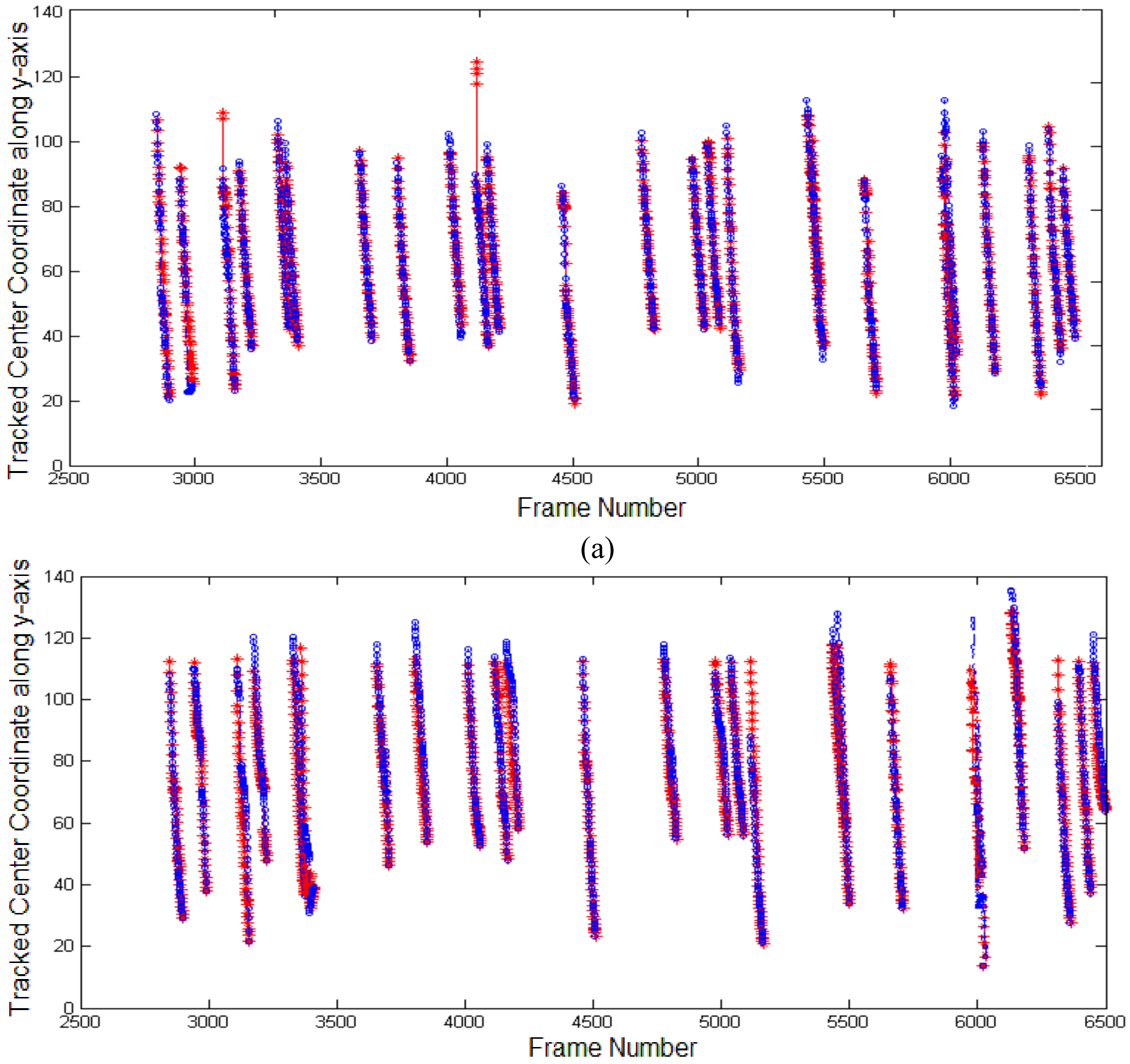


(a)



(b)

Figure 4.5: Tracking center coordinate along flow direction of first typical sequence in forward and backward direction, using (a) MTSIIKF and (b) MTSIIPF

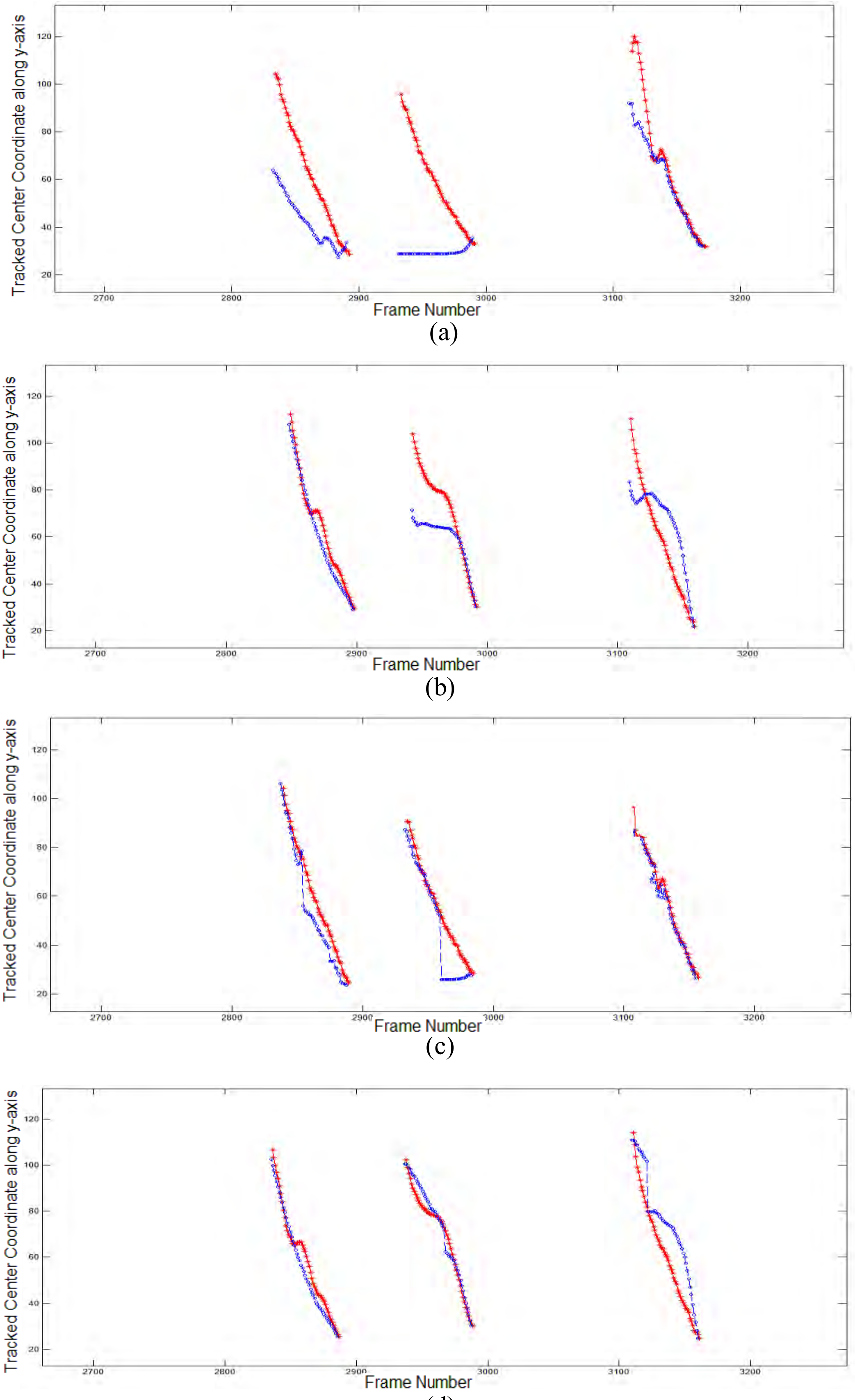


(a)

(b)

Figure 4.6: Tracking center coordinate along flow direction of second typical sequence in forward and backward direction, using (a) MTSIIKF and (b) MTSIIPF

It can be seen from the figures that the difference between forward and backward tracking coordinate is larger in MTSIIPF approach than that of MTSIIKF. For a better understanding of performance before and after integrating MTSI based correction, the zoomed-in trajectory of three vehicle of second typical video-sequence is shown in Figure 4.7. Moreover, the tracking trajectory of left-most and center vehicle of Figure 4.7 estimated using existing and proposed approaches, are shown in Figures 4.8 and 4.9.



(a)

(b)

(c)

(d)

Figure 4.7: Zoomed-in center coordinate tracking along flow direction of three vehicles in forward and backward direction using (a) Kalman filter (b) Particle filter (c) MTSIIKF and (d) MTSIIPF

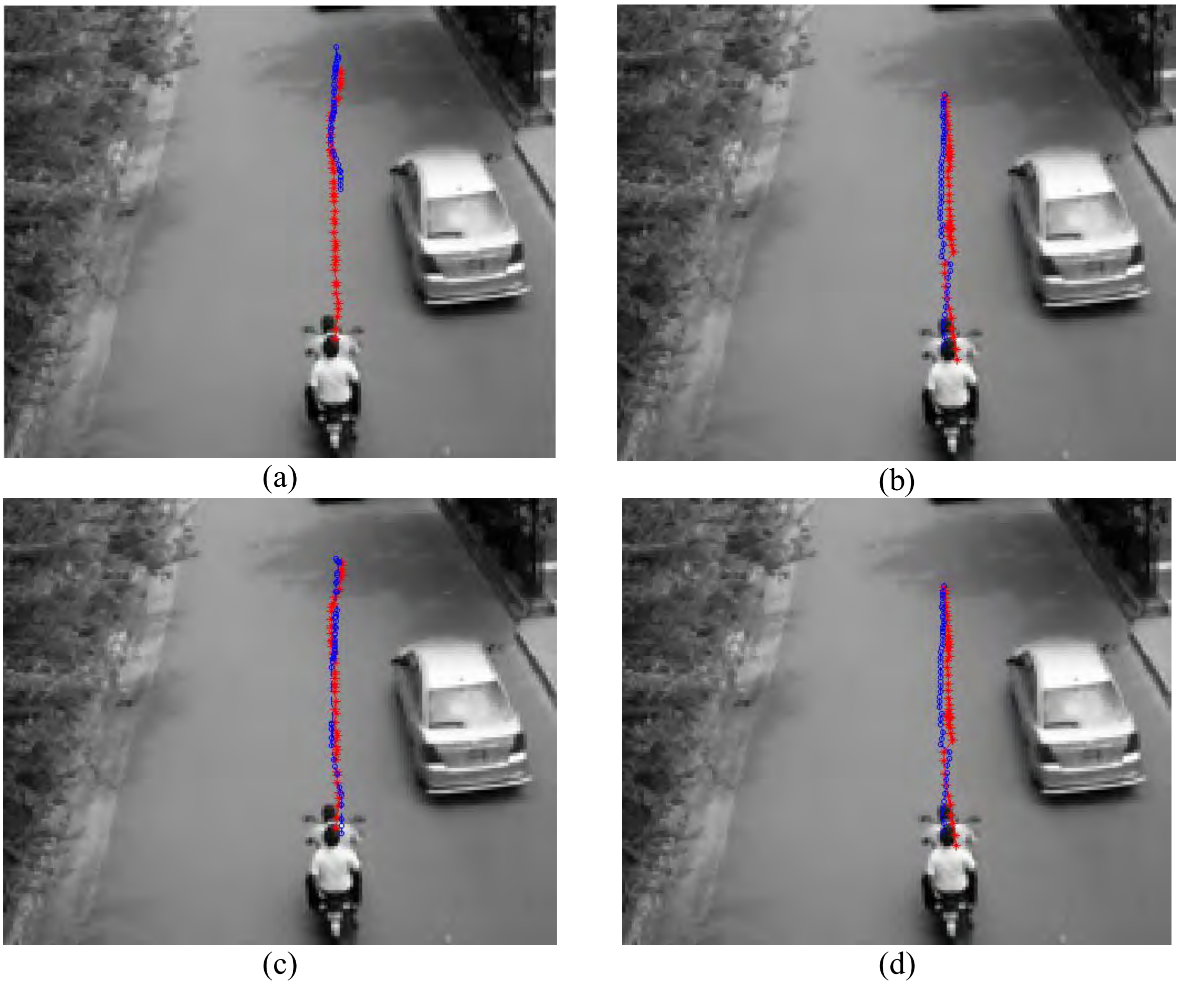
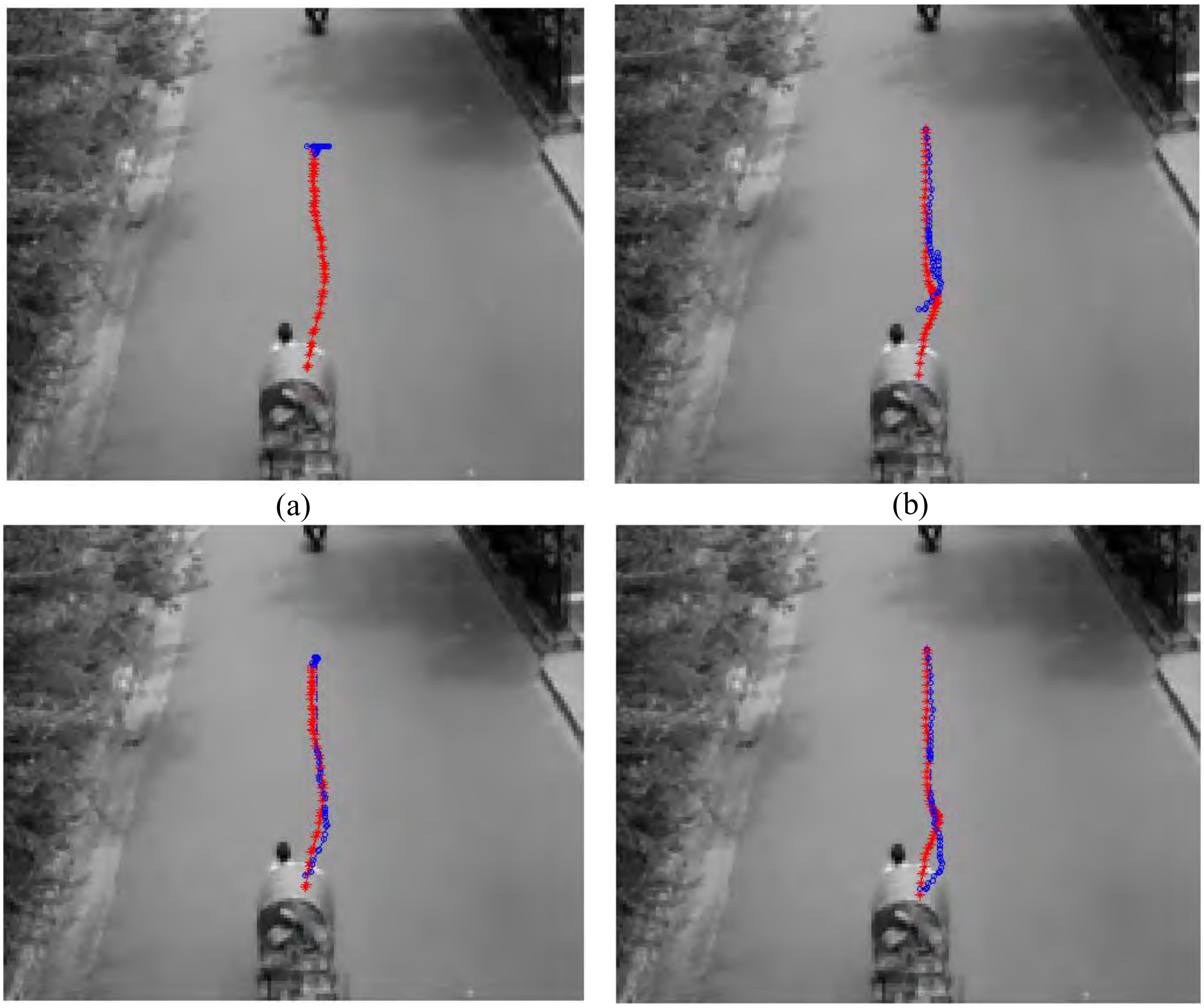


Figure 4.8: Tracking Trajectory of left most vehicle of the sequence considered in Figure 4.7 using (a) Kalman filter, (b) Particle filter, (c) MTSIIKF and (d) MTSIIPF



(a)

(b)

(c)

(d)

Figure 4.9: Tracking Trajectory of middle vehicle of the sequence considerd in Fig 4.7 using (a) Kalman filter, (b) Particle filter, (c) MTSIIKF and (d) MTSIIPF

From the Figures 4.6 and 4.7, it is evident that after integrating MTSI based correction, the proposed tracker identifies the case of significant displacement from the actual trajectory using the of correction mechanism on KVFs. As a result, the tracking trajectory estimate is changed only for those vehicles and a significantly improved trajectory is found. Moreover, the closeness between forward and backward tracking is also increased for those vehicles. Additionally, from Figures 4.7, 4.8 and 4.9, it can be said that before integrating MTSI, tracker with Particle filter performs better than tracker with Kalman filter in terms of not losing trajectory. However, after integrating MTSI based correction, the chance of completely missing trajectory is significantly reduced. Thus, on the basis of closeness between trajectory coordinate, MTSIIKF performs better than MTSIIPF.

4.5 Results and Analysis

Extensive experimentations have been carried out on the database discussed in Section 4.2, in order to perform objective evaluation of the performance of the proposed MTSI based vehicle tracking methods. For this, the proposed MTSI-based vehicle tracking methods are compared with two other existing methods, i.e., the Particle filter based tracking method [49] and Kalman filter based tracking method [24]. The results concerning the performance of vehicle tracking algorithms on different vehicle class, based on closeness between forward and backward trajectory coordinate is shown in Table 4.3 in three video sequences, when the environment was sunny, cloudy, and normal. Here, ANRDT value of different vehicle-type in video-sequences is shown in Table 4.3 the proposed and existing tracking algorithms. A lower ANRDT value indicates better tracking performance. From the table, it is found that MTSIKF provides the best result.

On the other hand, the results concerning the error in tracking vehicles of four methods, mainly considered in our experiment are shown in Table 4.4. Here, NRDT value for each vehicle is calculated to measure the correctness of tracking. From the visual inspections and observing the corresponding NRDT values, a vehicle with NRDT value lower than 0.07 is considered as correctly tracked. From the table, it is found that MTSIKF and MTSIPF provide best result in terms of number of vehicle correctly tracked and their performance is same in the video sequences of vehicle-traffic video databse [40].

Table 4.3: Results Concerning the Performance of vehicle Tracking Algorithms using ANRDT Value of Different Vehicle Class

Test Sequence	Methods	2Wh	3Wh		4Wh		6Wh	
		Type I	Type II	Type III	Type IV	Type V	Type VI	Type VIII
Normal Sequence	Vehicle Count	29	41	78	47	51	31	42
	Particle[49]	0.0607	0.0582	0.0595	0.0632	0.0529	0.0221	0.0248
	Kalman[24]	0.0414	0.0471	0.0356	0.0381	0.0391	0.0152	0.0169
	MTSIIPF	0.0358	0.0411	0.0328	0.0346	0.0427	0.0188	0.0212
	MTSIIKF	0.0225	0.0272	0.0171	0.0244	0.0291	0.0152	0.0169
Sunny Sequence	Vehicle Count	31	48	89	68	34	28	57
	Particle[49]	0.0745	0.0669	0.0588	0.0538	0.0455	0.0259	0.0210
	Kalman[24]	0.0552	0.0501	0.0347	0.0401	0.0274	0.0113	0.0127
	MTSIIPF	0.0513	0.0528	0.0361	0.0383	0.0285	0.0259	0.0210
	MTSIIKF	0.0239	0.0351	0.0189	0.0263	0.0181	0.0113	0.0127
Cloudy Sequence	Vehicle Count	23	27	58	45	37	22	29
	Particle[49]	0.0698	0.0622	0.0620	0.0515	0.0403	0.0197	0.0229
	Kalman[24]	0.0486	0.0514	0.0450	0.0433	0.0295	0.0141	0.0146
	MTSIIPF	0.0452	0.0498	0.0363	0.0339	0.0317	0.0197	0.0188
	MTSIIKF	0.0197	0.0312	0.0226	0.0236	0.0231	0.0141	0.0146
Total	Particle[49]	0.0683	0.0627	0.0598	0.0559	0.0470	0.0228	0.0227
	Kalman[24]	0.0485	0.0493	0.0377	0.0404	0.0329	0.0136	0.0145
	MTSIIPF	0.0442	0.0479	0.0350	0.0359	0.0354	0.0215	0.0205
	MTSIIKF	**0.0222**	**0.0314**	**0.0192**	**0.0250**	**0.0242**	**0.0136**	**0.0145**

From these results, it is evident that tracking performance of both the MTSIIKF and MTSIIPF metods are more reliable than that of traditional filter-based methods. However, due to the fact that vehicular traffics follow nearly linear trajectories mostly, in general, the MTSIIKF method shows better performance, as compared to the MTSIIPF method in terms of closeness between forward and backward tracking trajectory.

TABLE 4.4: RESULTS CONCERNING THE PERFORMANCE OF THE VEHICLE TRACKING ALGORITHMS IN TERMS OF CORRECTNESS IN TRACKING VEHICLES

Test Sequence	Methods	2Wh	3Wh		4Wh		6Wh		Total Tracked	Tracking Error
		Type I	Type II	Type III	Type IV	Type V	Type VI	Type VIII		
Normal Sequence	Kalman[24]	24(29)	38(41)	70(78)	42(47)	49(51)	31(31)	42(42)	296(319)	7.21%
	Particle[49]	26(29)	40(41)	72(78)	44(47)	50(51)	31(31)	42(42)	305(319)	4.39%
	Particle+TSI	29(29)	41(41)	77(78)	47(47)	51(51)	31(31)	42(42)	319(319)	-
	Kalman+TSI	29(29)	41(41)	78(78)	47(47)	51(51)	31(31)	42(42)	319(319)	-
Sunny Sequence	Kalman[24]	28(31)	46(48)	78(89)	64(68)	32(34)	28(28)	57(57)	335(355)	5.63%
	Particle[49]	28(31)	47(48)	81(89)	65(68)	31(34)	28(28)	57(57)	338(355)	4.79%
	Particle+TSI	31(31)	48(48)	89(89)	68(68)	34(34)	28(28)	57(57)	355(355)	-
	Kalman+TSI	31(31)	48(48)	89(89)	68(68)	34(34)	28(28)	57(57)	355(355)	-
Cloudy Sequence	Kalman[24]	19(23)	26(27)	48(58)	41(45)	36(37)	22(22)	29(29)	221(241)	8.29%
	Particle[49]	21(23)	26(27)	49(58)	44(45)	36(37)	22(22)	29(29)	227(241)	5.81%
	Particle+TSI	23(23)	27(27)	57(58)	45(45)	37(37)	22(22)	29(29)	241(241)	-
	Kalman+TSI	23(23)	27(27)	57(58)	45(45)	37(37)	22(22)	29(29)	241(241)	-
Total	Kalman[24]	71(83)	110(116)	196(225)	147(160)	117(122)	81(81)	128(128)	852(915)	6.89%
	Particle[49]	75(83)	113(116)	202(225)	153(160)	117(122)	81(81)	128(128)	870(915)	4.92%
	Particle+TSI	**83(83)**	**116(116)**	**225(225)**	**160(160)**	**122(122)**	**81(81)**	**128(128)**	**915(915)**	-
	Kalman+TSI	**83(83)**	**116(116)**	**225(225)**	**160(160)**	**122(122)**	**81(81)**	**128(128)**	**915(915)**	-

4.6 Conclusion

In this chapter, the performance of proposed MTSI based tracking algorithms has been evaluated in terms of closeness between forward and backward tracking trajectories. Both the graphical representation and NRDT value has been considered for the detection of correctness in trajectory estimation and closeness with actual trajectory. The performance of proposed MTSI integrated filter-based methods has been compared with two existing filter-based tracking methods, i.e. Kalman and Particle filter. From these evaluations, it is found that the proposed tracking algorithms perform better than the existing algorithms in terms of accuracy in tracking the complete trajectory. Additionally, it is also found that, among the proposed methods MTSIKF performs better than MTSIPF in terms of closeness with actual trajectory.

5. CONCLUSION

5.1 Conclusion and Discussion

The development of the efficient tracking algorithm for vehicles is crucial in designing a video-based intelligent transportation system. Traditional tracking algorithms mainly suffer from initialization of tracking, sensitivity to drift from true object position in long sequences, and absence of corrective mechanism. In this thesis, MTSI based detection mechanism is integrated into the Monte-Carlo Particle filter and Kalman filter based-tracking algorithms. Such a use of MTSIs in tracking algorithm not only provide the opportunity of tracking initialization with a low computational complexity, but also supplies a corrective measure for the traditional tracking methods. To evaluate the performance of tracking algorithms, all the vehicles have been tracked from both the forward and backward direction. The net difference between forward and backward tracking coordinate is used as the peroformance measurement criterion. Extensive experimentations have been carried out on a number of video sequences obtained under different illumination conditions to evaluate the tracking performance of the proposed tracking methods, compared with that of existing methods. The result provides a useful insight about choosing method for tracking in terms of environment. Furthermore, results have shown that the proposed methods perform better than existing methods in terms of accuracy. It has also been shown that among the proposed MTSI integrated methods, relatively low computationally complex MTSIIKF method performs better than the more computationally complex MTSIIPF method when most of the vehicles on the road proceeds with a linear movement. Hence, the proposed MTSI-based methods may be highly effective in designing a real-time video-based intelligent transportation system.

5.2 Scope of Further Work

There are a number of scopes to extend the research done in the thesis. For example, in the proposed approach only single static camera is used to store video. Video sequence obtained from multiple number of moving cameras or vehicle-mounted cameras may be examined to further elaborate this method with the consideration of stereo vision and relative speed of vehicles.

REFERENCES

[1] Peden, M. M., Scurfield, R., Sleet, D., Mohan, D., Hyder, A. A., Jarawan, E., and Mathers, C., “World report on road traffic injury prevention: Summary,” *World Health Organization, Geneva, Switzerland*, Apr. 2004, [Online]. Available: http://www.who.int/ world-health-day/2004/ infomaterials/world_report/en.

[2] Sun, Z., Bebis, G., and Miller, R., “On-road vehicle detection: A review,” *IEEE Transaction on Pattern Anaysis and Machine Intelligence,* vol. 28, no. 5, pp. 694–711, May 2006.

[3] “Traffic safety facts 2011: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System,” Tech. Rep. no. 811754AR ,*U.S. Dept. Transp., Nat. Highway Traffic Safety Admin.,* Washington, DC, Jan. 2012.

[4] Duarte, M. F. and Hen Hu, Y., “Vehicle classification in distributed sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 64, no.7, pp. 826-838, July 2004.

[5] Brooks, R. R., Ramanathan, P. and Sayeed, A. M., “Distributed target classification and tracking in sensor networks,” *Proceedings of the IEEE*, Aug. 2003, vol. 91, no.8, pp. 1163-1171.

[6] Tokoro, S., Kuroda, K., Kawakubo, A., Fujita, K., and Fujinami, H., “Electronically scanned millimeter-wave radar for pre-crash safety and adaptive cruise control system,” in *Proceedings of Intelligent Vehicles Symposium*, Columbus, OH, USA, June 2003, pp. 304–309.

[7] Gunnarsson, J., Svensson, L., Danielsson, L., and Bengtsson, F., “Tracking vehicles using radar detections,” In *Proceedings of IEEE Intelligent Vehicles Symposium,* Istanbul, Turkey, June 2007, pp. 296-302.

[8] Weigel, H., Lindner, P., and Wanielik, G., “Vehicle tracking with lane assignment by camera and lidar sensor fusion,” In *Proceedings of IEEE Intelligent Vehicles Symposium*, Xi'an, China, June 2009, pp. 513-520.

[9] Premebida, C., Monteiro, G., Nunes, U., and Peixoto, P., “A lidar and vision-based approach for pedestrian and vehicle detection and tracking,” In *Proceedings of IEEE Intelligent Transportation Systems Conference*, Seattle, WA, Sep. 2007, pp. 1044-1049.

[10] McCall, J. C., and Trivedi, M. M., “Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation,” *IEEE Transactions on Intelligent Transportation Systems*, vol.7, no.1, pp.20-37, Mar. 2006.

[11] Sivaraman, S., and Trivedi, M. M., "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773-1795, Dec. 2013.

[12] Hoogendoorn, S. P., Van Zuylen, H. J., Schreuder, M., Gorte, B., and Vosselman, G., "Microscopic traffic data collection by remote sensing," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855, no. 1, pp.121-128, 2003.

[13] Kim, Z., "Real time object tracking based on dynamic feature grouping with background subtraction," In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, June 2008, pp.1-8.

[14] Amer A., "Voting-based simultaneous tracking of multiple video objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1448-1462, Nov. 2005.

[15] Dellaert F., and Thorpe C., "Robust car tracking using Kalman filtering and Bayesian templates," In *Proceedings of SPIE Conference on Intelligent Transportation Systems*, Pittsburgh, PA, Jan. 1998, pp.72-83.

[16] Stauffer C., and Grimson W. E. L., "Adaptive background mixture models for real-time tracking," In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, Jun. 1999, vol. 2, pp. 1-7.

[17] Isard M., and Blake A., "CONDENSATION-Conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol.29, no.1, pp.5-28, Aug. 1998.

[18] Segall, C. A., Chen, W., and Acton, S. T., "Video tracking using the morphological pyramid," *Journal of Electronic Imaging*, vol. 8, no. 2, pp. 176 –184, Apr. 1999.

[19] Aksel A., and Acton S. T., "Target tracking using the snake particle filter," In *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, Austin, TX, May 2010, pp. 33-36.

[20] Shi, J., and Tomasi, C., "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, June 1994, pp. 593 – 600.

[21] Bay, H., Ess, A., Tuytelaars, T., and Gool, L.V., "Surf: Speeded up robust features," *Elsevier Journal of Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[22] Lowe, D. G., "Object recognition from local scale-invariant features," In *Proceedings of 7th IEEE international conference on Computer vision* , Kerkyra, Greece, Sept. 1999, vol. 2, pp. 1150-1157.

[23] Lowe, D.G., “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91 – 110, Nov. 2004.

[24] Li, X., Wang, K., Wang, W., and Li, Y. (2010, June). “A multiple object tracking method using Kalman filter,” In *Proceedings of International Conference on IEEE Information and Automation* (ICIA), Harbin, China, Jun. 2010, pp. 1862-1866.

[25] Simon, D. and Chia, T.L., “Kalman filtering with state equality constraints,” *IEEE Transactions on Aerospace and Electronic Systems*, vol.38, no.1, pp. 128-136, Jan. 2002.

[26] Isard, M. and Blake, A., “A mixed-state condensation tracker with automatic model-switching,” In *Proceedings of 6th International Conference on Computer Vision*, Bombay, India, Jan. 1998, pp. 107 – 112.

[27] Lehuger, A., Lechat, P., and Perez, P., “An adaptive mixture color model for robust visual tracking,” In *Proceedings of IEEE International Conference on Image Processing*, Atlanta, GA, , Oct., 2006, pp. 573 – 576.

[28] Nummiaro, K., Koller-Meier, E. and Van Gool, L., “A color-based particle filter,” In *Proceedings of 1st Int.Workshop on Generative-Model-Based Vision*, 2002, pp. 53 – 60.

[29] Kumar, T.S., and Sivanandam, S.N., “Object detection and tracking in video using particle filter,” In *Proceedings of 3rd International Conference on Computing Communication & Networking Technologies*, Coimbatore, India, Jul. 2012, pp. 1–10.

[30] Yilmaz, A., Javed, O., and Shah, M., “Object tracking: A survey,” *Journal of ACM Computing Survey*, vol.38, no.4, pp.1–45, Dec. 2006.

[31] Bar-Shalom, Y., Fortmann, T. E., and Cable, P. G., “Tracking and data association,” *Journal of Acoustical. Society of America*, vol. 87, no. 2, pp. 918, 1990.

[32] Cox, I.., and Hingorani, S., “An efficient implementation of Reid's Multiple Hypothesis Tracking algorithm and its evaluation for the purpose of visual tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.18, pp:138-150, Feb. 1996.

[33] Park, K., Lee, D., and Park, Y., “Video-based detection of street-parking violation,” In *Proceedings of International Conference on Image Processing, Computer Vision, Pattern Recognition*, Las Vegas, NV, 2007, vol. 1, pp. 152–156,.

[34] Rivlin, E., Rudzsky, M., Goldenberg, M., Bogomolov, U., and Lapchev, S., “A real-time system for classification of moving objects,” In *Proceedings of 16th International Conference on Pattern Recognition,* Quebec City, QC, Canada, 2002, vol. 3, pp. 688–691.

[35] Morris, B. T., and Trivedi, M. M., “Learning, modeling, and classification of vehicle track patterns from live video,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, Sep. 2008.

[36] Yalcin, H., Herbert, M., Collins, R., and Black, M. J., “A flow-based approach to vehicle detection and background mosaicking in airborne video,” In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, vol. 2, pp. 1202

[37] Veeraraghavan, H., Masoud, O., and Papanikolopoulos, N. , “Vision-based monitoring of intersections,” In *Proceedings of IEEE 5th International Conference on Intelligent Transportation Systems, Singapore, 2002*, pp.7–12.

[38] Zhang, Z., Cai, Y., Huang, K., and Tan,T., “Real-time moving object classification with automatic scene division,” In *Proceedings of IEEE International Conference on Image Processing*, San Antonio, TX, 2007, vol. 5, pp.149-152.

[39] Rashid, N. U., Mithun, N. C., Joy, B. R., and Rahman, S. M. M., “detection and classification of vehicles from a video using time-spatial image,” In *Proceedings of IEEE Intl Conf. on Electrical and Computer Engineering*, Dhaka, Bangladesh, Dec 2010, pp. 502-505.

[40] Mithun, N. C., Rashid, N. U., and Rahman, S. M. M., “Detection and classification of vehicles from video using multiple time-spatial images,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1215–1225, Sep.2012.

[41] Cui, J., Acton, S. T., and Lin, Z., “A Monte Carlo approach to rolling leukocyte tracking in vivo,” *Elsevier Journal of Medical Image Analysis*, vol.10, no.4, pp.598-610, Aug. 2006.

[42] Grenander, U., Chow, Y., Keenan, D., “HANDS. A Pattern Theoretical Study of Biological Shapes,” Springer-Verlag, NJ, 1991

[43] Acton, S. T., and Li, B., “A sequential monte carlo method for real-time tracking of multiple targets,” Tech Rep. no. A675235, Virginia University, Charlottesville Office of Sponsored Programs, pp. 1-19, May 2010.

[44] Blake, A. and Isard, M., “Active Contours: The application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion,” Springer- Verlag, NJ, USA, 1998.

[45] Reneau, J., “Tracking using intensity gradients and particle filtering,” In *Proceedings of Fall 2004 conference, Clemson University*, pp. 1-5, Clemson, SC, USA, 2004.

[46] Birchfield, S., “An elliptical head ttracker,” In *Proceedings of 31st Asilomar Conference on Signals Systems and Computers*, Pacific Grove, CA, USA, Nov. 1997, vol.2, pp.1710-1714.

[47] Bruno, M.G.S, “A particle filter algorithm for target tracking in images,” In *Proceedings of* International *Telecommunication Symposium*, Natal, Brazil, 2002.

[48] Isard M. and Blake A., “Contour tracking by stochastic propagation of conditional density,” In *Proceedings of 4th European Conference on Computer Vision*, Cambridge, UK, Apr. 1996, vol.1, pp. 343-356.

[49] Yang, C., Duraiswami, R., and Davis, L., “Fast multiple object tracking via a hierarchical particle filter,” In *Proceedings of 10th IEEE International Conference on Computer Vision*, Beijing, China, Oct. 2005, vol. 1, pp. 212-219.