Multiple On-road Vehicle Tracking Using Microscopic Traffic Flow Models

MULTIPLE ON-ROAD VEHICLE TRACKING USING MICROSCOPIC TRAFFIC FLOW MODELS

BY

DAN SONG, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2019) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Multiple On-road Vehicle Tracking Using Microscopic Traffic Flow Models
AUTHOR:	Dan Song B.Sc. (Information Engineering), National University of Defense Technology, Changsha, Hunan China
SUPERVISOR:	Prof. T. Kirubarajan
NUMBER OF PAGES:	xviii, 185

To my family

Abstract

In this thesis, multiple on-road vehicle tracking problem is explored, with greater consideration of road constraints and interactions between vehicles. A comprehensive method for tracking multiple on-road vehicles is proposed by making use of domain knowledge of on-road vehicle motion.

Starting with raw measurements provided by sensors, bias correction methods for sensors commonly used in vehicle tracking are briefly introduced and a fast but effective bias correction method for airborne video sensor is proposed. In the proposed method, by assuming errors in sensor parameter measurements are close to zero, the bias is separately addressed in converted measurements of target position by a linear term of errors in sensor parameter measurements. Based on this model, the bias is efficiently estimated by addressing it while tracking or using measurements of targets that are observed by multiple airborne video sensors simultaneously. The proposed method is compared with other airborne video bias correction methods through simulations. The numerical results demonstrate the effectiveness of the proposed method on correcting bias as well as its high computational efficiency.

Then, a novel tracking algorithm that utilizes domain knowledge of on-road vehicle motion, i.e., road-map information and interactions among vehicles, by integrating a car-following model into a road coordinate system, is proposed for tracking multiple

vehicles on single-lane roads. This algorithm is extended for tracking multiple vehicles on multi-lane roads: The road coordinate system is extended to two-dimension to express lanes on roads and a lane-changing model is integrated for modeling lane-changing behavior of vehicles. Since the longitudinal and lateral motions are mutually dependent, the longitudinal and lateral states of vehicles are estimated sequentially in a recursive manner. Two estimation strategies are proposed: a) The unscented Kalman filter combined with the multiple hypothesis tracking framework to estimate longitudinal and lateral states of vehicles, respectively. b) A unified particle filter framework with a specifically designed computationally-efficient joint sampling method to estimate longitudinal and lateral states of vehicles jointly. Both of two estimation methods can handle unknown parameters in motion models. A posterior Cramer-Rao lower bound is derived for quantifying achievable estimation accuracy in both single-lane and multi-lane cases, respectively. Numerical results show that the proposed algorithms achieve better track accuracy and consistency than conventional multi-vehicle tracking algorithms, which assumes that vehicles move independently of one another.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to some people very special to me. Without their encouragement and support, this thesis would never have been possible. First and foremost, I would like to acknowledge my deepest gratitude to my supervisor, Prof. T. Kirubarajan, for offering me such a great opportunity to join the Estimation, Tracking and Fusion Research Lab (ETFLab). I appreciate all his academic guidance and funding that makes my Ph.D. experience productive and unforgettable. His hardworking and rigorous attitude to research constantly inspires me how to be a good scientist.

I would like to thank the rest of my Ph.D. committee members: Prof. Ian Bruce and Prof. Aleksandar Jeremic, for their grateful support and insightful comments on my research work. It is greatly appreciated for theirs contributions of time in my supervisory meetings and this thesis. I am also thankful to the external examiner Prof. Balasingam Balakumar for taking his time in my thesis and defence. I would like to thank Dr. Ratnasingham Tharmarasa, for constructive advice and encouragement that he gives me every time when my research works encounter difficulties. I would like to acknowledge Christina Dalima, for her valuable editorial comments on my research papers and this thesis. My sincere thanks also goes to the administrative staff of the Electrical and Computer Engineering department, especially Ms. Cheryl

Gies and Tracey Coop, for their assistance on administrative matters.

I am grateful to all my lab mates and friends, in particular, Krishanth, Yuanhao, Qingsong, Yinghui, Duo, Ben, Jianfeng, Pengfei and Tongyu. Being with such great friends makes my life of studying abroad more exciting. I am also thankful to professors, Shunping Xiao, Wei Wang and Zhenhai Xu, and my friends, Haoyang, Ziwei and Bing in my previous university for their help and support throughout my Ph.D. study.

I would as well wish to express my appreciation to the China Scholarship Council and the School of Graduate Studies for offering me generous scholarships that support my study at McMaster University.

Last and most importantly, a special thanks goes to my family. Words can not express how grateful I am to my father, mother and my grandmother, for endless love they give me in my life, without which all my achievements have never been possible. Though my grandmother may not remember me at all, it does not matter to me to give her my love. The mind may forget, but the heart remembers. I would also like to thank my girlfriend Ting, for her love and companion.

Abbreviations

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AME	Adaptive Mileage Estimator
AP	Action Point
CA	Collision Avoidance
CFM	Car-following Model
EKF	Extended Kalman Filter
GA	Genetic Algorithm
GHR	Gazis-Herman-Rothery
GIS	Geographical Information System
GM	General Motors
GMTI	Ground Moving Target Indicator
GPS	Global Positioning System
GPU	Graphical Processing Unit
IDM	Intelligent Drive Model
IMM	Interacting Multiple Model
ITS	Intelligent Transportation System

MAA	Mean-adaptive Acceleration
MHT	Multiple Hypothesis Tracking
MOBIL	Minimizing Overall Braking Induced by Lane Change
MTF	Microscopic Traffic Flow
MTT	Multiple Target Tracking
NCV	Nearly Constant Velocity
OID	Optimal Importance Density
PCRLB	Posterior Cramer-Rao Lower Bound
PF	Particle Filter
RML	Recursive Maximum Likelihood
RMSE	Root Mean Square Error
SFM	Social Force Model
TPM	Transition Probability Matrix
UKF	Unscented Kalman Filter
UT	Unscented Transformation
VS-IMM	Variable Structure Interacting Multiple Model
2D	Two-dimensional

Contents

List of Tables

List of Figures

Declaration of Academic Achievement

This research presents analytical and computational work carried out solely by Dan Song, herein referred to as "the author", with advice and guidance provided by the academic supervisor Prof. T. Kirubarajan. Information that is presented from outside sources, which has been used towards analysis or discussion, has been cited when appropriate, all other materials are the sole work of the author.

Chapter 1

Introduction

1.1 Multiple On-road Vehicle Tracking: A Brief Review

Tracking multiple on-road vehicles is an important task in various applications, such as ground surveillance [23], traffic surveillance [3] and intelligent transportation systems (ITS) [4]. It aims to generate continuous trajectories of vehicles with improved accuracy and consistency in real time by using various sensors that can provide information about target position. The track results are necessary for anomaly detection in security-related applications and for autonomous vehicle executing appropriate control actions. This thesis addresses the multiple on-road vehicle tracking problem, with greater consideration being given to the use of domain knowledge of on-road vehicle motion.

Fig. 1.1 shows a typical processing loop of a conventional multiple target tracking (MTT) system [2]. Input data, consisting of formated measurements of targets, are

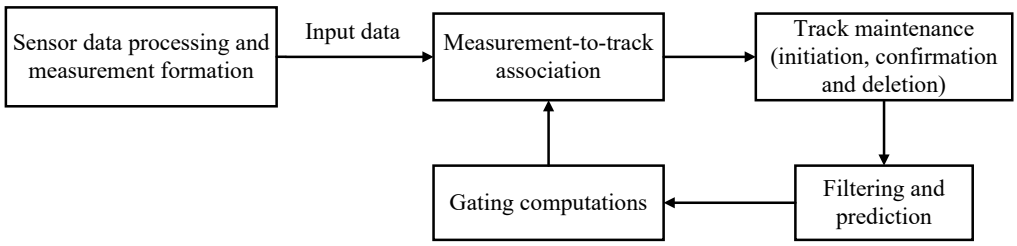


Fig. 1.1: Typical structure of a conventional MTT system.

extracted from raw signal data received by the sensor. The incoming measurements are first considered for updating existing tracks from the previous scan. The Gating step removes unreasonable measurement-to-track parings to reduce computation load before a more refined association algorithm is used to determine final parings. Then, the existing tracks are updated by using a Bayesian filter such as Kalman filter [13] and particle filter (PF) [9] with the associated measurement. Those measurements that are not associated with the existing tracks initiate new tentative tracks. In the track maintenance block, tentative tracks are confirmed when the confirmation criteria are satisfied and also low-quality tracks are terminated.

Radar and cameras are the two most widely-used sensors for airborne ground target tracking. However, both sensors provide measurements of target position in their local coordinates. It is required to convert the measurements into the world coordinate for tracking. However, the converted measurements may be perturbed by bias, which is caused by errors in measurements of sensor states and parameters that are used during conversion. The bias is hard to remove by filtering and can adversely affect gating, data association and tracking results [2]. Thus, bias correction is an indispensable step to mitigate biases in converted measurements before using them to form or update tracks.

Bias correction methods for radar can be found in [16, 20, 25]. Chapter 2 revisits the problem of bias correction for airborne video and a fast but effective bias correction method is proposed. The main idea behind the proposed method is to model the bias in the converted measurement as a function of errors in sensor measurements and decouple it from the true target position in the same way as in [21], based on the imaging model proposed in [1]. Then, the errors in sensor measurements can be estimated by using observations of common targets from two airborne video sensors or jointly estimated with target position during tracking. Once the error estimates are obtained, it can be used to debiase the converted measurements. In [21], the bias is expressed as a highly non-linear function of the errors in sensor measurements. As a result, the numerical search algorithms, such as the genetic algorithm (GA) [7], has to be used to estimate the errors with a high computational cost. In contrast, by assuming that the errors are small, the proposed method reformulates the bias as a linear function of the errors in sensor measurements. Then, two computationally-efficient methods for estimating the errors are proposed.

The motion models of targets that are used in general MTT system are assumed unconstrained and independent of one another. However, this assumption does not hold for on-road vehicle motion. As vehicles move along roads, their motions are subjected to various constraints imposed by those roads. The road map can then be utilized as prior information, improving the tracking performance while reducing target uncertainty and false alarms [6, 11]. Due to traffic volume and limited lane resources, vehicles have to interact with surrounding traffic while moving along roads in order to improve their navigability while maintaining safety. This leads to highly dependent motion across vehicles, a particular difficulty in vehicle tracking.

Though the comprehensive studies have been carried out for using road-map information [14, 19, 8, 23, 5], interactions among vehicles has not yet been well-addressed in vehicle tracking. To the best of our knowledge, the interactions among vehicles are considered in evaluating the probability of candidate tracks in the multiple hypothesis tracking (MHT) algorithm in [18]. However, the motion model of a vehicle in [18] is still assumed independent to other vehicles. In [8], the social force model [12], which has been used in pedestrian tracking for modeling pedestrian motion in the presence of interactions with surroundings by using virtual forces [10, 24, 15], is modified to describe interactions among vehicles and used in vehicle tracking. However, using virtual forces to model interactions lacks any direct physical meaning and therefore may not always be appropriate. For example, it is difficult to model how a vehicle makes lane-changing decisions is influenced by its neighboring vehicles by using virtual forces. In contrast, the microscopic traffic flow (MTF) model [22] has been developed for nearly half a century in the area of traffic theory to specifically describe the dynamics of individual vehicle in the presence of interactions with surrounding traffic. The MTF model has been widely used in many traffic simulators and intelligent transportation systems [17]. Given this gap in the research, in this thesis, a novel tracking algorithm, which integrates the MTF model as motion model of vehicles into a road coordinate system, is proposed to track multiple vehicles on single-lane roads. Then, the proposed algorithm is extended to handle vehicle tracking on multi-lane roads by integrating a lane-changing model for modeling lane-changing behavior of vehicles. A posterior Cramer-Rao lower bound (PCRLB) is also derived to quantify achievable estimation accuracy for tracking multiple vehicles with the use of domain knowledge on both single-lane and multi-lane roads.

1.2 Theme and Objectives of Dissertation

In compliance with the terms and regulations of McMaster University, this dissertation has been assembled by four articles in a *sandwich thesis* format. These articles represent the independent research work of the author of this dissertation, Dan Song.

The articles in the dissertation focuses on multiple on-road vehicle tracking using domain knowledge of on-road vehicle motion. The general research topics include:

1. To propose a bias correction method for airborne video sensor with good performance and efficiency (Paper I).

2. To mathematically formulate a motion model of on-road vehicles based on domain knowledge of on-road vehicle motion (Paper II and III).

3. To exploit domain knowledge of on-road vehicle motion for tracking multiple on-road vehicles and compare the proposed tracking algorithm with conventional MTT algorithms (Paper II, III and IV).

4. To derive the PCRLB for the problem of multiple on-road vehicle tracking using domain knowledge (Paper II and IV).

1.3 Summary of Enclosed Articles

The articles enclosed in this thesis are listed as follows:

1.3.1 Paper I (Chapter 2)

Dan Song, Ratnasingham Tharmarasa, Daly Brown and Thiagalingam Kirubarajan

Airborne Video Geo-registration for Ground Target Tracking, To be submitted to *Signal Processing*.

Preface: This paper proposes a fast but effective bias correction method for airborne video sensor. By assuming that errors in sensor parameter measurements are small, bias in converted measurements of target position is decoupled from the true target position and expressed as a linear function of the errors in sensor measurements. Based on this model, two computationally-efficient methods are proposed to estimate the errors in sensor measurements. The error estimates are then used for bias correction. Simulation results demonstrate the effectiveness and computational-efficiency of the proposed method.

1.3.2 Paper II (Chapter 3)

Dan Song, Ratnasingham Tharmarasa, Thiagalingam Kirubarajan and Xavier N. Fernando

Multi-Vehicle Tracking With Road Maps and Car-Following Models, *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1375–1386, May 2018. (doi: 10.1109/TITS.2017.2723575)

Preface: This paper expands the idea of making use of domain knowledge of on-road vehicle motion for tracking multiple vehicles on single-lane roads. a novel tracking algorithm is proposed, which considers interactions among vehicles by integrating a car-following model into the tracking process with on-road constraints. The notion of car-following clusters is defined to address the motion dependence across vehicles in a group. Then, the states of vehicles in the same cluster are concatenated and recursively updated using a stacked-update strategy. Furthermore, the variable

structure interacting multiple model (VS-IMM) estimator is modified and integrated into the proposed algorithm to handle maneuvers that may violate the integrated car-following model. The PCRLB is also derived for this problem. The superiority of the proposed multiple on-road vehicle tracking algorithm over other state-of-art tracking algorithms are demonstrated through simulations.

1.3.3 Paper III (Chapter 4)

Dan Song, Ratnasingham Tharmarasa, Gongjian Zhou, Mihai C. Florea, Nicolas Duclos-Hindie and Thiagalingam Kirubarajan

Multi-Vehicle Tracking Using Microscopic Traffic Models, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 149–161, Jan. 2019. (doi: 10.1109/TITS.2018.2804894)

Preface: This paper extends the idea of paper II to handle tracking multiple vehicles on multi-lane roads. An extra dimension is added in the road coordinate system to express lanes on roads. The motion of a vehicle on multi-lane roads is decomposed into longitudinal motion and lateral motion, which are described by using a car-following model and a lane-changing model, respectively. Since the longitudinal and lateral motions are mutually dependent, their states are estimated sequentially in a recursive manner. Due to high nonlinearity of the car-following model and discreteness of lateral state, the longitudinal and lateral states are estimated by integrating the unscented Kalman filter (UKF) into the multiple hypothesis tracking (MHT) framework. An adaptive deferred decision logic is proposed to improve the accuracy of the lateral state estimates and thus improve overall performance.

1.3.4 Paper IV (Chapter 5)

Dan Song, Ratnasingham Tharmarasa, Thiagalingam Kirubarajan and Xavier N. Fernando

Multi-vehicle Tracking with Microscopic Traffic Flow Model-based Particle Filtering, Submitted to *Automatica* (second round revision), March 2018.

Preface: This paper continues the work in paper III. Rather than estimate the longitudinal and lateral states of vehicles using the UKF combined with MHT method, respectively, a unified particle filter (PF) framework with a specifically designed computationally-efficient joint sampling method is proposed to jointly estimate the longitudinal and lateral states of vehicles. The recursive maximum likelihood method is integrated into the PF to online estimate unknown parameters in the MTF models. The PCRLB is derived for tracking multiple vehicles on multi-lane roads. Numerical results show that the proposed algorithm requires less prior information while generating more accurate and consistent tracks than the algorithm proposed in paper III.

1.4 Other Articles Written by the Author During PhD

Other articles of the author that are not enclosed in the dissertation are listed below:

1. Dan Song, Wei Wang, Zhenhai Xu, Ziyuan Xiong and Thiagalingam Kirubarajan, "Focused energy delivery with protection for precision electronic warfare", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 3053–3064, December 2016. (doi: 10.1109/TAES.2016.150713)

2. Dan Song, T.Bahadir Sarikaya, Selda Taskin Serkan, Ratnasingham Tharmarasa, Engin Sobaci and Thiagalingam Kirubarajan, "Heavy-tailed sea clutter modeling for shore-based radar detection", *2018 IEEE Radar Conference*, pp. 1504–1509, April 2018. (doi: 10.1109/RADAR.2018.8378789)

3. Ehsan Taghavi, Dan Song, Ratnasingham Tharmarasa, Thiagalingam Kirubarajan, Anne-Claire Boury-Brisset and Bhashyam Balaji, "Object recognition and identification using ESM data", *2016 19th International Conference on Information Fusion*, pp. 1–8, July 2016.

4. Dan Song, Tongyu Ge, Ratnasingham Tharmarasa, Mike McDonald and Thiagalingam Kirubarajan, "Video SAR image fusion using the effective reflection coefficient", Submitted to *Signal Processing*.

Bibliography

[1] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo–location using a fixed–wing miniature air vehicle," *J. Intell. Rob. Syst.*, vol. 47, no. 4, pp. 361–382, Nov. 2006.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA: Artech House, 1999.

[3] X. Cao, X. Jiang, X. Li, and P. Yan, "Correlation-based tracking of multiple targets with hierarchical layered structure," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 90–102, Jan. 2018.

[4] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[5] Y. Chen, V. P. Jilkov, and X. R. Li, "Multilane-road target tracking using radar and image sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 65–80, Jan. 2015.

[6] Y. Cheng and T. Singh, "Efficient particle filtering for road-constrained target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1454–1469, Oct. 2007.

[7] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[8] R. Ding, M. Yu, H. Oh, and W. H. Chen, "New multiple-target tracking strategy using domain knowledge and optimization," *IEEE Trans. Syst. Man. Cybern. Syst.*, vol. 47, no. 4, pp. 605–616, Apr. 2017.

[9] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, no. 656-704, p. 3, Dec. 2009.

[10] P. Feng, W. Wang, S. Dlay, S. M. Naqvi, and J. Chambers, "Social force model based MCMC-OCSVM particle PHD filter for multiple human tracking," *IEEE Trans. Multimedia*, vol. PP, no. 99, pp. 1–15, Dec. 2016.

[11] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous localization and mapping for path-constrained motion," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 541–552, Jun. 2012.

[12] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[14] T. Kirubarajan, Y. Bar-Shalom, and K. R. Pattipati, "Ground target tracking with variable structure IMM estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 26–46, Jan. 2000.

[15] K. Krishnan, X. Chen, R. Tharmarasa, T. Kirubarajan, and M. McDonald, "The social force PHD filter for tracking pedestrians," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 2045–2059, Aug. 2017.

[16] W. Li, H. Leung, and Y. Zhou, "Space-time registration of radar and esm using unscented kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 824–836, Jul. 2004.

[17] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 3, pp. 314–325, Sep. 2005.

[18] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 198–219, Sep. 2013.

[19] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 1, pp. 128–136, Jan. 2002.

[20] E. Taghavi, R. Tharmarasa, T. Kirubarajan, Y. Bar-Shalom, and M. Mcdonald, "A practical bias estimation algorithm for multisensor-multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 1, pp. 2–19, Feb. 2016.

[21] E. Taghavi, D. Song, R. Tharmarasa, T. Kirubarajan, M. McDonald, and B. Balaji, "Geo-registration and geo-location using two airborne video sensors," *IEEE Trans. Aerosp. Electron. Syst.*, To be appear.

[22] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin Heidelberg, Springer-Verlag, 2013.

[23] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[24] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, and J. A. Chambers, "Multi-target tracking and occlusion handling with learned variational bayesian clusters and a social force model," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1320–1335, Mar. 2016.

[25] W. Wu, J. Jiang, W. Liu, X. Feng, L. Gao, and X. Qin, "Augmented state GM-PHD filter with registration errors for multi-target tracking by doppler radars," *Signal Process.*, vol. 120, pp. 117–128, Mar. 2016.

The following chapter is a reproduction of a ready to submit paper to *Signal Processing*, Elsevier:

Dan Song, Ratnasingham Tharmarasa, Daly Brown and Thiagalingam Kirubarajan Airborne Video Geo-registration for Ground Target Tracking, To be submitted to *Signal Processing*.

Chapter 2

Airborne Video Geo-registration

2.1 Abstract

Accurate geo-registration is required in airborne surveillance of ground targets since even a small bias in sensor registration may result in a large error in the converted target geo-location. Thus, this paper addresses the problem of estimating and correcting the sensor biases in target geo-location. The effect of sensor biases on target geo-location is modeled and decoupled from the true target geo-location. By assuming that the biases are small, the effect of biases is separately addressed by a linear term in the converted target geo-location. Based on this model, two computationally-efficient methods are proposed to estimate the biases. The converted target geo-location can then be corrected based on the bias estimates. Simulation results demonstrate the effectiveness of the proposed methods in estimating sensor biases and correcting target geo-locations as well as their high computational efficiency.

2.2 Introduction

Airborne surveillance of ground targets plays an important role in various civil and military applications [28, 23]. The objective here is to accurately localize ground targets in world or inertial coordinates, i.e. generate geo-locations, in real-time. The commonly used sensors for such a task include the ground moving target indicator (GMTI) radar [9] and video sensors such as electro-optical/infrared camera [20], among which video sensors are increasingly used due to their portability and accuracy.

In a surveillance system, the measurements provided by sensors are fed into a tracker, in which the measurements are associated and filtered across time to generate continuous trajectories of ground targets. Various methods have been developed in recent years for ground target tracking [11, 30, 28, 27, 20, 22, 21]. Though these tracking algorithms can reduce noises in sensor measurements, the systematic errors in measurements, which mainly originate from sensor biases, cannot be removed by these algorithms. Unmitigated sensor biases can adversely affect , filtering and fusion results [3]. Thus, accurate geo-registration is an indispensable step in airborne surveillance to mitigate the biases in sensor measurements.

Geo-registration for radar [13, 24, 29] is different from that of video sensors. This study focuses on geo-registration for video sensors on airborne platforms. A video sensor can be geo-registered by comparing live images with pre-existing geo-referenced imagery or a terrain model [12, 5, 18]. In [19], instead of using a reference terrain model, a terrain model is built online by dense and coherent stereo matching, and used for video geo-registration. The cross-modality across multiple images is exploited for image geo-registration in [16]. By controlling the flight path of an airborne platform along a circular orbit around the target, the cumulative bias errors are demonstrated

to be uncorralated with the target position and thus a minimum variance approach is used to address geo-registration in [2]. Based on the imaging model proposed in [2], the geo-registration problem is addressed in a systematic manner in [25], where the bias in the converted target geo-location is modeled and decoupled from the true target geo-location. Then a pseudo-measurement, which depends only on the biases in sensor parameters only, is proposed by making use of the information from the common targets surveilled by two airborne video sensors in their respective fields of view, and then used to estimate the biases. Due to the high non-linearity of the derived pseudo-measurement, the biases have to be estimated using numerical search algorithms, such as the genetic algorithm (GA) [6]. Though this method can estimate and remove the biases effectively, its computational cost is significantly high.

To address the high computational cost of the method proposed in [25], this paper reformulates the modeling of biases in converted target geo-location. By assuming that the biases in sensor measurements are small, a novel decoupled method, in which the effect of sensor biases on the converted target geo-location is decoupled from the true target geo-location and expressed by a linear term, is proposed. Based on the linear decoupled reformulation, two computationally-efficient methods for estimating the sensor biases are then proposed by (a) estimating the biases jointly with target geo-locations within a unified estimation framework, (b) estimating the biases separately by using the observations of common targets from multiple airborne video sensors. Once the estimates of sensor biases are obtained, the converted target geo-location can be debiased. The validity of the proposed geo-registration methods is demonstrated through simulations in comparison with the method proposed in [25] and against a lower bound on the estimation errors with a bias-free sensor.

The rest of the paper is organized as follows: In Section 2.3, the modeling of an airborne video sensor, based on which the target geo-location is converted from its pixel location in an image, is briefly reviewed. The bias in the converted target geo-location is reformulated and decoupled from the true target geo-location in Section 2.4. Two methods for bias estimation and geo-registration are proposed in Section 2.5. In Section 2.6, the performance of the proposed methods is evaluated and compared in simulated scenarios. Conclusions are given in Section 2.7.

2.3 Target Geo-location by an Airborne Video Sensor

In this section, an imaging model for an airborne video sensor [2] is briefly reviewed to give the necessary background to extract a measurement of target geo-location given the pixel location of the target in an image and sensor parameters.

2.3.1 Coordinate Frames

This model involves a series of coordinate frames, as shown in Fig. 2.1 and Fig. 2.2. Their definitions are given as follows:

Inertial frame (X_I, Y_I, Z_I)

The inertial frame is a fixed frame originating from a fixed point with X_I, Y_I and Z_I pointing to the North, the East and the center of the earth, respectively.

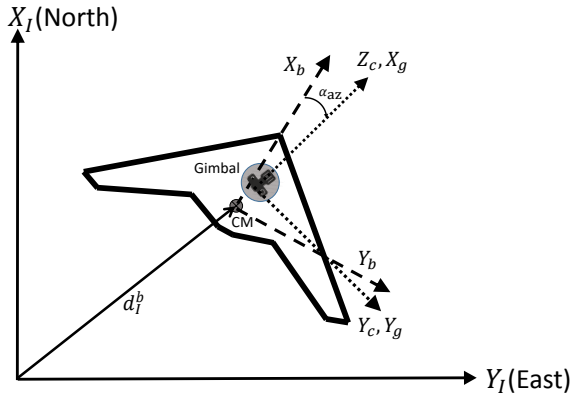


Fig. 2.1: Lateral view of coordinate frames [2]

Body frame (X_b, Y_b, Z_b)

The body frame originates at the center of mass (COM) of the airborne platform with X_b, Y_b and Z_b pointing out the nose, the right wing and the belly of the platform, respectively.

Gimbal frame (X_g, Y_g, Z_g)

The gimbal frame originates at the gimbal rotation center. Here, X_g is the direction along the optical axis of the video sensor and Y_g and Z_g point right and down in the image plane, respectively.

Camera frame (X_c, Y_c, Z_c)

The camera frame, defined to represent the directions in images, originates at the optical center of the video sensor with Z_c directed along the optical axis while X_c and Y_c point up and right in the image plane, respectively.

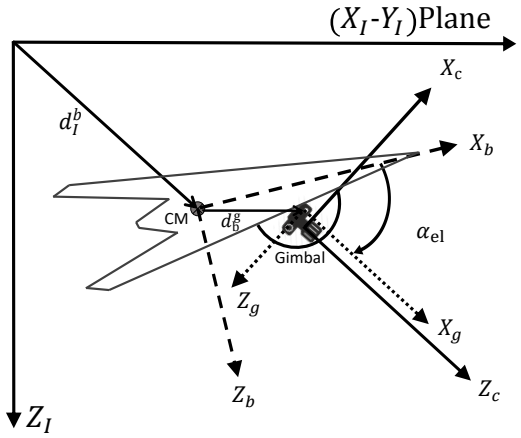


Fig. 2.2: Longitudinal view of coordinate frames [2]

2.3.2 Transformations Between Coordinate Frames

The transformations between these coordinate frames can be expressed using a unified homogeneous transformation matrix

$$\begin{bmatrix} v^j \\ 1 \end{bmatrix} = T_i^j \begin{bmatrix} v^i \\ 1 \end{bmatrix} \tag{2.1}$$

where v_i and v_j denote the vector v represented in frame i and j, respectively, and T_i^j is the homogeneous transformation matrix from frame i to frame j given by

$$T_i^j = \begin{bmatrix} R_i^j & -d_i^j \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2.2}$$

Here, R_i^j corresponds to the rotation operations that align the frame i with the frame j and d_i^j is the vector from the origin of the frame i to the origin of the frame j represented in frame j. It can be verified that the transformation matrix from frame

j to frame i, as the inverse of T_j^i, is then given by

$$T_j^i = T_i^{j-1} = \begin{bmatrix} {R_i^j}^{\mathrm{T}} & {R_i^j}^{\mathrm{T}} d_i^j \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.3}$$

where $(\cdot)^{\mathrm{T}}$ denotes the transpose.

Inertial-to-body frame transformation

The transformation from inertial frame to body frame depends on the position and the orientation of the airborne platform. Denote the position of the platform's COM as $[x_p \ y_p \ h_p]^{\mathrm{T}}$, where x_p and y_p represent the North and East locations of the platform, respectively, while h_p represents its altitude. The platform's Euler angles, yaw, pitch and roll, are denoted by ψ, θ and ϕ, respectively. The transformation matrix T_I^b is then given by

$$T_I^b = \begin{bmatrix} R_I^b & -d_I^b \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.4}$$

where

$$R_I^b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.5}$$

and

$$d_I^b = R_I^b \begin{bmatrix} x_p & y_p & -h_p \end{bmatrix}^{\mathrm{T}}. \tag{2.6}$$

Body-to-gimbal frame transformation

The transformation matrix from body frame to gimbal frame can be written as

$$T_b^g = \begin{bmatrix} R_b^g & -d_b^g \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.7}$$

where

$$\begin{aligned} R_b^g &= \begin{bmatrix} \cos\alpha_{\text{el}} & 0 & \sin\alpha_{\text{el}} \\ 0 & 1 & 0 \\ -\sin\alpha_{\text{el}} & 0 & \cos\alpha_{\text{el}} \end{bmatrix} \begin{bmatrix} \cos\alpha_{\text{az}} & \sin\alpha_{\text{az}} & 0 \\ -\sin\alpha_{\text{az}} & \cos\alpha_{\text{az}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\alpha_{\text{el}}\cos\alpha_{\text{az}} & \cos\alpha_{\text{el}}\sin\alpha_{\text{az}} & \sin\alpha_{\text{el}} \\ -\sin\alpha_{\text{az}} & \cos\alpha_{\text{az}} & 0 \\ -\sin\alpha_{\text{el}}\cos\alpha_{\text{az}} & -\sin\alpha_{\text{el}}\sin\alpha_{\text{az}} & \cos\alpha_{\text{el}} \end{bmatrix} \end{aligned} \tag{2.8}$$

with α_{az} and α_{el} denoting the azimuth angle of rotation about Z_b and the elevation angle of rotation about Y_b, respectively, that aligns X_b with X_g. Here, d_b^g can be obtained from the location of the gimbal rotation center with respect to the platform's COM d_g^b as

$$d_b^g = -R_b^g d_g^b \tag{2.9}$$

since d_g^b is fixed when it is resolved in the body frame.

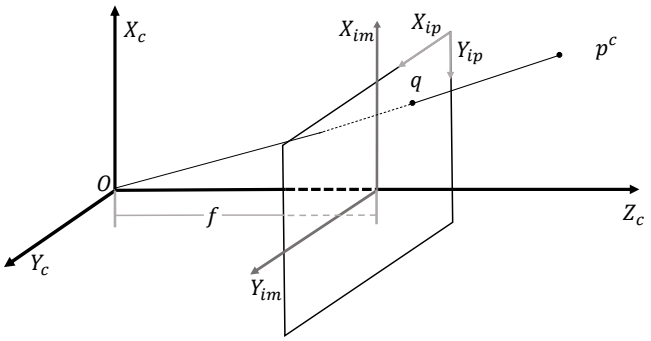


Fig. 2.3: Camera projection model [2]

Gimbal-to-camera frame transformation

The transformation matrix from gimbal frame to camera frame is given by

$$T_g^c = \begin{bmatrix} R_g^c & -d_g^c \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.10}$$

where d_g^c is the location of the gimbal rotation center with respect to the camera optical center. As shown in Fig. 2.2, we have X_c and Z_c aligning with $-Z_g$ and X_g, respectively. Thus, R_g^c is given by

$$R_g^c = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.11}$$

2.3.3 Camera Model

A simple camera projection model [2, 25], as shown in Fig. 2.3, is used in this paper to model the relationship between positions in camera frame and their projections on the image plane in pixels. Let $p^c = [x_p^c \; y_p^c \; z_p^c \; 1]^{\mathrm{T}}$ denote a position in camera frame.

The position of its projection q in the image frame (X_{im}, Y_{im}) is determined by

$$\begin{bmatrix} x_q^{im} \\ y_q^{im} \end{bmatrix} = \begin{bmatrix} \frac{f}{p_z^c} p_x^c & 0 \\ 0 & \frac{f}{p_z^c} p_y^c \end{bmatrix} \tag{2.12}$$

where f is the focal length of the camera. Then, the position of q on the image plane in pixels is given by [26]

$$x_q^{ip} = -\frac{y_q^{im} - 0_y}{S_x} \tag{2.13}$$

$$y_q^{ip} = \frac{x_q^{im} - 0_x}{S_y} \tag{2.14}$$

where 0_x and 0_y denote the offsets of the upper-left hand corner of image with respect to the image center in X_{im} and Y_{im}, respectively, and S_x and S_y are the sizes of a pixel in X_{ip} and Y_{ip}, respectively. To unify representations, let $q^{ip} = [x_q^{ip}\ y_q^{ip}\ 1\ 1]^{\mathrm{T}}$ denote the position of q in image frame in pixels. Combining (2.12)–(2.14), the relationship between q^{ip} and p^c is given by

$$\Lambda q^{ip} \triangleq C p^c = \begin{bmatrix} 0 & f_x & 0_x & 0 \\ -f_y & 0 & 0_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} p^c \tag{2.15}$$

where $f_x \triangleq f/S_x$, $f_y \triangleq f/S_y$ and

$$\Lambda = \begin{bmatrix} z_p^c I_{3\times 3} & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.16}$$

Here, $I_{n\times n}$ denotes an n by n identity matrix.

2.3.4 Target Geo-location

Based on the frame transformations and the camera model described in the previous sections, once the pixel location of a target in the image, q^{ip}, is given, its position in the inertial frame can be obtained by

$$\begin{aligned} p^I &= \left(CT_g^c T_b^g T_I^b\right)^{-1} \Lambda q^{ip} \\ &= T_b^I T_g^b T_c^g C^{-1} \Lambda q^{ip} \end{aligned} \tag{2.17}$$

given the sensor parameters if z_p^c in Λ is known. Here, z_p^c, referred to as the image depth, can be estimated based on the flat earth assumption [15], which is valid when the coverage area of the camera is not large. Let p_c^I denote the position of the camera optical center resolved in the inertial frame, which is given by

$$p_c^I = T_b^I T_g^b T_c^g p_c^c \tag{2.18}$$

where p_c^c is the location of the camera optical center resolved in the camera frame given by $p_c^c = [0\ 0\ 0\ 1]^{\mathrm{T}}$ as it is the origin of the camera frame. The location of q resolved in the inertial frame can be similarly obtained by

$$q^I = T_b^I T_g^b T_c^g C^{-1} q^{ip}. \tag{2.19}$$

Then, based on the flat earth assumption and the camera projection geometry shown in Fig. 2.3, the image depth z_p^c is given by

$$z_p^c = \frac{z_c^I}{z_c^I - z_q^I} \tag{2.20}$$

where z_c^I and z_q^I are the z-components of p_c^I and q^I, respectively. Note that a similar method can be derived when the flat earth assumption does not hold and the terrain data is known. In this case, the intersection of earth and the extension line from p_c^I to q^I can be obtained by using the method in [8]. Denote the height of the intersection as h_p. The image depth z_p^c is then given by

$$z_p^c = \frac{z_c^I + h_p}{z_c^I - z_q^I}. \tag{2.21}$$

2.4 Modeling of Bias in Video Measurements

Targets can be localized in the world or inertial frame based on (2.17) given their pixel positions in the image and sensor parameters. However, some of the sensor parameters, such as the position and the orientation of the platform and the orientation of the gimbal, are time-varying and have to be measured by on-board sensors in real time. For example, the North and East locations of the platform can be measured by the Global Positioning System (GPS) sensor and the altitude of the platform can be measured by the barometric pressure sensor. Due to sensor noise and geometric uncertainties, these measurements are inevitably perturbed by zero-mean noises and biases, which may cause large errors in the converted target geo-location. Though the effect of zero-mean noises on the converted geo-location can be removed

by filtering algorithms [1], the systematic biases are not handled by filtering. Thus, this paper focuses on estimating and correcting the constant biases in measurements of sensor parameters in order to make the converted target geo-location more accurate.

Since the GPS sensor has a high accuracy [17] and it has been shown that the accuracy of converted target geo-locations is not significantly affected by the biases in pitch and altitude measurements [2], the biases considered in this paper include those in the measurements of roll, yaw, gimbal azimuth and elevation angles. Platforms used in airborne surveillance often fly at a nearly constant altitude, implying that the pitch angle is close to zero. From Fig. 2.1 and Fig. 2.2, it can be seen that in this case, the roll and yaw axes of the platform are closed to the elevation and azimuth axes of the gimbal, respectively. Thus, the effects of biases in roll and yaw measurements are numerically indistinguishable from those of biases in gimbal elevation and azimuth measurements. For simplicity, the biases in roll and yaw measurements are considered merged into the biases in gimbal elevation and azimuth and corrected together with them. As a result, the bias vector that we need to estimate is reduced to $b = [\bar{\alpha}_{\text{az}}\ \bar{\alpha}_{\text{el}}]^{\text{T}}$, where $\bar{\alpha}_{\text{az}}$ and $\bar{\alpha}_{\text{el}}$ denote the biases in gimbal azimuth and elevation measurements, respectively.

In [25], the effect of the bias vector is decoupled from true target geo-location and expressed by a highly nonlinear term. Thus, numerical search algorithms, such as the GA algorithm [6], have to be used to estimate the bias vector. In this section, the bias in the converted target geo-location is reformulated and a novel linearly decoupled method is proposed.

Given the measurements of those time-varing sensor parameters and target pixel location in the image, the measurement of target geo-location z, based on (2.17), is

given by

$$z = \hat{T}_b^I \hat{T}_g^b q^g \tag{2.22}$$

where $\hat{T}_i^j$ denotes the estimate of transformation matrix T_i^j given the measurements of sensor parameters and

$$\begin{aligned} q^g &\triangleq T_c^g C^{-1} \Lambda q^{ip} \\ &= [x_q^g \; y_q^g \; z_q^g \; 1]^{\mathrm{T}}. \end{aligned} \tag{2.23}$$

The bias vector only affects the transformation matrix from gimbal frame to body frame, $\hat{T}_g^b$, which becomes

$$\hat{T}_g^b = \begin{bmatrix} \hat{R}_g^b & -d_g^b \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.24}$$

where

$$\hat{R}_b^g = \begin{bmatrix} \cos\hat{\alpha}_{\mathrm{el}} \cos\hat{\alpha}_{\mathrm{az}} & -\sin\hat{\alpha}_{\mathrm{az}} & -\sin\hat{\alpha}_{\mathrm{el}} \cos\hat{\alpha}_{\mathrm{az}} \\ \cos\hat{\alpha}_{\mathrm{el}} \sin\hat{\alpha}_{\mathrm{az}} & \cos\hat{\alpha}_{\mathrm{az}} & -\sin\hat{\alpha}_{\mathrm{el}} \sin\hat{\alpha}_{\mathrm{az}} \\ \sin\hat{\alpha}_{\mathrm{el}} & 0 & \cos\hat{\alpha}_{\mathrm{el}} \end{bmatrix} \tag{2.25}$$

given the measurement of gimbal azimuth, $\hat{\alpha}_{\mathrm{az}} = \alpha_{\mathrm{az}} + \bar{\alpha}_{\mathrm{az}}$ and the measurement of gimbal elevation, $\hat{\alpha}_{\mathrm{el}} = \alpha_{\mathrm{el}} + \bar{\alpha}_{\mathrm{el}}$. Expanding each term in $\hat{R}_g^b$ and applying some approximations, namely, $\sin\bar{\alpha}_{\mathrm{az}} \approx \bar{\alpha}_{\mathrm{az}}$, $\sin\bar{\alpha}_{\mathrm{el}} \approx \bar{\alpha}_{\mathrm{el}}$, $\cos\bar{\alpha}_{\mathrm{az}} \approx 1$, $\cos\bar{\alpha}_{\mathrm{el}} \approx 1$, $\sin\bar{\alpha}_{\mathrm{az}} \sin\bar{\alpha}_{\mathrm{el}} \approx 0$ and $\cos\bar{\alpha}_{\mathrm{az}} \cos\bar{\alpha}_{\mathrm{el}} \approx 1$ by assuming that $\bar{\alpha}_{\mathrm{az}}$ and $\bar{\alpha}_{\mathrm{el}}$ are close to

zero, $\hat{R}_g^b$ can be approximately decoupled as

$$\hat{R}_g^b \approx R_g^b + R_{g,\text{bias}}^b \tag{2.26}$$

where the terms in $R_{g,\text{bias}}^b$ are given by

$$R_{g,\text{bias}}^b(1,1) = -\bar{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{az}} - \bar{\alpha}_{\text{az}} \cos\hat{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{az}} \tag{2.27}$$

$$R_{g,\text{bias}}^b(1,2) = -\bar{\alpha}_{\text{az}} \cos\hat{\alpha}_{\text{az}} \tag{2.28}$$

$$R_{g,\text{bias}}^b(1,3) = \bar{\alpha}_{\text{az}} \sin\hat{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{az}} - \bar{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{az}} \tag{2.29}$$

$$R_{g,\text{bias}}^b(2,1) = \bar{\alpha}_{\text{az}} \cos\hat{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{az}} - \bar{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{az}} \tag{2.30}$$

$$R_{g,\text{bias}}^b(2,2) = -\bar{\alpha}_{\text{az}} \sin\hat{\alpha}_{\text{az}} \tag{2.31}$$

$$R_{g,\text{bias}}^b(2,3) = -\bar{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{az}} - \bar{\alpha}_{\text{az}} \sin\hat{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{az}} \tag{2.32}$$

$$R_{g,\text{bias}}^b(3,1) = \bar{\alpha}_{\text{el}} \cos\hat{\alpha}_{\text{el}} \tag{2.33}$$

$$R_{g,\text{bias}}^b(3,2) = 0 \tag{2.34}$$

$$R_{g,\text{bias}}^b(3,3) = -\bar{\alpha}_{\text{el}} \sin\hat{\alpha}_{\text{el}} \tag{2.35}$$

Substituting (2.26) into (2.24), $\hat{T}_g^b$ can be rewritten as

$$\hat{T}_g^b = T_g^b + T_{g,\text{bias}}^b \tag{2.36}$$

where

$$T_{g,\text{bias}}^b = \begin{bmatrix} R_{g,\text{bias}}^b & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{2.37}$$

Then, the measurement of target geo-location in (2.22) can be expressed as

$$z = \hat{T}_b^I T_g^b q^g + \hat{T}_b^I T_{g,\text{bias}}^b q^g. \tag{2.38}$$

Multiplying out the second term in (2.38) and substituting (2.27)–(2.35) into the result, the second term can be represented as a linear function of the bias vector b as follows:

$$\hat{T}_b^I T_{g,\text{bias}}^b q^g = \hat{R}_b^I U b \tag{2.39}$$

where $\hat{R}_b^I = \hat{R}_I^{b^{\mathrm{T}}}$ is obtained by substituting Euler angle measurements into (2.5) and $U = [U_1\ U_2]$ with

$$U_1 = \begin{bmatrix} -\cos\hat{\alpha}_{\text{el}}\sin\hat{\alpha}_{\text{az}} & -\cos\hat{\alpha}_{\text{az}} & \sin\hat{\alpha}_{\text{el}}\sin\hat{\alpha}_{\text{az}} \\ \cos\hat{\alpha}_{\text{el}}\cos\hat{\alpha}_{\text{az}} & -\sin\hat{\alpha}_{\text{az}} & -\sin\hat{\alpha}_{\text{el}}\cos\hat{\alpha}_{\text{az}} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_q^g \\ y_q^g \\ z_q^g \end{bmatrix} \tag{2.40}$$

and

$$U_2 = \begin{bmatrix} -\sin\hat{\alpha}_{\text{el}}\cos\hat{\alpha}_{\text{az}} & 0 & -\cos\hat{\alpha}_{\text{el}}\cos\hat{\alpha}_{\text{az}} \\ -\sin\hat{\alpha}_{\text{el}}\sin\hat{\alpha}_{\text{az}} & 0 & -\cos\hat{\alpha}_{\text{el}}\sin\hat{\alpha}_{\text{az}} \\ \cos\hat{\alpha}_{\text{el}} & 0 & -\sin\hat{\alpha}_{\text{el}} \end{bmatrix} \begin{bmatrix} x_q^g \\ y_q^g \\ z_q^g \end{bmatrix}. \tag{2.41}$$

As discussed in the previous section, the biases in $\hat{T}_b^I$ (roll and yaw measurements) are transferred into the bias vector b and the first term in (2.38), $\hat{T}_b^I T_g^b q^g$, is approximately equal to the true target geo-location p^I. Since (2.22) is linear and the

noises in measurements of sensor parameters are small and follow zero-mean Gaussian distributions, the effect of the noises on the converted target geo-location can be approximately modeled by adding a Gaussian noise w into (2.22). Then, the measurement of target geo-location can be rewritten as

$$z = p^I + \hat{R}_b^I U b + w. \tag{2.42}$$

2.5 Bias Estimation and Geo-Registration

Equation (2.42) shows that the converted measurement (2.22) is linear with the true target geo-location and the bias vector. Thus, the bias vector can be estimated together with the target geo-location over a period of time or estimated separately by canceling out the target geo-location terms for the common targets that are observed by multiple airborne sensors. The two methods are briefly introduced below. The following derivations assume that there are M airborne video sensors observing N ground targets in the scenario.

2.5.1 Geo-registration while Tracking

Let $x_i = [p_i^I \; v_i^I]^{\mathrm{T}}$ denote the kinematic state of the ith target, where p_i^I and v_i^I are the position and velocity of the target, respectively. Stacking the kinematic states of the targets and bias vectors of the sensors, the vector $X = [x_1^{\mathrm{T}} \; \cdots \; x_N^{\mathrm{T}} \; b_1^{\mathrm{T}} \; \cdots \; b_M^{\mathrm{T}}]^{\mathrm{T}}$ is required to be inferred at each time step k. Given the dynamics model of each target as [14]

$$x_i(k) = F_i(k-1)x_i(k-1) + G_i(k-1)v_i(k-1) \tag{2.43}$$

where $v_i(k-1)$ denotes the process noise sequence, the evolution of X can be described by

$$X(k) = \mathbf{F}(k-1)X(k-1) + \mathbf{G}(k-1)\mathbf{v}(k-1) \tag{2.44}$$

where $\mathbf{F}(k-1) = \text{diag}(F_1(k-1), \cdots, F_N(k-1), I_{2M\times 2M})$,

$$\mathbf{G}(k-1) = \begin{bmatrix} G_1(k-1) & & \\ & \ddots & \\ & & G_N(k-1) \\ & \mathbf{0} & \end{bmatrix} \tag{2.45}$$

and $\mathbf{v}(k-1) = [v_1(k-1)^{\mathrm{T}} \ \cdots \ v_N(k-1)^{\mathrm{T}}]^{\mathrm{T}}$. Here, diag$(\cdot)$ denotes the block diagonal matrix construction operator. Based on (2.42), the measurement that originates from the jth target and is observed by the ith sensor at time step k is given by

$$z_{ij}(k) = H_x x_j(k) + \hat{R}^I_{b,i}(k)U_{ij}(k)b_i + w_{ij}(k) \tag{2.46}$$

where $H_x = [I_{3\times 3}\ \mathbf{0}]$, $\hat{R}^I_{b,i}$ is the matrix $\hat{R}^I_b$ for the ith sensor and $U_{ij}(k)$ is the matrix U corresponding to the ith sensor observing the jth target. Then, $z_{ij}(k)$ can be expressed as a function of $X(k)$ as follows:

$$z_{ij}(k) \triangleq H_{ij}(k)X(k) + w_{ij}(k) \tag{2.47}$$

where $H_{ij}(k)$ is obtained by putting H_x and $\hat{R}^I_{b,i}(k)U_{ij}(k)$ to the positions corresponding to $x_j(k)$ and b_i in $X(k)$, respectively. For example, $H_{11}(k)$ is given by

$$H_{11}(k) = \begin{bmatrix} H_x & \mathbf{0} & \cdots & \mathbf{0} & \hat{R}^I_{b,1}(k)U_{11}(k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \tag{2.48}$$

When a new measurement becomes available, substituting (2.44) and (2.47) into the Kalman filter equations [10], the estimate of $X(k)$ and its associated covariance matrix $P(k|k)$ can be recursively updated. The details on the Kalman filter can be found in [1].

2.5.2 Decoupled Geo-registration

When there are common targets observed by multiple airborne sensors simultaneously, a pseudo-measurement that addresses only the effect of biases can be derived by subtracting the converted measurements (2.22) originating from the common target and observed by two different sensors. Based on (2.46), the pseudo-measurement corresponding to the jth target observed by the i_1th and i_2th sensors can be expressed as

$$\begin{aligned} z^{i_1 i_2}_{b,j}(k) &= z_{i_1 j}(k) - z_{i_2 j}(k) \\ &= \hat{R}^I_{b,i_1}(k)U_{i_1 j}(k)b_{i_1} - \hat{R}^I_{b,i_2}(k)U_{i_2 j}(k)b_{i_2} + w_{i_1 j} - w_{i_2 j}. \end{aligned} \tag{2.49}$$

Stacking bias vectors of the sensors, $B = [b_1^{\mathrm{T}} \ \cdots \ b_2^{\mathrm{T}}]^{\mathrm{T}}$, which is required to be inferred, $z^{i_1 i_2}_{b,j}(k)$ can be rewritten as

$$z^{i_1 i_2}_{b,j}(k) = H^{i_1 i_2}_{b,j}(k)B + w^{i_1 i_2}_j(k) \tag{2.50}$$

where $H_{b,j}^{i_1 i_2}(k)$ is obtained by putting $\hat{R}_{b,i_1}^{I}(k)U_{i_1 j}(k)$ and $\hat{R}_{b,i_2}^{I}(k)U_{i_2 j}(k)$ to the positions corresponding to b_{i_1} and b_{i_2}, respectively, and $w_j^{i_1 i_2}(k) = w_{i_1 j}(k) - w_{i_2 j}(k)$. By assuming that $w_{i_1 j}(k)$ is independent of $w_{i_2 j}(k)$, $w_j^{i_1 i_2}(k)$ is also a zero-mean Gaussian noise with covariance $R_j^{i_1 i_2}(k) = R_{i_1 j}(k) + R_{i_2 j}(k)$, where $R_{ij}(k)$ is the covariance of $w_{ij}(k)$. Note that when a target is observed by $M' \geq 2$ sensors simultaneously, $M'-1$ pseudo-measurements can be generated by subtracting the measurement observed by a common sensor from the measurements observed by the other $M'-1$ sensors, respectively. The correlation between the noises in these pseudo-measurements then need to be considered. For example, assuming that the jth target is observed by the i_1th, i_2th and i_3th sensors simultaneously and that the i_1th sensor is picked up as the common sensor, two pseudo-measurements $z_{b,j}^{i_2 i_1}(k)$ and $z_{b,j}^{i_3 i_1}(k)$ are generated. The correlation between $w_j^{i_2 i_1}(k)$ and $w_j^{i_3 i_1}(k)$ is then given by $R_{i_1 j}(k)$. Now, the stacked bias vector B can be estimated based on the pseudo-measurements by using the recursive or batch least squares method [1].

Once the estimate of bias vector, $\hat{b} = [\hat{\bar{\alpha}}_{\text{az}}\ \hat{\bar{\alpha}}_{\text{el}}]^{\text{T}}$, is obtained, the converted measurement of target geo-location (2.22) can be debiased by replacing $\hat{\alpha}_{\text{az}}$ and $\hat{\alpha}_{\text{el}}$ in $\hat{R}_b^g$ (2.25) by $\hat{\alpha}_{\text{az}} - \hat{\bar{\alpha}}_{\text{az}}$ and $\hat{\alpha}_{\text{el}} - \hat{\bar{\alpha}}_{\text{el}}$, respectively.

2.6 Simulation Results

In this section, the performances of the proposed methods are evaluated and compared with that of the method for airborne video geo-registration [25] through simulations.

Table 2.1: Initial states and standard deviations of process noise for platforms and targets with respect to the inertial frame

Parameter	Platform 1	Platform 2	Target 1	Target 2
Position	$\begin{bmatrix} 0\text{m} \\ 0\text{m} \\ -300\text{m} \end{bmatrix}$	$\begin{bmatrix} 400\text{m} \\ 400\text{m} \\ -300\text{m} \end{bmatrix}$	$\begin{bmatrix} 100\text{m} \\ 100\text{m} \\ 0\text{m} \end{bmatrix}$	$\begin{bmatrix} 300\text{m} \\ 200\text{m} \\ 0\text{m} \end{bmatrix}$
Speed	$8\,\text{m/s}$	$8\,\text{m/s}$	$10\,\text{m/s}$	$10\,\text{m/s}$
Heading	15°	-120°	20°	-105°
Elevation	2°	2°	0°	0°
$\sigma_v\,(\text{m/s}^2)$	$\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.25 \\ 0.25 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.25 \\ 0.25 \\ 0 \end{bmatrix}$

2.6.1 Simulation Scenario

The scenario consists of two airborne sensors and two ground targets moving in their overlapping fields of view. The motions of the airborne platforms and targets are generated based on the nearly constant velocity model with the initial states and standard deviation of process noise shown in Table 2.1. Other platform parameters and their on-board camera parameters are given in Table 2.2 and Table 2.3, respectively. The video detections in 100 frames at a rate of five frames per second (fps) are then generated by assuming that the centroids of the targets have been detected by image processing algorithms [7, 4, 31]. To make the simulations more realistic, it is assumed that zero-mean noises and biases are generally present in the measurements of sensor parameters. The standard deviation of platform position measurements is 2m in each dimension and the standard deviation of all angle measurements ($\psi, \theta, \phi, \alpha_{az}$, α_{el}) is 0.2°. The values of biases in measurements of yaw, pitch and roll of each platform

Table 2.2: Platform parameters

Parameter	ψ	θ	ϕ	α_{az}	α_{el}	d_g^b	d_g^c
Platform 1	15°	2°	2°	80°	80°	$\begin{bmatrix} 0.05\text{m} \\ 0.05\text{m} \\ 0.02\text{m} \end{bmatrix}$	$\begin{bmatrix} 0.01\text{m} \\ -0.01\text{m} \\ -0.05\text{m} \end{bmatrix}$
Platform 2	−120°	2°	2°	−45°	60°	$\begin{bmatrix} 0.05\text{m} \\ 0.05\text{m} \\ 0.02\text{m} \end{bmatrix}$	$\begin{bmatrix} 0.01\text{m} \\ -0.01\text{m} \\ -0.05\text{m} \end{bmatrix}$

Table 2.3: On-board camera parameters (in pixels)

Parameter	Camera 1	Camera 2
f_x	548	548
f_y	556	556
O_x	415	225
O_y	520	498

are randomly selected between $-0.5°$ and $0.5°$.

2.6.2 Numerical Results

The performances of the methods are evaluated in four different sets of values for biases in gimbal azimuth and elevation measurements. The details of the test sets are given in Table 2.4. For each test set, the performance measures are obtained over one hundred Monte Carlo runs. The method proposed in [25] estimates the biases using the GA algorithm. Thus, we refer to this method as the GA-based method. The GA-based method is evaluated with the configuration in [25] and two different numbers of generations, namely, $N_g = 5$ and $N_g = 20$, are experimented with. The second proposed method in this paper, which implements geo-registration separately,

Table 2.4: Test sets and corresponding biases in gimbal azimuth and elevation

Test set	$\bar{\alpha}_{\mathrm{az},1}$	$\bar{\alpha}_{\mathrm{el},1}$	$\bar{\alpha}_{\mathrm{az},2}$	$\bar{\alpha}_{\mathrm{el},2}$
1	$2°$	$-2.5°$	$1.5°$	$2°$
2	$-1.5°$	$2°$	$-1.5°$	$1°$
3	$2°$	$2°$	$2°$	$2°$
4	$-2°$	$-2°$	$-2°$	$-2°$

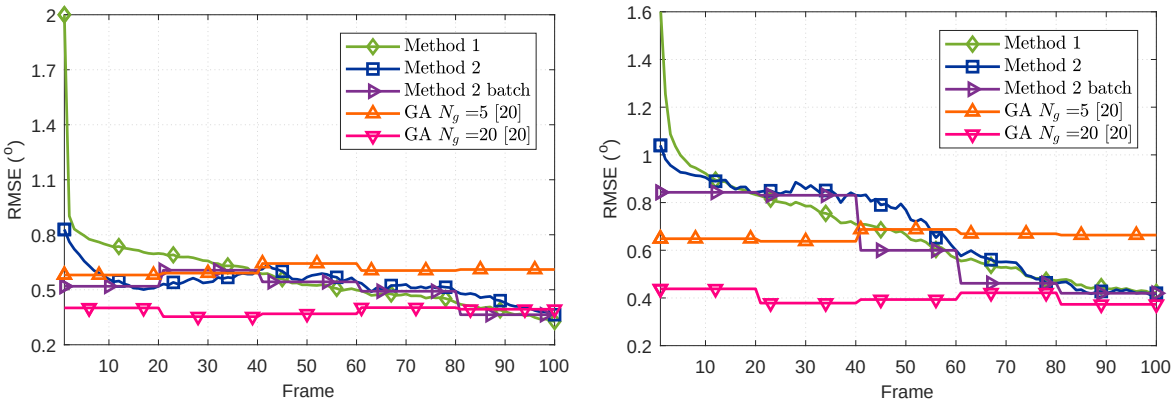
Table 2.5: RMSE of gimbal azimuth and elevation bias estimates (in degrees)

Method	Test set 1				Test set 2				Test set 3				Test set 4			
	$\hat{\bar{\alpha}}_{\mathrm{az},1}$	$\hat{\bar{\alpha}}_{\mathrm{el},1}$	$\hat{\bar{\alpha}}_{\mathrm{az},2}$	$\hat{\bar{\alpha}}_{\mathrm{el},2}$	$\hat{\bar{\alpha}}_{\mathrm{az},1}$	$\hat{\bar{\alpha}}_{\mathrm{el},1}$	$\hat{\bar{\alpha}}_{\mathrm{az},2}$	$\hat{\bar{\alpha}}_{\mathrm{el},2}$	$\hat{\bar{\alpha}}_{\mathrm{az},1}$	$\hat{\bar{\alpha}}_{\mathrm{el},1}$	$\hat{\bar{\alpha}}_{\mathrm{az},2}$	$\hat{\bar{\alpha}}_{\mathrm{el},2}$	$\hat{\bar{\alpha}}_{\mathrm{az},1}$	$\hat{\bar{\alpha}}_{\mathrm{el},1}$	$\hat{\bar{\alpha}}_{\mathrm{az},2}$	$\hat{\bar{\alpha}}_{\mathrm{el},2}$
Method 1	0.15	0.59	0.27	0.31	0.44	0.84	0.20	0.21	0.36	0.51	0.44	0.19	0.55	0.36	0.24	0.42
Method 2	0.28	0.51	0.38	0.30	0.33	0.88	0.27	0.18	0.35	0.52	0.44	0.24	0.31	0.39	0.30	0.33
Method 2 batch	0.28	0.51	0.38	0.30	0.33	0.88	0.27	0.18	0.35	0.52	0.44	0.24	0.31	0.39	0.30	0.33
GA, $N_g = 5$	0.69	0.60	0.65	0.50	0.78	0.73	0.62	0.52	0.57	0.60	0.56	0.59	0.67	0.53	0.57	0.52
GA, $N_g = 20$	0.58	0.28	0.46	0.26	0.39	0.36	0.31	0.44	0.27	0.23	0.69	0.49	0.27	0.23	0.50	0.30

is run in recursive and batch modes. With its batch mode and the GA-based method, a window of 20 frames is used.

Table 2.5 shows the root mean square error (RMSE) of gimbal azimuth and elevation bias estimates at the end of 100 frames while Fig. 2.4 shows the average RMSE over the bias estimates of two platforms in test sets 1 and 2. It can be seen from Table 2.5 that the accuracies of the geo-registration methods proposed in this paper are similar to that of the GA-based method with 20 generations after 100 frames. The GA-based method with 5 generations is less accurate than the others. Compared with the methods proposed in this paper, the GA-based method has a faster convergence rate. Since calibration needs to be done only once (i.e., at the beginning of a mission), the convergence rate is not an issue.

The accuracies of debiased target geo-locations by using the obtained bias estimates are then evaluated. Table 2.6 gives the RMSE of debiased target geo-locations



(a) Test set 1 (b) Test set 2

Fig. 2.4: RMSE averaged over the bias estimates of two sensors in test sets 1 and 2

averaged over frames, which is taken as a measure of improvement in target geo-location that can be achieved by geo-registration methods. The RMSEs of converted target geo-locations given the ture gimbal azimuth and elevation of the sensors are also provided as a lower bound on geolocation accuracy. Compared with the RMSEs of original (without debiasing) target geo-locations, all geo-registration methods improve their accuracies significantly. The target geo-locations debiased by the GA-based method with 20 generations have slightly better accuracy than those debiased by the other methods. Fig. 2.5 shows the average RMSE of the converted target geo-locations of the first target that are observed by the two sensors in test sets 1 and 2. The average RMSEs for the second target show a similar trend. It can be seen that the average RMSEs for all debiased methods converge close to that of converted target geo-locations given the true gimbal azimuth and the elevation of the sensors at the end of 100 frames. Note that the RMSE of target geo-locations debiased using bias estimates, which include estimation errors, can reach that of converted target geo-locations given the true gimbal azimuth and the elevation of the sensors.

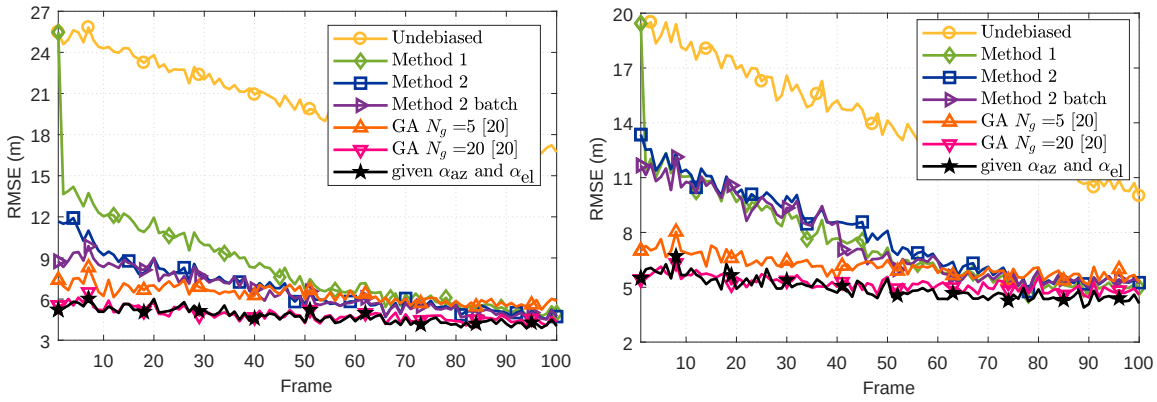
Table 2.6: RMSE of converted target geo-locations (in meters)

Method	Test set 1				Test set 2				Test set 3				Test set 4			
	z_{11}	z_{12}	z_{21}	z_{22}	z_{11}	z_{12}	z_{21}	z_{22}	z_{11}	z_{12}	z_{21}	z_{22}	z_{11}	z_{12}	z_{21}	z_{22}
Undebiased	19.4	22.4	21.4	18.5	11.1	11.1	17.2	16.0	13.5	14.4	19.3	17.2	14.4	14.3	25.2	22.2
Method 1	7.4	8.4	9.1	7.3	7.1	7.8	7.7	6.6	5.9	7.0	7.2	5.9	8.3	8.7	10.5	8.0
Method 2	6.1	6.9	7.5	6.1	7.0	7.6	8.7	7.0	5.7	6.4	7.0	5.8	6.7	7.1	9.8	7.4
Method 2 batch	5.8	6.4	7.1	5.9	6.7	7.3	8.2	6.7	5.7	6.4	6.8	5.7	6.4	6.9	9.6	7.1
GA, $N_g = 5$	5.8	6.3	6.9	5.9	5.7	6.1	6.6	5.8	5.4	6.0	6.3	5.2	6.0	6.4	6.7	5.8
GA, $N_g = 20$	4.4	4.8	5.3	4.8	4.7	5.0	5.6	5.1	4.3	4.8	5.0	4.3	4.9	5.2	5.4	4.9
Bias-free	3.8	4.2	5.7	5.2	4.4	4.8	5.4	5.3	4.1	4.7	6.2	5.2	4.4	4.7	5.6	5.2

This is because the effects of biases in yaw and roll measurements, indistinguishable from those of biases in gimbal azimuth and elevation measurements, respectively, are partially removed together when debiasing.

A more accurate target geo-location can be obtained by filtering the converted geo-locations across frames. The improvement in the final target geo-location estimates after filtering achieved by geo-registration methods is also analyzed by evaluating the RMSE of target geo-location estimates, which is directly generated for the first proposed geo-registration method and can be obtained by feeding undebiased/debiased converted target geo-locations into a Kalman filter for the other methods. The RMSE of geo-location estimates averaged over two targets in test sets 1 and 2 is shown in Fig. 2.6. It shows a trend similar to that of the RMSE of converted target geo-location. The accuracies of geo-location estimates obtained using the geo-registration methods proposed in this paper fall in between those obtained using the GA-based method with 5 and 20 generations.

Finally, the computational costs of the geo-registration methods are evaluated on a 2.5 GHz intel i7-4710 HQ laptop with 16 GB RAM. The computation times per run averaged over 100 Monte Carlo runs are given in Table 2.7, which shows that



(a) Test set 1 (b) Test set 2

Fig. 2.5: RMSE values averaged over the first target geo-locations that are observed by the two sensors in test sets 1 and 2

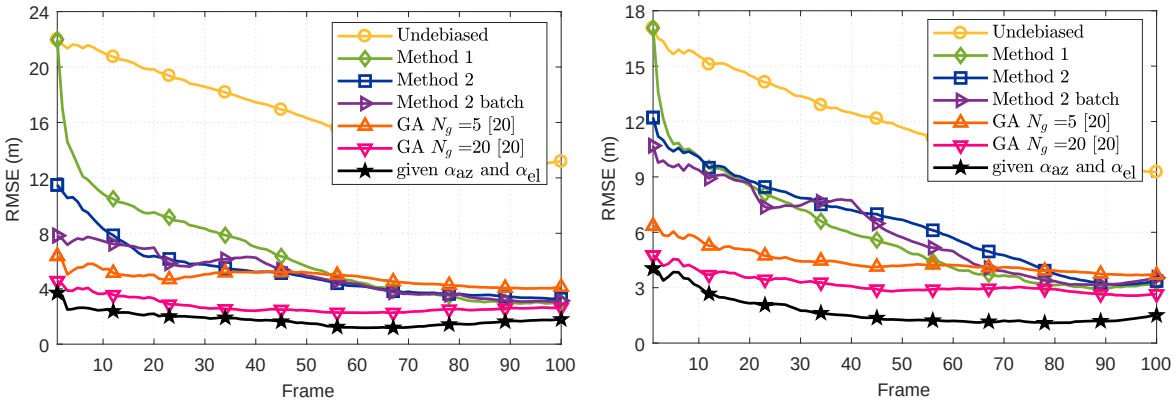
Table 2.7: Computation times averaged over 100 Monte Carlo runs (in seconds)

	Method 1	Method 2	Method 2 batch	GA, $N_g = 5$	GA, $N_g = 20$
Computation time	0.16	0.13	0.12	12.31	46.79

the computational demands of the two methods proposed in this paper are almost the same but are significantly less than those of the GA-based methods. This makes the proposed algorithms real-time feasible even at high frame rates. In contrast, the GA-based method may not be real-time ready at high frame rates.

2.7 Conclusions

The problem of airborne video geo-registration for tracking ground targets was studied in this paper. The effect of sensor biases on converted target geo-location was analyzed and modeled. It was then decoupled from true target geo-location and approximately expressed by a term, which is linear with sensor biases, by assuming



(a) Test set 1 (b) Test set 2

Fig. 2.6: RMSE values of geo-location estimate averaged over two targets in test sets 1 and 2

that the biases are small. Based on the decoupled result, two methods were proposed to estimate the biases by (a) estimating the biases jointly with target geo-locations, (b) estimating the biases separately by using the information from the common targets that are observed by multiple airborne sensors simultaneously. Numerical results demonstrated that the proposed methods achieve the same performance as the previous optimization-based method, in which the biases are modeled by a highly non-linear term and estimated using the genetic algorithm. The proposed methods are significantly simpler and substantially faster in terms of CPU time, making them real-time feasible even on airborne platforms with moderate computational resources and at high frame rates.

Bibliography

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.* Wiley, NY, 2001.

[2] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo–location using a fixed–wing miniature air vehicle," *J. Intell. Rob. Syst.*, vol. 47, no. 4, pp. 361–382, Nov. 2006.

[3] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Boston, MA: Artech House, 1999.

[4] Y. Chen, J. Wang, M. Xu, X. He, and H. Lu, "A unified model sharing framework for moving object detection," *Signal Process.*, vol. 124, pp. 72–80, Jul. 2016.

[5] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, and P. Burt, "A system for video surveillance and monitoring," *VSAM final report*, pp. 1–68, May 2000.

[6] L. Davis, *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York, 1991.

[7] H. Deng, X. Sun, M. Liu, C. Ye, and X. Zhou, "Infrared small-target detection

using multiscale gray difference weighted image entropy," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 1, pp. 60–72, Feb. 2016.

[8] V. N. Dobrokhodov, I. Kaminer, K. D. Jones, and R. Ghabcheloo, "Vision–based tracking and motion estimation for moving targets using small UAVs," in *American Control Conference*, Minneapolis, USA, Jun. 2006, pp. 1428–1433.

[9] J. R. Guerci, *Space-time Adaptive Processing for Radar.* Boston, MA: Artech House, 2014.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Fluids Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[11] T. Kirubarajan, Y. Bar-Shalom, and K. R. Pattipati, "Ground target tracking with variable structure IMM estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 26–46, Jan. 2000.

[12] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna, "Registration of highly–oblique and zoomed in aerial video to reference imagery," in *International Conference on Pattern Recognition*, vol. 4, Barcelona, Spain, Sep. 2000, pp. 303–307.

[13] W. Li, H. Leung, and Y. Zhou, "Space-time registration of radar and ESM using unscented Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 824–836, Jul. 2004.

[14] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Jul. 2003.

[15] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Science & Business Media, Berlin, Germany, 2012.

[16] T. Mckay and H. Hirsch, "A fast, accurate, cross–modality image geo-registration and target/object detection algorithm," in *Proceedings of SPIE, Defense, Security, and Sensing*, vol. 8747, May 2013, pp. 1–8.

[17] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*, Ganga-Jamuna Press, Lincoln, MA, USA, 2006.

[18] P. Qiu and C. Xing, "Feature based image registration using non-degenerate pixels," *Signal Process.*, vol. 93, no. 4, pp. 706–720, Apr. 2013.

[19] S. Ruano, G. Gallego, C. Cuevas, and N. Garca, "Aerial video georegistration using terrain models from dense and coherent stereo matching," in *Proceedings of SPIE, Geospatial InfoFusion and Video Analytics IV*, vol. 9089, Jun. 2014, pp. 1–10.

[20] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 198–219, Sep. 2013.

[21] D. Song, R. Tharmarasa, T. Kirubarajan, and X. Fernando, "Multi-vehicle tracking using microscopic traffic models," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–13, Mar. 2018.

[22] ——, "Multi-vehicle tracking with road maps and car-following models," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1375–1386, May 2018.

[23] S. Srinivasan, H. Latchman, J. Shea, T. Wong, and J. McNair, "Airborne traffic surveillance systems: Video surveillance of highway traffic," in *Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks*, New York, USA, 2004, pp. 131–135.

[24] E. Taghavi, R. Tharmarasa, T. Kirubarajan, Y. Bar-Shalom, and M. McDonald, "A practical bias estimation algorithm for multisensor-multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 1, pp. 2–19, Feb. 2016.

[25] E. Taghavi, D. Song, R. Tharmarasa, T. Kirubarajan, M. McDonald, and B. Balaji, "Geo-registration and geo-location using two airborne video sensors," *IEEE Trans. Aerosp. Electron. Syst.*, To appear.

[26] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision.* Prentice Hall Englewood Cliffs, N.J., USA, 1998.

[27] M. Ulmke, O. Erdinc, and P. Willett, "GMTI tracking via the Gaussian mixture cardinalized probability hypothesis density filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 1821–1833, Oct. 2010.

[28] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[29] W. Wu, J. Jiang, W. Liu, X. Feng, L. Gao, and X. Qin, "Augmented state GM-PHD filter with registration errors for multi-target tracking by Doppler radars," *Signal Process.*, vol. 120, pp. 117–128, Mar. 2016.

[30] M. Yu, C. Liu, B. Li, and W. H. Chen, "An enhanced particle filtering method

for GMTI radar tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 3, pp. 1408–1420, Jun. 2016.

[31] L. Zheng and K. Tian, "Detection of small objects in sidescan sonar images based on POHMT and Tsallis entropy," *Signal Process.*, vol. 142, pp. 168–177, Jan. 2018.

The following chapter is a reproduction of an Institute of Electrical and Electronics Engineers (IEEE) copyrighted, published paper:

Dan Song, Ratnasingham Tharmarasa, Thiagalingam Kirubarajan and Xavier N. Fernando
Multi-Vehicle Tracking With Road Maps and Car-Following Models, *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1375–1386, May 2018. (doi: 10.1109/TITS.2017.2723575)

Chapter 3

Multiple Vehicle Tracking on Single-lane Roads

3.1 Abstract

Multi-vehicle tracking is crucial in many applications, such as traffic surveillance, intelligent transportation systems and advanced driver assistance systems. Most conventional multitarget tracking algorithms are not ideal for multi-vehicle tracking since they assume that targets move independently of one another. However, due to traffic volume and limited lane resources, vehicles have to interact with their neighbors, resulting in highly dependent motions. To address this limitation, this paper proposes a novel multi-vehicle tracking algorithm for the single-lane case that considers motion dependence across vehicles by integrating the car-following model (CFM) into the tracking process with on-road constraints. A new CFM-based motion model that describes the dependent motion of vehicles in the single lane case is proposed and the notion of car-following clusters is defined. In order to exploit all available information

in sensor measurements, the proposed algorithm updates the state estimates of car-following clusters by utilizing a stacked-update strategy. Furthermore, the variable structure interacting multiple model (VS-IMM) estimator is modified and integrated into the proposed algorithm to handle maneuvers that may violate the CFM. Simulation results demonstrate the superiority of the proposed multi-vehicle tracking algorithm over other state-of-the-art multi-vehicle tracking algorithms.

3.2 Introduction

Vehicle tracking is important in many different applications in ground surveillance [46], traffic surveillance [10], intelligent transportation systems (ITS) [11] and advanced driver assistance systems (ADAS) [9]. The goal of the work presented in this paper is tracking multiple vehicles in applications such as traffic surveillance and intelligent traffic monitoring through infrastructure-mounted sensors to support trajectory planning [11] and traffic management. Due to the characteristics that distinguish it from the standard multitarget tracking (MTT) problem [14], significant research in this area has been done in recent years [46, 10, 11, 13, 23, 31, 30, 39, 42, 16, 24, 12, 28, 35, 38, 15].

One of the distinguishing features of vehicle tracking is that as vehicles move along roads, their motions are subjected to various constraints imposed by those roads. The road map can then be utilized as prior information, significantly improving the tracking performance while reducing target uncertainty and false alarms [13, 24]. Making use of the particle filter, the road constraints can be used to align the state probability densities along roads by discarding off-road particles [23, 31, 13]. However, most of the works on particle-filter based approaches consider only

single-target problems due to high computational cost, especially with closely-spaced objects, which is a common challenge in vehicle tracking scenarios. The notion of directional process noise is introduced in [30, 39] to impose road constraints on state estimation in a "soft" manner and thus the resulting state estimate is not guaranteed to be located on the road. To overcome this deficiency, the vehicle tracking with road constraints is identified as the state estimation with equality constraints in [42], by modeling roads as sequences of linear segments, like most geographical information systems (GIS) do. Also, it was demonstrated that the solution to this problem is the projection of Kalman filter estimates onto the roads. Another way is to consider the road constraints as pseudo-measurements from a virtual sensor fed into the standard unconstrained filter along with other measurements [16]. However, this method may lead to numerical problems due to the use of noise-free pseudo-measurements. Unlike the above methods that describe the vehicle motions in the ground coordinates with additional ways to constrain them on-road, it is more convenient to model them in road coordinates [46, 24, 12, 28] so that on-road constraints are directly integrated: In [46], a vehicle is tracked in road coordinates using one-dimensional representation of vehicle kinematics. The geometric map error is considered during tracking in [24]. This method is extended in [12] with consideration for the lateral motion of vehicles as another dimension in road coordinates. The curvilinear version of two-dimensional road model is proposed in [28].

Due to traffic volume and limited lane resources [14], vehicles usually interact with neighboring vehicles while moving along roads. This leads to highly dependent motion across vehicles, a particular difficulty in vehicle tracking. The motion dependence can be regarded as another piece of prior information, similar to the road

map, to further improve tracking performance. The motion dependence in pedestrian motion is considered in [47, 17] with the social force model (SFM) being proposed in [25] to model the interactions between pedestrians. The modified SFM is presented to model vehicle dynamics and used in vehicle tracking in [15]. However, the SFM is an empirical model in which the interactions are modeled as virtual forces without any direct physical meaning and thus the SFM cannot be used to describe vehicle dynamics accurately. In contrast, the car-following model (CFM), describing the vehicle dynamics as the process of one vehicle following another in traffic flow, has been studied for nearly half a century [34]. The most well-known CFM is the Gazis-Herman-Rothery (GHR) model [19], which describe the motion dependence by modeling the acceleration of the following vehicle as a function of its motion relative to the leading vehicle. Other models similar to the GHR model include the linear Helly model [26] and the Intelligent drive model (IDM) [45]. The Gipps model [20], unlike the aforementioned models that define the acceleration of the following vehicle, aim to define a safe distance for avoiding collisions. The Gipps model is widely used in many traffic simulators because of its ability to describe the propagation of disturbances. Other CFMs include the Action Point (AP) model, Cellular Automation Model and the fuzzy logic-based model [34].

While the use of road information has been explored in detail in the literature, the information about dependent motion across vehicles has not yet been well-addressed. To the best of our knowledge, the motion dependence between vehicles is used for multi-vehicle tracking only in [38], in which the GHR model is considered for evaluating candidate track probabilities within the multiple hypothesis tracking (MHT) framework, and in [15], in which the modified SFM, though inaccurate as mentioned

above, is used as the dynamic model for targets.

Thus the motivation for this paper is to propose a new algorithm for multi-vehicle tracking in the single-lane case with the integration of both road maps and motion dependence information. In the proposed tracking algorithm, the car-following cluster is defined as the basic unit, where complete motion dependence is present and is described by CFM in road coordinates. A stacked-vector strategy, commonly used for correlated measurements and target motions [37, 18], is applied to each car-following cluster to update estimates. Given the possibility that some vehicles may not follow their leading vehicle to accelerate when they reach their desired speed, the variable structure interacting multiple model (VS-IMM), an adaptive version of the fixed IMM estimator [3, 32], in which the active model set is adjusted according to current conditions, is used in the proposed tracking algorithm. The corresponding Posterior Cramer-Rao Lower Bound (PCRLB) is also derived in this paper to quantify the performance of the proposed tracking algorithm.

The remainder of the paper is organized as follows: Section 3.3 introduces the basic models necessary for the multi-vehicle tracking problem. In Section 3.4, the CFM-based multi-vehicle tracking algorithm, with the integrated road map, is proposed. The PCRLB is derived in Section 3.5. In Section 3.6, numerical results are provided to demonstrate the advantages of the proposed CFM-based multi-vehicle tracking algorithm over other state-of-the-art algorithms [46, 15, 12]. Concluding remarks are given in Section 3.7.

3.3 Background

3.3.1 One-dimensional Road Representation

Most GIS works represent roads as sequences of linear segments [7]. This model has also been used in vehicle tracking [46, 12]. Ignoring altitude information and map mismatch, each segment i is defined by the following three attributes:

1. starting point $\mathbf{s}_i$
2. unit direction vector $\mathbf{d}_i$
3. segment length λ_i.

The relationship among these attributes is expressed as $\mathbf{s}_{i+1} = \mathbf{s}_i + \lambda_i \mathbf{d}_i$. The arc length l can be uniquely defined corresponding to each point on the road as $l = \sum_{j=1}^{k-1} \lambda_j + \hat{\lambda}_k$, where the kth segment is the segment that the point is on and $\hat{\lambda}_k$ is the distance from the point to $\mathbf{s}_k$. Thus, the arc length corresponding to the starting point of each segment is $l_i = \sum_{j=1}^{i-1} \lambda_j$. Once the arc length is given, its corresponding position in the ground coordinate can be obtained by

$$\mathbf{r}_g(l) \triangleq \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{s}_k + (l - l_k)\mathbf{d}_k \tag{3.1}$$

where k is determined by the following inequality:

$$l_k \leq l \leq l_{k+1}. \tag{3.2}$$

3.3.2 Car-Following Model

The linear Helly model [26] is used in this paper to describe the longitudinal dynamics of vehicles that moves along a single lane by following the leading vehicle. It is used not only for its simplicity but also for the practical meaning of each term in the linear Helly model. Other CFMs can also be incorporated into the dependent tracking algorithm proposed in this paper with the framework of an extended Kalman filter (EKF) [3].

The Helly model gives the formulation of acceleration of the following vehicle a_F as [26]

$$a_F(t) = C_1\Delta v(t-T) + C_2\Delta x(t-T) + C_3 v_F(t-T) + c \tag{3.3}$$

where v_F is the speed of the following vehicle, Δv and Δx are the relative distance and speed, respectively between the leading and following vehicle. In the above, T is the reaction time for drivers, while C_1, C_2, C_3, c and T are the parameters that are determined by experiments. The values given by Helly [26] are $C_1 = 0.5$, $C_2 = 0.125$, $C_3 = -0.125$, $c = -2.5$ and $T = 0.5\,\text{s}$. Calibrations of these parameters have been carried out by Rockwell [36], Bekey [6], and Aron [1]. Note that the intent of this paper is to develop a way to integrate motion dependence information into the tracking process, not to determine the optimal solution of these parameters. Therefore, we still use Helly's results and parameter mismatch is not considered in this paper. The parameter c in (3.3) indicates the aggressiveness of driving behavior that determines the distance between the two vehicles when they reach steady state equilibrium. For example, aggressive drivers can be modeled by a low absolute value, which means

they follow the leading vehicle very closely. Considering different driving behaviors, c is assumed normally distributed with mean $\bar{c} = -2.5$ for each single vehicle in this paper.

3.3.3 Vehicle Motion Model in the Road Coordinate

Assuming vehicles only move on roads with single lane, only the longitudinal motion of the vehicle needs to be considered and it can be described in the road coordinate [46]. In that case, the kinematic state of the vehicle at time t_k can be defined by

$$\mathbf{x}(t_k) \triangleq [l(t_k), \dot{l}(t_k)]^{\mathrm{T}} \tag{3.4}$$

where $(\cdot)^{\mathrm{T}}$ denotes the transpose, $l(t_k)$ and $\dot{l}(t_k)$ are the position and speed, respectively, of the vehicle at time t_k and are represented by the arc length defined in Section 3.3.1.

Vehicle motion has two different modes depending on whether or not there is a leading vehicle in front. The effective car-following distance l_e represented by the arc length is introduced to distinguish these two cases: The free driving-mode is used when there are no other targets within the range of effective car-following distance, while the following driving-mode is used otherwise. For a vehicle in free driving-mode, the evolution of its state can be described by the discrete nearly constant velocity model [3]:

$$\mathbf{x}(t_k) = F(\delta_k)\mathbf{x}(t_{k-1}) + \Gamma(\delta_k)v(t_{k-1}) \tag{3.5}$$

where $\delta_k = t_k - t_{k-1}$ is the sampling interval and $v(t_{k-1})$ is the white Gaussian process noise sequence with standard deviation $\sigma_v(\delta_k)$ considered as a noisy acceleration component to model disturbances of driving behaviors. The state transition matrix F and noise gain Γ are

$$F(\delta_k) = \begin{bmatrix} 1 & \delta_k \\ 0 & 1 \end{bmatrix}, \quad \Gamma(\delta_k) = \begin{bmatrix} \delta_k^2/2 \\ \delta_k \end{bmatrix}. \tag{3.6}$$

The dynamics of a vehicle in following driving-mode can be similarly described by

$$\mathbf{x}(t_k) = F(\delta_k)\mathbf{x}(t_{k-1}) + \Gamma(\delta_k)\left(a(t_{k-1}) + v(t_{k-1})\right) \tag{3.7}$$

where $a(t_{k-1})$ is an additional acceleration term determined by the CFM. Based on (3.3), we have

$$a(t_{k-1}) = F_l\mathbf{x}_l(t_{k-1}) + F_f\mathbf{x}(t_{k-1}) + c \tag{3.8}$$

where $F_l = [C_2 \;\; C_1]$, $F_f = [-C_2 \;\; -C_1 + C_3]$ and $\mathbf{x}_l(t_{k-1})$ denotes the state of the leading vehicle.

We first define a notion of a car-following pair as two adjacent vehicles such that the distance between them is smaller than l_e. The car-following cluster is defined as a sequence of car-following pairs such that adjacent pairs share a common vehicle. For example, three vehicles driving together form a car-following cluster consisting of two car-following pairs that share the second vehicle. Because of shared vehicles, the motions of all vehicles in the same car-following cluster are dependent on each other. Thus, the states of all vehicles in the same car-following cluster are stacked

and propagated together.

Assume that there are $N(t_k)$ vehicles in one car-following cluster at time t_k and let $\mathbf{X}(t_k)$ denote the stacked vehicle state obtained by stacking the state of these vehicles $\mathbf{x}_n(t_k),\ n = 1, ..., N(t_k)$ from the front. Note that the first vehicle in the car-following cluster is in free driving-mode, while the other vehicles evolve according to the car-following mode. Based on (3.5) and (3.7), the evolution of $\mathbf{X}(t_k)$ can be expressed as

$$\mathbf{X}(t_k) = \mathbf{F}_s(\delta_k)\mathbf{X}(t_{k-1}) + \mathbf{\Gamma}(\delta_k)\left(\mathbf{a}(t_{k-1}) + \mathbf{v}(t_{k-1})\right) \tag{3.9}$$

where $\mathbf{F}_s(\delta_k)$ and $\mathbf{\Gamma}(\delta_k)$ are the stacked form of $F(\delta_k)$ and $\Gamma(\delta_k)$ in (3.5) and (3.7), respectively. That is,

$$\mathbf{F}_s(\delta_k) = \mathrm{diag}(\underbrace{F(\delta_k), \ldots, F(\delta_k)}_{N(t_k)}) \tag{3.10}$$

$$\mathbf{\Gamma}(\delta_k) = \mathrm{diag}(\underbrace{\Gamma(\delta_k), \ldots, \Gamma(\delta_k)}_{N(t_k)}) \tag{3.11}$$

with $\mathrm{diag}(\cdot)$ denoting (block) diagonal matrix construction operator. Also, $\mathbf{a}(t_{k-1}) = \begin{bmatrix}0 & a_2(t_{k-1}) & \ldots & a_{N(t_k)}(t_{k-1})\end{bmatrix}^{\mathrm{T}}$ and $\mathbf{v}(t_{k-1}) = \begin{bmatrix}v_1(t_{k-1}) & \ldots & v_{N(t_k)}(t_{k-1})\end{bmatrix}^{\mathrm{T}}$, with $a_n(t_{k-1})$ and $v_n(t_{k-1})$ representing the additional acceleration term and process noise for the nth vehicle, respectively. Compared to (3.8), $\mathbf{a}(t_{k-1})$ can also be written as

$$\mathbf{a}(t_{k-1}) = \mathbf{A}\mathbf{X}(t_{k-1}) + \mathbf{c} \tag{3.12}$$

where $\mathbf{c} = [0 \;\; c_2 \;\; \dots c_{N(t_k)}]^{\mathrm{T}}$ and

$$\mathbf{A} = \begin{bmatrix} 0 & & & & \\ F_l & F_f & & & \\ & F_l & F_f & & \\ & & \ddots & \ddots & \\ & & & F_l & F_f \end{bmatrix}. \tag{3.13}$$

Then, (3.9) can be further simplified as

$$\mathbf{X}(t_k) = \mathbf{F}(\delta_k)\mathbf{X}_r(t_{k-1}) + \mathbf{\Gamma}(\delta_k)\left(\mathbf{c} + \mathbf{v}(t_{k-1})\right) \tag{3.14}$$

$$\mathbf{F}(\delta_k) = \mathbf{F}_s(\delta_k) + \mathbf{\Gamma}(\delta_k)\mathbf{A}. \tag{3.15}$$

Note that since (3.7) is equivalent to (3.14) with $N(t_k) = 1$, the vehicle that is not in any car-following pairs can also be treated as a car-following cluster.

3.3.4 Measurement Model

Many different sensors are available for localization and tracking. In uncooperative applications, radar and camera are the primary sensors, while the Global Positioning System (GPS) and Inertial Navigation System (INS) are generally used in cooperative applications. However, most of these sensors report measurements in their local coordinates. Without loss of generality and for simplicity, we assume that the sensor provides only the position data [14], which has been converted to the ground coordinate before being entered into the tracking system. The target position in the ground coordinate at time t_k, denoted by $\mathbf{p}(t_k)$, can be converted from the road coordinate

using (3.1) as

$$\mathbf{p}(t_k) = \mathbf{s}_{i(t_k)} + (l(t_k) - l_{i(t_k)})\mathbf{d}_{i(t_k)} \tag{3.16}$$

where $i(t_k)$ is the segment the target is on, which is determined by (3.2) replacing l with $l(t_k)$. Rewrite (3.16) as

$$\mathbf{p}(t_k) = T(t_k)\mathbf{x}(t_k) - \mathbf{g}(t_k) \tag{3.17}$$

where

$$T(t_k) = \begin{bmatrix} \mathbf{d}_{i(t_k)} & \mathbf{0} \end{bmatrix} \tag{3.18}$$

$$\mathbf{g}(t_k) = \mathbf{s}_{i(t_k)} - l_{i(t_k)}\mathbf{d}_{i(t_k)}. \tag{3.19}$$

Due to the sensor's capability being limited by noise, multipath and clutter in harsh environments, the received measurements are inaccurate and there are target misdetections and false alarms [4, 5]. Thus, the target-originated measurement received at time t_k, $\mathbf{z}(t_k)$, is modeled as

$$\begin{aligned} \mathbf{z}(t_k) &= \mathbf{p}(t_k) + \mathbf{w}(t_k) \\ &= T(t_k)\mathbf{x}(t_k) - \mathbf{g}(t_k) + \mathbf{w}(t_k) \end{aligned} \tag{3.20}$$

where $\mathbf{w}(t_k)$ is the measurement noise modeled as the white Gaussian noise with covariance $R(t_k)$. The target-originated measurements are detected with probability P_D and false alarms are uniformly distributed in the field of view of the sensor

with their cardinality being Poisson-distributed with parameter λ [8]. The set of all measurements reported at time t_k is denoted as

$$Z(t_k) \triangleq \mathbf{z}_m(t_k),\ m = 1, \ldots, M(t_k) \tag{3.21}$$

and the cumulative set of measurements available up to time t_k is denoted as

$$Z^k \triangleq Z(t_i),\ i = 1, \ldots, k. \tag{3.22}$$

3.4 CFM-based Multi-vehicle Tracking in Road Coordinate

Based on the models described in Section 3.3, the CFM-based multi-vehicle tracking algorithm with integrated road-map is proposed and the block diagram is shown in Fig. 3.1.

3.4.1 CFM-based Kalman Filter

The CFM-based Kalman filter (CFM-KF), incorporating the CFM under the standard Kalman filter [29] framework, is proposed to update the stacked vehicle state of each car-following cluster. We use time step k to represent the real time t_k for simplicity. Since the parameter c in (3.3) is unknown from the tracker's point of view, it needs to be estimated along with target kinematic state. Thus, the augmented state at time step k is given by $\bar{\mathbf{x}}(k) \triangleq [\mathbf{x}(k)^{\mathrm{T}}\ c]^{\mathrm{T}}$ and its corresponding estimate at time step k is denoted as $\hat{\bar{\mathbf{x}}}(k|k)$. In the following, the track state refers to $\hat{\bar{\mathbf{x}}}(k|k)$. The

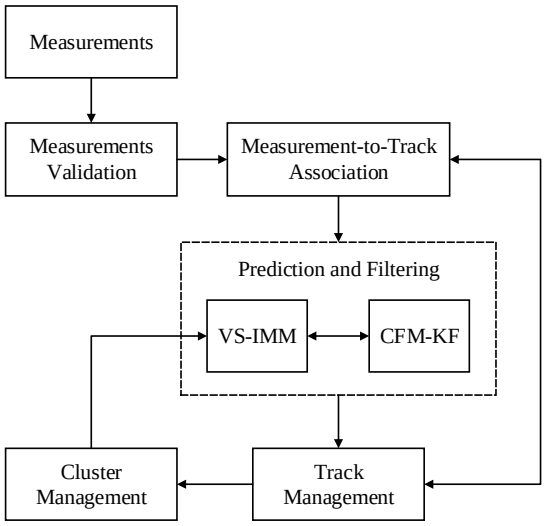


Fig. 3.1: Block diagram of proposed dependent-motion multi-vehicle tracking algorithm.

steps of CFM-KF are as follows:

State prediction

Assume that there are $N(k-1)$ active tracks in one car-following cluster at the end of processing scan $k-1$ given Z^k. Based on the stacked state equation shown in (3.14), the predicted stacked state at time step $k-1$, $\hat{\bar{\mathbf{X}}}(k|k-1)$, is given by

$$\hat{\bar{\mathbf{X}}}(k|k-1) = \bar{\mathbf{F}}(\delta_k)\hat{\bar{\mathbf{X}}}(k-1|k-1) \tag{3.23}$$

where $\hat{\bar{\mathbf{X}}}(k-1|k-1)$ is the stacked state estimate at the end of processing scan $k-1$. Due to the augmentation of track state, $\bar{\mathbf{F}}(\delta_k)$ is obtained by (3.15) but $F(\delta_k)$, $\Gamma(\delta_k)$,

F_l, and F_f also need to be augmented as follows:

$$\bar{F}(\delta_k) = \begin{bmatrix} F(\delta_k) & \\ & 1 \end{bmatrix}, \ \bar{\Gamma}(\delta_k) = \begin{bmatrix} \Gamma(\delta_k) \\ 0 \end{bmatrix}$$

$$\bar{F}_l = \begin{bmatrix} F_l & 0 \end{bmatrix}, \ \bar{F}_f = \begin{bmatrix} F_f & 0 \end{bmatrix}. \tag{3.24}$$

The associated predicted stacked state covariance $\mathbf{P}(k|k-1)$ is

$$\mathbf{P}(k|k-1) = \bar{\mathbf{F}}(\delta_k)\mathbf{P}(k-1|k-1)\bar{\mathbf{F}}(\delta_k)^{\mathrm{T}} + \bar{\mathbf{\Gamma}}(\delta_k)\mathbf{Q}(\delta_k)\bar{\mathbf{\Gamma}}(\delta_k)^{\mathrm{T}} \tag{3.25}$$

where $\mathbf{P}(k-1|k-1)$ is the covariance matrix associated with $\hat{\bar{\mathbf{X}}}(k-1|k-1)$, $\bar{\mathbf{\Gamma}}(\delta_k)$ is obtained by (3.11) with the augmented $\bar{\Gamma}(\delta_k)$ and $\mathbf{Q}(\delta_k) = \mathrm{diag}(\underbrace{\sigma_v^2(\delta_k), \ldots, \sigma_v^2(\delta_k)}_{N(k-1)})$ by assuming that the process noise for each track has the same standard deviation $\sigma_v(\delta_k)$. It should be noted that since the update interval of prediction δ_k is limited at 0.5s (3.3), it has to make multiple predictions when the interval between two time steps is $u\delta_k$, with u being a positive integer. Otherwise, interpolation is used to make the prediction.

Measurement prediction

Based on the measurement equation (3.20), the predicted stacked measurement, $\hat{\mathbf{Z}}(k|k-1)$, is given by

$$\hat{\mathbf{Z}}(k|k-1) = \left[\hat{\mathbf{z}}_1(k|k-1)^{\mathrm{T}}, \ldots, \hat{\mathbf{z}}_{N(k-1)}(k|k-1)^{\mathrm{T}}\right]^{\mathrm{T}}$$

$$= \bar{\mathbf{T}}(k)\hat{\bar{\mathbf{X}}}(k|k-1) - \mathbf{G}(k) \tag{3.26}$$

where $\bar{\mathbf{T}}(k) = \text{diag}\left(\bar{T}_1(k), \ldots, \bar{T}_{N(k-1)}(k)\right)$ and $\mathbf{G}(k) = [\mathbf{g}_1(k)^{\mathrm{T}}, \ldots, \mathbf{g}_{N(k-1)}(k)^{\mathrm{T}}]^{\mathrm{T}}$ with $\bar{T}_n(k) = [T_n(k)\ \mathbf{0}]$ being the augmented $T(k)$ defined in (3.18) and $\mathbf{g}_n(k)$ being defined in (3.19) for the nth track in the car-following cluster. The associated innovation covariance $\mathbf{S}(k)$ is given by

$$\mathbf{S}(k) = \bar{\mathbf{T}}(k)\mathbf{P}(k|k-1)\bar{\mathbf{T}}(k)^{\mathrm{T}} + \mathbf{R}(k) \tag{3.27}$$

where $\mathbf{R}(k) = \text{diag}\left(R_1(k), \ldots, R_{N(k-1)}(k)\right)$ is the stacked measurement covariance and $R_n(k)$ is the covariance matrix of measurement noise for the nth track in the car-following cluster.

State update

The states of all tracks in the same car-following cluster are dependent on each other and on all measurements associated with that cluster. That is, each measurement associated with a cluster contains information about all the tracks in that car-following cluster. Thus, updating each track on the basis of all measurements associated with the tracks in the same car-following cluster makes the best use of available information. The stacked-update strategy is used to update all the tracks in the same car-following cluster.

Based on the measurement-to-track association result obtained in Section 3.4.4, all measurements associated with the tracks in the car-following cluster are stacked together and the stacked measurement is denoted as $\mathbf{Z}(k)$. Then, using the standard

Kalman filter [3], the stacked state estimate is updated by

$$\hat{\tilde{\mathbf{X}}}(k|k) = \hat{\tilde{\mathbf{X}}}(k|k-1) + \mathbf{W}(k)\boldsymbol{\nu}(k) \tag{3.28}$$

where $\mathbf{W}(k)$ is the filter gain given by

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\tilde{\bar{\mathbf{T}}}(k)^{\mathrm{T}}\tilde{\mathbf{S}}(k)^{-1} \tag{3.29}$$

and $\boldsymbol{\nu}(k)$ is the stacked innovation evaluated as

$$\boldsymbol{\nu}(k) = \mathbf{Z}(k) - \tilde{\hat{\mathbf{Z}}}(k|k-1). \tag{3.30}$$

Note that since certain tracks may be associated with no real measurement, $\tilde{\hat{\mathbf{Z}}}(k|k-1)$, $\tilde{\bar{\mathbf{T}}}(k)$, and $\tilde{\mathbf{S}}(k)$ are the modified versions of $\hat{\mathbf{Z}}(k|k-1)$ and $\bar{\mathbf{T}}(k)$ in (3.26) and $\mathbf{S}(k)$ in (3.27) obtained by removing all rows and columns related to the tracks that are associated with no real measurement at time step k, respectively. The covariance matrix associated with $\hat{\mathbf{X}}_r^i(k|k)$ is evaluated by

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\tilde{\mathbf{S}}(k)\mathbf{W}(k)^{\mathrm{T}}. \tag{3.31}$$

3.4.2 VS-IMM Estimator

When the leading vehicle accelerates, the following vehicle may or may not obey the CFM to accelerate (depending on whether the following vehicle reaches its desired speed). Note that this is not the case when the leading vehicle decelerates since the safety distance has to be kept. In order to cope with this possible violation of the

CFM, the VS-IMM is introduced and modified to be compatible with the car-following cluster.

Variable filter module selection

Since the violation of the CFM may occur in any car-following pair of a car-following cluster, besides the two basic motion models (i.e., non-maneuvering and maneuvering modes without the violation of the CFM), there are $N(k)-1$ possible motion models where a certain number of vehicles in the front accelerate but other vehicles in the same car-following cluster still maintain a constant speed for a car-following cluster consisting of $N(k)$ vehicles at time step k. Therefore, $N(k)+1$ filter modules are set with one-to-one correspondence with these possible motion models. Denote the jth filter module by M_j and let M_1 and $M_{N(k)+1}$ correspond to the two basic motion models without or with maneuver, respectively; let $M_{n+1}, n = 1, ..., N(k)-1$ correspond to the motion model in which the first n vehicles in the car-following cluster accelerate while the remaining vehicles keep a constant speed.

The basic filter for multi-vehicle tracking, CFM-KF in Section 3.4.1, can be directly used for M_1 and $M_{N(k)+1}$, but a higher level of process noise is used in $M_{N(k)+1}$ to handle maneuvers, as in [30]. The proposed filter can be used with other motion models, but $\bar{\mathbf{F}}(\delta_k)$ in (3.23) and (3.25) and $\mathbf{Q}(\delta_k)$ in (3.25) need to be replaced in M_{n+1} by $\bar{\mathbf{F}}^{n+1}(\delta_k) = \text{diag}(\bar{\mathbf{F}}_n(\delta_k), \bar{\mathbf{F}}_{N(k)-n}(\delta_k))$ and $\mathbf{Q}^{n+1}(\delta_k) = \text{diag}(\underbrace{\sigma^2_{v_h}, \ldots, \sigma^2_{v_h}}_{n}, \underbrace{\sigma^2_{v_l}, \ldots, \sigma^2_{v_l}}_{N(k)-n})$ where $\bar{\mathbf{F}}_n(\delta_k)$ and $\bar{\mathbf{F}}_{N(k)-n}(\delta_k)$ are $\bar{\mathbf{F}}(\delta_k)$ corresponding to the car-following cluster that consists of n tracks and $N(k)-n$ tracks, respectively, while $\sigma_{v_h}(\delta_k)$ is the standard deviation of the high process noise that models maneuvers and $\sigma_{v_l}(\delta_k)$ is the standard deviation of the low process noise that models non-maneuvers.

VS-IMM estimator update

For more details on the VS-IMM estimator update, the reader is referred to [32, 30]. Note that since the mode set depends on the number of tracks in the car-following cluster, the mode set update step is carried out in the cluster management stage, which is described in detail in Section 3.4.6.

3.4.3 Measurement Validation

All measurements are validated before data association to eliminate false alarms and to reduce the number of candidate assignments. Validation is performed in two stages.

On-road constraint validation

The on-road constraint can be used to remove unlikely measurements from targets if their confidence regions do not intersect with any segment of the road. Based on the road model and the measurement model described in Section 3.3, this validation problem is equivalent to testing whether a sphere intersects with a linear segment. To be specific, the confidence region of measurement $\mathbf{z}_m(k)$ is given by

$$\mathcal{V}_1(k, m, \gamma_1) = \left\{ \mathbf{z} | \left[\mathbf{z} - \mathbf{z}_m(k)\right]^{\mathrm{T}} R(k)^{-1} \left[\mathbf{z} - \mathbf{z}_m(k)\right] \leq \gamma_1 \right\} \tag{3.32}$$

where γ_1 is the threshold for on-road constraint validation. Now, introduce a new vector $\tilde{\mathbf{z}}$ as

$$\tilde{\mathbf{z}} \triangleq R(k)^{-\frac{1}{2}} \left[\mathbf{z} - \mathbf{z}_m(k)\right] \tag{3.33}$$

where $R(k)^{-\frac{1}{2}}$ is the Cholesky factor of $R(k)^{-1}$ [21]. The confidence region becomes a sphere described by $\tilde{\mathbf{z}}$ as

$$\mathcal{V}_1(k, m, \gamma_1) = \left\{\tilde{\mathbf{z}} |\ \tilde{\mathbf{z}}^{\mathrm{T}}\tilde{\mathbf{z}} \leq \gamma_1\right\} \tag{3.34}$$

and the ith road segment defined in (3.2) now becomes

$$\tilde{\mathbf{z}} = \tilde{\mathbf{s}}_i + \tilde{\mathbf{d}}_i l, \quad 0 \leq l \leq l_{i+1} - l_i \tag{3.35}$$

with $\tilde{\mathbf{s}}_i = R(k)^{-\frac{1}{2}}\left[\mathbf{s}_i - \mathbf{z}_m(k)\right]$ and $\tilde{\mathbf{d}}_i = R(k)^{-\frac{1}{2}}\mathbf{d}_i$ when represented by $\tilde{\mathbf{z}}$. After some conversions, $\mathcal{V}_1(k, m, \gamma_1)$ intersects with the ith road segment only when one of the following conditions is satisfied:

1. $|\tilde{\mathbf{s}}_i|^2 \leq \gamma_1$
2. $\left|\tilde{\mathbf{s}}_i + \tilde{\mathbf{d}}_i(l_{i+1} - l_i)\right|^2 \leq \gamma_1$
3. $|\tilde{\mathbf{s}}_i|^2 - (\frac{\tilde{\mathbf{s}}_i \cdot \tilde{\mathbf{d}}_i}{|\tilde{\mathbf{d}}_i|})^2 \leq \gamma_1$ and $-(l_{i+1} - l_i) \leq \frac{\tilde{\mathbf{s}}_i \cdot \tilde{\mathbf{d}}_i}{|\tilde{\mathbf{d}}_i|} \leq 0$

Only those measurements that pass the on-road constraint validation are considered for data association and track initialization.

Gating

A gate for measurement-to-track association is formed for each track based on its predicted measurement. Only those measurements falling within the gate are considered for associating with this track. Specifically, measurement $\mathbf{z}_m(k)$ is valid

for associating with the nth track only if

$$\nu_{m,n}(k)^{\mathrm{T}} S_n(k)^{-1} \nu_{m,n}(k) \leq \gamma_2 \tag{3.36}$$

where $\nu_{m,n}(k) = \mathbf{z}_m(k) - \hat{\mathbf{z}}_n(k)$ and γ_2 is the gate threshold. $\hat{\mathbf{z}}_n(k|k-1)$ and $S_n(k)$, the predicted measurements for the nth track at time step k and its associated innovation covariance, can be respectively found from $\hat{\mathbf{Z}}(k|k-1)$ and $\mathbf{S}(k)$ of the car-following cluster that the nth track belongs to. In the case of VS-IMM, $\hat{\mathbf{z}}_n(k|k-1)$ and $S_n(k)$ in (3.36) are replaced by $\hat{\mathbf{z}}_n^j(k|k-1)$ and $S_n^j(k)$ corresponding to the filter module with the largest $|S_n^j(k)|$, respectively, in order to determine whether a measurement is validated or rejected by all filter modules [30].

3.4.4 Data Association

The data association (i.e., measurement-to-track association) problem need to be solved to deal with the measurement origin uncertainty in MTT. Though the state of tracks in the same car-following cluster are dependent with each other, their corresponding likelihood function for measurement-to-track association are still independent since the measurement noise is assumed to be independent. Therefore, the 2-D assignment algorithm that has been commonly used in MTT [30, 2] for association between the list of measurements and the list of tracks can be used here as well. The details on 2-D assignment algorithm can be found in [8].

3.4.5 Track Management

The status of existing tracks is updated and new tracks are initialized based on the data association results. The score-function based method [8] is applied to confirm tentative tracks and terminate dead ones. Those measurements that passed on-road constraint validation but were not associated with any existing track are initialized as tentative tracks. Note that with the consideration of the tracker's stability, the tentative track is not considered in car-following cluster identification and is updated in only free-driving mode until it is promoted as confirmed.

3.4.6 Cluster Management

Based on the results of track management, the structure of car-following clusters is updated by considering four potential types of changes to car-following clusters, as described below. It should be mentioned that the following derivations use the premise that the ID assigned to each car-following cluster as well as each track in the same car-following cluster follow the descending order of their arc length estimates.

Track insertion

When a tentative track is confirmed during track management, it is inserted into the car-following cluster if its arc length estimate is within the effective range of the car-following cluster, i.e., if the inequality

$$\hat{l}_{N(k-1)}(k|k) - l_e \leq \hat{l}(k|k) \leq \hat{l}_1(k|k) + l_e \tag{3.37}$$

is satisfied, where $\hat{l}(k|k)$, $\hat{l}_1(k|k)$ and $\hat{l}_{N(k-1)}(k|k)$ are the arc length estimate at time step k for the newly confirmed track, the first and the last vehicle in the car-following cluster, respectively. The position the newly confirmed track is inserted into is determined by the arc length estimate for tracks in the descending order. Next, the state estimate for the newly confirmed track is inserted into the stacked state estimate for the car-following cluster and the associated covariance matrix is also inserted with the assumption that the newly confirmed track is independent of all the tracks in the car-following cluster. When VS-IMM is used, not only are the overall estimate, each mode-conditioned state estimate, and the covariance matrix each extended, but a new filter module M_{j+1} is added and the IDs of the original filter modules are adjusted as well with the assumption of the insertion position between the $(j-1)$th and jth tracks. The initial state estimate and associated covariance matrix conditioned on M_{j+1} are assigned identically with those conditioned on M_j and its mode probability is set to zero. If the inequality in (3.37) does not hold for any existing car-following clusters, the newly confirmed track forms a new car-following cluster.

Track deletion

When a confirmed track is deleted during track management, it also needs to be removed from the car-following cluster that it belongs to. Assuming that the jth track in the car-following cluster is deleted, all rows and columns corresponding to the jth track in the state estimate and its associated covariance matrix are deleted. Filter module M_{j+1} is also removed and the IDs of the rest filter modules with VS-IMM are adjusted and the mode probabilities are normalized.

Cluster merging

When a car-following cluster approaches its leading car-following cluster, the two clusters are merged into one. The $(i+1)$th car-following cluster merges with the ith car-following cluster to form the new ith car-following cluster at time step k if

$$\hat{l}^i_{N_i(k-1)}(k|k) - \hat{l}^{i+1}_1(k|k) \leq l_e. \tag{3.38}$$

where $\hat{l}^i_{N_i(k-1)}(k|k)$ and $\hat{l}^{i+1}_1(k|k)$ are the state estimate at time step k for the last and first vehicle of the ith and $(i+1)$th car-following cluster, respectively. Then, the stacked state estimate and the associated covariance matrix for the new ith cluster are obtained by block merging those for two clusters as $\hat{\bar{\mathbf{X}}}^i(k|k) = [\hat{\bar{\mathbf{X}}}^i(k|k)^{\mathrm{T}}, \hat{\bar{\mathbf{X}}}^{i+1}(k|k)^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{P}^i(k|k) = \mathrm{diag}(\mathbf{P}^i(k|k), \mathbf{P}^{i+1}(k|k))$. The stacked state estimate and the associated covariance matrix conditioned on the jth filter module of the new ith cluster are obtained by similarly merging those conditioned on the pth filter module of the ith cluster and those conditioned on the qth filter module of the $(i+1)$th cluster, respectively, if VS-IMM is used. Here, $p = j$ and $q = 1$ when $j = 1, ..., N_i(k-1)+1$, while $p = N_i(k-1)+1$ and $q = j - N_i(k-1)$ when $j = N_i(k-1)+2, ..., N_i(k-1)+N_{i+1}(k-1)+1$. The corresponding mode probability is also merged as

$$\mu^i_j(k) = \mu^i_p(k)\mu^{i+1}_q(k) \tag{3.39}$$

and then normalized.

Cluster splitting

Cluster may split when a vehicle in the car-following cluster does not follow the leading vehicle to accelerate or when a track in the middle of cluster is deleted. The ith car-following cluster splits into two new car-following clusters numbered as i and $i+1$ and consisting of the first $j-1$ tracks and the remaining ones, respectively, when

$$\hat{l}^i_{j-1}(k|k) - \hat{l}^i_j(k|k) > l_e. \tag{3.40}$$

Then, the stacked state estimate for two new clusters are obtained by dividing the original $\hat{\bar{\mathbf{X}}}^i(k|k)$ at the jth track, denoted as $\hat{\bar{\mathbf{X}}}^i(k|k) = \hat{\bar{\mathbf{X}}}^i_{1:j-1}(k|k)$ and $\hat{\bar{\mathbf{X}}}^{i+1}(k|k) = \hat{\bar{\mathbf{X}}}^i_{j:N_i(k-1)}(k|k)$, respectively and their associated covariance matrices can be found in the original $\mathbf{P}^i(k|k)$. If VS-IMM is used, due to the way the filter modules are numbered, the pth module, $p = 1, ..., j$, of the new ith cluster and the qth module, $q = 1, \ldots, N_i(k-1) - j + 2$, of the new $(i+1)$th cluster are determined by the pth and $(p + j - 1)$th modules of the original ith cluster, respectively. Therefore, the stacked state estimate and associated covariance matrix conditioned on each filter module for the two new clusters are assigned based on those conditioned on the corresponding filter modules of the original car-following cluster in the similar way. Their mode probabilities are also assigned the same value as the corresponding filter modules in the original car-following cluster and normalized thereafter.

3.5 Posterior Cramer-Rao Lower Bound for Multi-vehicle Tracking

In this section, the Posterior Cramer-Rao lower bound (PCRLB) is derived for multi-vehicle tracking problem with the integrated road map and the CFM. The CRLB, given by the inverse of the Fisher information matrix, provides a lower bound on the minimum mean square error (MSE) for unbiased static estimators. It has been used as a measure for the best possible accuracy any sensor can yield [41, 40]. The PCRLB then sets a lower bound for unbiased dynamic estimators [48]. Let $\hat{\mathbf{X}}(k)$ denote an unbiased estimate of $\mathbf{X}(k)$ based on Z^k. Then,

$$\mathbf{P}(k) = \mathrm{E}\left\{\left(\hat{\mathbf{X}}(k) - \mathbf{X}(k)\right)\left(\hat{\mathbf{X}}(k) - \mathbf{X}(k)\right)^{\mathrm{T}}\right\} \geq J(k)^{-1} \tag{3.41}$$

where $J(k)$ is the Fisher information matrix for $\mathbf{X}(k)$. A computationally efficient formulation is proposed in [44] to recursively evaluate $J(k)$ as

$$\begin{aligned} J(k+1) &= D_k^{22} - D_k^{21}(J(k) + D_k^{11})^{-1}D_k^{12} + J_Z(k+1) \\ &\triangleq J_X(k+1) + J_Z(k+1) \end{aligned} \tag{3.42}$$

where

$$D_k^{11} = E\left\{-\Delta_{\mathbf{X}(k)}^{\mathbf{X}(k)} \ln p(\mathbf{X}(k+1)|\mathbf{X}(k))\right\} \tag{3.43}$$

$$D_k^{12} = E\left\{-\Delta_{\mathbf{X}(k)}^{\mathbf{X}(k+1)} \ln p(\mathbf{X}(k+1)|\mathbf{X}(k))\right\} \tag{3.44}$$

$$D_k^{21} = (D_k^{12})^{\mathrm{T}} \tag{3.45}$$

$$D_k^{22} = E\left\{-\Delta^{\mathbf{X}(k)}_{\mathbf{X}(k+1)} \ln p(\mathbf{X}(k+1)|\mathbf{X}(k+1))\right\} \tag{3.46}$$

$$J_Z(k+1) = E\left\{-\Delta^{\mathbf{X}(k+1)}_{\mathbf{X}(k+1)} \ln p(\mathbf{Z}(k+1)|\mathbf{X}(k+1))\right\}. \tag{3.47}$$

For multi-vehicle tracking, the state vector $\mathbf{X}(k)$ is created by stacking the state of all vehicles, which is denoted by $\mathbf{X}(k) = [\bar{\mathbf{x}}_1(k), \bar{\mathbf{x}}_2(k), \ldots, \bar{\mathbf{x}}_N(k)]$. Then, the overall state equation can be written as

$$\mathbf{X}(k+1) = \mathbf{F}(k)\mathbf{X}(k) + \mathbf{\Gamma}(k)\mathbf{v}(k) \tag{3.48}$$

where $\mathbf{F}(k)$ and $\mathbf{\Gamma}(k)$ are obtained by placing the blocks of $\bar{\mathbf{F}}(\delta_k)$ and $\bar{\mathbf{\Gamma}}(\delta_k)$ for the existing car-following clusters corresponding to the existing targets in the order of the targets in $\mathbf{X}(k)$, respectively, and $\mathbf{v}(k) = [v_1(k), v_2(k), \ldots, v_N(k)]^{\mathrm{T}}$. For example, without loss of generality, we assume that the state of vehicles are stacked in the descending order of their position represented by the arc length in $\mathbf{X}(k)$ and these vehicles comprise $I(k-1)$ car-following clusters at the end of processing scan $k-1$. Thus $\mathbf{F}(k) = \mathrm{diag}(\bar{\mathbf{F}}^1(\delta_k)), \ldots, \bar{\mathbf{F}}^{I(k-1)}(\delta_k))$ and $\mathbf{\Gamma}(k) = \mathrm{diag}(\bar{\mathbf{\Gamma}}^1(\delta_k)), \ldots, \bar{\mathbf{\Gamma}}^{I(k-1)}(\delta_k))$. Substituting (3.48) into (3.42) – (3.46) and using the matrix inverse lemma [3], $J_X(k+1)$ can be expressed as

$$J_X(k+1) = \left[\mathbf{\Gamma}(k)\mathbf{Q}(k)\mathbf{\Gamma}(k)^{\mathrm{T}} + \mathbf{F}(k)J(k)\mathbf{F}(k)^{\mathrm{T}}\right]^{-1} \tag{3.49}$$

where $\mathbf{Q}(k) = \mathrm{diag}(\sigma_v^2(\delta_k), \ldots, \sigma_v^2(\delta_k))$. The measurement contribution $J_Z(k+1)$ is evaluated following the method proposed in [43].

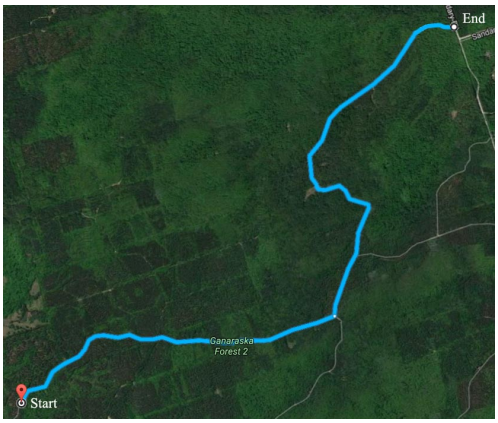


Fig. 3.2: The sample road map in Scenario I and II.

3.6 Simulation Results

In this section, the proposed CFM-based tracking algorithm is compared against other multi-vehicle tracking algorithms in two scenarios simulated on a real road map (near Toronto in Ontario, Canada) with geographic information being obtained from the Google Maps as shown in Fig. 3.2 to illustrate the merits of the proposed algorithm.

3.6.1 Scenario I

In Scenario I, three vehicles following the dynamic model (3.14) without any abnormal maneuvers are simulated to move along the sample road. They start at 400m, 300m, and 250m from the initial point in the road coordinate and move at a nearly constant speed of 50 km/h, 60 km/h, and 60 km/h with corresponding parameter c values of -2.5, -1.5 and -3.5, respectively. The last two vehicles approach the first one at about 25s and all three move as a group with a nearly constant speed of 50 km/h thereafter. Since road intersections are not the focus of this paper, for

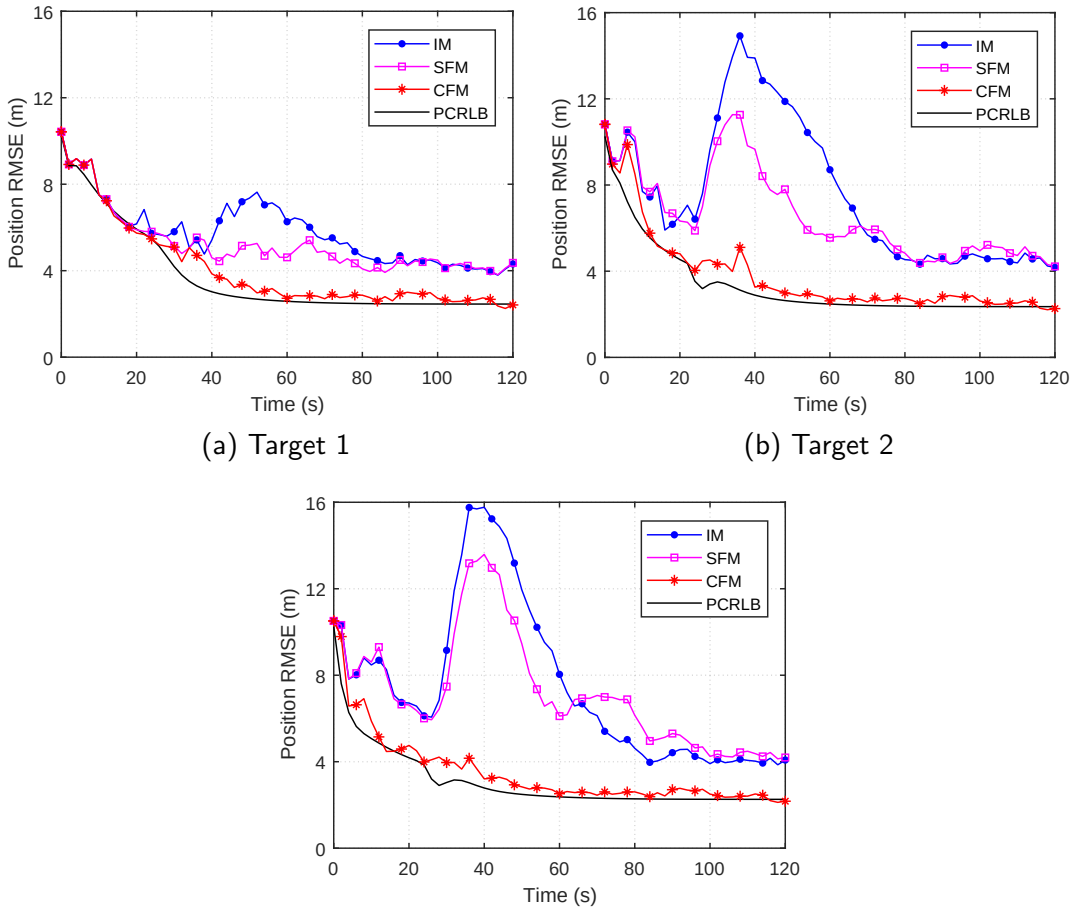
Table 3.1: Statistics of Track Swaps in Scenario I

Algorithm	Number of runs with swap	Statistics of swaps per run		Total number of swaps
		Max	Mean	
IM [46]	33	3	1.30	43
SFM [15]	6	1	1	6
CFM	0	–	–	0

simplicity, the direction uncertainty of intersections is not considered here but the proposed algorithm can also handle the multiroad with intersections by integrating the VS-MM algorithm on each track as it approaches an intersection, as in [46, 35]. Measurements from typical sensors for wide-area sensing, e.g., radar and GPS with vehicle-to-infrastructure communication [9], are assumed reported in the ground coordinate every 2s with the standard deviations of error being $\sigma_x = 10$m and $\sigma_y = 10$m along the x and y directions, respectively. The target detection probability $P_D = 0.95$ and the spatial density of false alarms $\lambda = 5.0 \times 10^{-6}/\text{m}^2$.

The proposed CFM-based tracking algorithm, the standard vehicle tracking algorithm with independent-motion assumption (IM-based algorithm) proposed in [46] and the SFM-based tracking algorithm that integrates the modified SFM proposed in [15] into the framework of IM-based algorithm are compared for the first scenario. The standard deviation of process noise is set as $\sigma_v = 0.1\,\text{m/s}^2$ in all algorithms. The performance of these algorithms is evaluated over 100 Monte Carlo runs. The 2-D assignment algorithm is applied for truth-to-track association to evaluate tracker performance.

Fig. 3.3 shows the Root Mean Square Errors (RMSE) of position estimates and the corresponding PCRLB obtained by the method presented in Section 3.5 for the three targets in road coordinates. It can be seen that the proposed CFM-based yields



(a) Target 1

(b) Target 2

(c) Target 3

Fig. 3.3: Position RMSE and PCRLB corresponding to the three targets in Scenario I.

the best performance and is nearly optimal (close to PCRLB). To be specific, the RMSE of all three targets, especially of target 2 and 3 between 25 s and 50 s, given by the IM-based algorithm are substantially higher. This is improved by integrating SFM but still significant higher than those given by the CFM-based algorithm. This is because the IM-based tracking algorithm, due to considering the motion of each target independently, cannot correctly predict the maneuver in the last two targets

Table 3.2: Average computation times of algorithms in Scenario I

	IM	SFM	CFM
Average computation time (s)	0.65	0.66	0.68

as they decelerate to avoid collision when approaching the first target. In the SFM-based method this maneuver is predicted but only inaccurately, while the proposed CFM-based algorithm predicts this maneuver in a timely and accurate manner. The IM-based algorithm and the SFM-based algorithm yield similar steady state errors because the SFM-based algorithm dose not benefit from the social force model when targets move with a nearly constant speed in steady state. In contrast, the CFM-based algorithm achieves better steady state errors by making use of all measurements that are associated with the tracks in the same car-following cluster to update each track.

Since quantifying track swaps is important when target IDs need to be maintained accurately, the number of track swaps is an important performance metric in multi-target tracking [22]. Table 3.1 gives the statistics of track swaps in 100 Monte Carlo runs in Scenario I. A total of 43 track swaps occur in Scenario I with swaps occurring in 33 out of 100 runs and maximum 3 swaps occurring in one run for the IM-based tracking algorithm. With the integrated SFM, the number of runs with track swap is reduced to 6 with one track swap in each of these six runs. The CFM-based tracking algorithm, by contrast, completely avoids track swaps.

The computation time of tracking algorithms is evaluated on a PC with 2.5 GHz Intel i7 processor. The average computation time per run over 100 Monte Carlo runs is shown in Table 3.2. The IM-based algorithm and the SFM-based algorithm

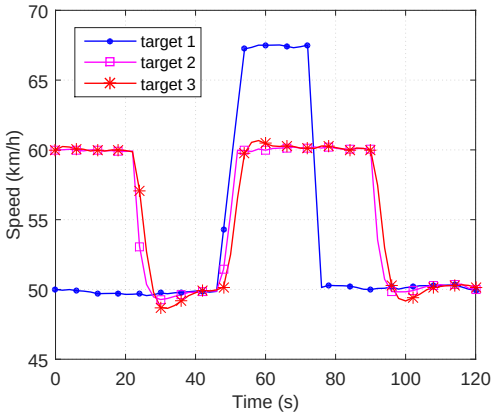


Fig. 3.4: True speed of the three targets in Scenario II.

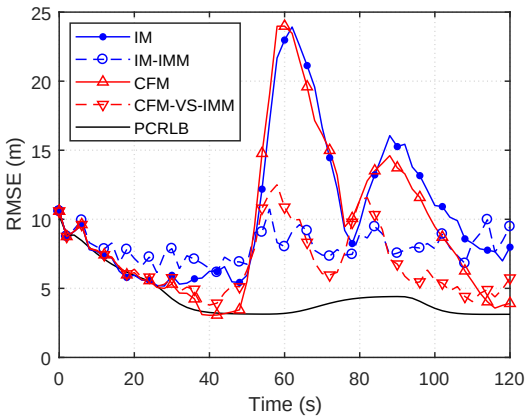
Table 3.3: Statistics of Track Swaps in Scenario II

Algorithm	Number of runs with swap	Statistics of swaps per run		Total number of swaps
		Max	Mean	
IM [46]	39	3	1.41	55
IM-IMM [12]	47	4	1.49	70
CFM	0	–	–	0
CFM-VS-IMM	0	–	–	0

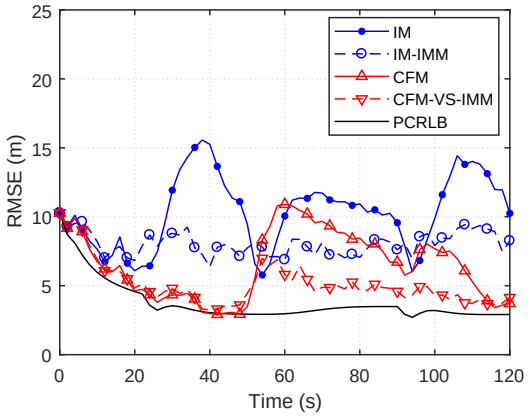
need almost the same computation time since they use the same tracking framework while the SFM-based algorithm additionally employs SFM for state prediction. The CFM-based algorithm requires a slightly higher computation time than other two algorithms.

3.6.2 Scenario II

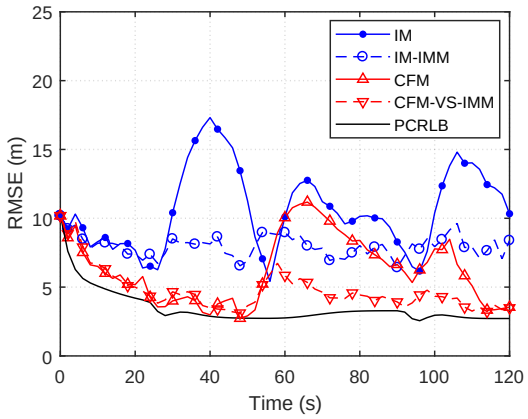
In order to test the ability of the tracking algorithms to adapt to abnormal target maneuvers, a more complex maneuver is assumed in Scenario II. The parameters are the same as those in Scenario I except that the first vehicle exhibits additional maneuvers: an acceleration starting from 46s to 84s with $0.6\,\mathrm{m/s^2}$ and a deceleration



(a) Target 1 position RMSE and PCRLB



(b) Target 2 position RMSE and PCRLB



(c) Target 3 position RMSE and PCRLB

Fig. 3.5: Position and speed RMSE and PCRLB corresponding to the three targets in Scenario II.

from 72s to 76s with $-1.2\,\mathrm{m/s^2}$. The last two vehicles follow the first one accelerating to the desired speed 60 km/h, then keep a constant speed and gradually drop away from the first vehicle thereafter. The last two vehicles catch up with the first one again after its deceleration. The true speeds of the three targets are shown in Fig. 3.4.

The IM-based tracking algorithm [46], the IM-based tracking algorithm with the fixed mode-set IMM [12] and the proposed CFM-based tracking algorithm with and without the VS-IMM are compared on Scenario II. The fixed mode-set IMM is set

Table 3.4: Average computation times of algorithms in Scenario II

	IM	IM-IMM	CFM	CFM-VS-IMM
Average computation time (s)	0.65	0.83	0.70	1.05

according to [12], in which it is used for estimating the longitudinal motion of vehicles, with the nearly constant velocity model and the mean-adaptive acceleration model [33]. The standard deviation of the two process noise levels are set as $\sigma_{v_h} = 0.5\,\mathrm{m/s^2}$ and $\sigma_{v_l} = 0.1\,\mathrm{m/s^2}$ for the VS-IMM estimator, and the transition probabilities are set as $p_{jj} = 0.85$ and $p_{jj'} = \frac{1-p_{jj}}{N}, j' \neq j$, where N is the number of tracks in the corresponding car-following cluster.

The position RMSEs are shown with their corresponding PCRLBs in Fig. 3.5. The figures indicate that the CFM-based tracking algorithms (with dependent-motion assumption) generally outperform the IM-based tracking algorithms (with independent-motion assumption) and, as expected, significant performance improvement is obtained by integrating the fixed IMM or the VS-IMM. The RMSEs of the CFM-based algorithm peak, similar to those of the IM-based algorithm, during target maneuvers but decrease faster thereafter due to the stacked-update mechanism. Note that the PCRLB is much lower than RMSEs in this scenario. The PCRLB derived in Section 3.5 is over-optimistic because it assumes that the exact sequence of motion models as they evolve over times, as can be seen in (3.48), is known, whereas that sequence needs to be learned online in practice. A more realistic lower bound can be found based on [27], but the problems of clusters and CFM models need to be factored in. This is not easily solvable within the PCRLB framework because of discrete uncertainties and time-varying vector dimensions.

Table 3.3 gives the statistics of track swaps in 100 Monte Carlo runs in Scenario II.

Due to the more complex maneuvers, track swaps occur more frequently in Scenario II than in Scenario I for the IM-based tracking algorithms. Though combining with the fixed IMM algorithm yields improvements in RMSEs, it leads to even more track swaps since the mean-adaptive acceleration model increases the adaptivity to maneuvers but at the expense of a large covariance matrix. By contrast, the proposed dependent-motion tracking algorithm with and without VS-IMM avoids track swaps altogether.

Finally, the computation times evaluated on the same device for Scenario II are shown in Table 3.4. Combining with the fixed IMM and VS-IMM algorithms requires extra computation as multiple filters need to be updated. However, the increment is not onerous.

Overall, the proposed CFM-based tracking algorithm with VS-IMM yields the best performance in terms of both RMSEs and track swaps with slightly higher computational cost.

We experimented with varying values of measurement update rate, P_D and λ and evaluated the performances of these algorithms. The observations made with the previous sets of values in scenarios I and II still remained valid. Decreasing P_D reduces the accuracy of tracks, and increases track swaps and track loss due to missing detections. Increasing λ yields the same effects while increasing the number of false tracks. Decreasing the measurement update rate increases track swaps and reduces track accuracy.

3.7 Summary and Conclusions

A novel multi-vehicle tracking algorithm was proposed in this paper. Based on the fact that vehicles interact with other vehicles when moving along roads, road map information and the motion dependence information were integrated into the proposed tracking algorithms by tracking vehicles in the road coordinate with the car-following models (CFM). The notion of the car-following cluster was defined in this paper to incorporate the CFM into the tracking filter. The stacked-vector strategy was then utilized to predict and update the state estimates of all tracks in the same car-following cluster in order to make full use of available measurements. To handle motion deviations from the CFM (for example, some vehicles may not follow the leading vehicle to accelerate when they reach their desired speed), a variable structure interacting multiple model algorithm was also integrated into the proposed CFM-based tracking algorithm.

Numerical simulations to compare the proposed tracking algorithm with other state-of-the-art multi-vehicle tracking algorithms were also proposed. The results demonstrated that the proposed CFM-based tracking algorithms outperform other algorithms, due to better prediction of target maneuvers caused by leading vehicles and the ensuing chain reaction. The VS-IMM effectively handles maneuvers and deviations from the CFM. The proposed CFM-based tracking algorithms avoid track swaps observed in other tracking algorithms.

Bibliography

[1] M. Aron, "Car following in an urban network: Simulation and experiments," *Planning & Transp. Res. & Comput.*, vol. 46, no. 8, pp. 27–39, Jul. 1988.

[2] Y. Bar-Shalom, T. Kirubarajan, and C. Gokberk, "Tracking with classification-aided multiframe data association," vol. 41, no. 3, pp. 868–878, Jul. 2005.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software.* John Wiley & Sons, 2004.

[4] S. Bartoletti, W. Dai, A. Conti, and M. Z. Win, "A mathematical model for wideband ranging," vol. 9, no. 2, pp. 216–228, Mar. 2015.

[5] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," vol. 64, no. 4, pp. 1388–1400, Apr. 2015.

[6] G. A. Bekey, G. O. Burnham, and J. Seo, "Control theoretic models of human drivers in car following," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 19, no. 4, pp. 399–413, Aug. 1977.

[7] D. Betaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," vol. 11, no. 4, pp. 786–798, Dec. 2010.

[8] S. Blackrnan and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA: Artech House, 1999.

[9] V. A. Butakov and P. Ioannou, "Personalized driver assistance for signalized intersections using V2I communication," vol. 17, no. 7, pp. 1910–1919, Jul. 2016.

[10] X. Cao, X. Jiang, X. Li, and P. Yan, "Correlation-based tracking of multiple targets with hierarchical layered structure," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–13, 2016.

[11] L. Chen and C. Englund, "Cooperative intersection management: A survey," vol. 17, no. 2, pp. 570–586, Feb. 2016.

[12] Y. Chen, V. P. Jilkov, and X. R. Li, "Multilane-road target tracking using radar and image sensors," vol. 51, no. 1, pp. 65–80, Jan. 2015.

[13] Y. Cheng and T. Singh, "Efficient particle filtering for road-constrained target tracking," vol. 43, no. 4, pp. 1454–1469, Oct. 2007.

[14] C. Y. Chong, D. Garren, and T. P. Grayson, "Ground target tracking—A historical perspective," in *Proc. IEEE Aerosp. Conf.*, vol. 3, Mar. 2000, pp. 433–448.

[15] R. Ding, M. Yu, H. Oh, and W. H. Chen, "New multiple-target tracking strategy using domain knowledge and optimization," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. PP, no. 99, pp. 1–12, 2016.

[16] Z. Duan and X. R. Li, "The role of pseudo measurements in equality-constrained state estimation," vol. 49, no. 3, pp. 1654–1666, Jul. 2013.

[17] P. Feng, W. Wang, S. Dlay, S. M. Naqvi, and J. Chambers, "Social force model-based MCMC-OCSVM particle PHD filter for multiple human tracking," vol. 19, no. 4, pp. 725–739, Apr. 2016.

[18] D. Franken, "Some notes on Gaussian tracking of objects exhibiting correlated motion," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct. 2014, pp. 1–6.

[19] D. C. Gazis, R. Herman, and R. B. Potts, "Car-following theory of steady-state traffic flow," *Operation Res.*, vol. 7, no. 4, pp. 499–505, Aug. 1959.

[20] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res.Part B*, vol. 15, no. 2, pp. 105–111, Apr. 1981.

[21] G. Golub and C. V. Loan, *Matrix Computations 4th Edition*. Johns Hopkins University Press, 2013.

[22] A. Gorji, R. Tharmarasa, and T. Kirubarajan, "Performance measures for multiple target tracking problems," *Proc. 14th Int. Conf. Inf. Fusion*, pp. 1560–1567, Jul. 2011.

[23] F. Gustafsson, F. Gunnarsson, N. Bergman, U. F. J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation, and tracking," vol. 50, no. 2, pp. 425–437, Feb. 2002.

[24] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous localization and mapping for path-constrained motion," vol. 13, no. 2, pp. 541–552, Jun. 2012.

[25] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[26] W. Helly, "Simulation of bottlenecks in single lane traffic flow." in *Proc. Theory of Traffic Flow*, Apr. 1959, pp. 207–238.

[27] M. Hernandez, B. Ristic, A. Farina, T. Sathyan, and T. Kirubarajan, "Performance measure for Markovian switching systems using best-fitting Gaussian distributions," vol. 44, no. 2, pp. 724–747, Apr. 2008.

[28] K. Jo, M. Lee, J. Kim, and M. Sunwoo, "Tracking and behavior reasoning of moving vehicles based on roadway geometry constraints," vol. 18, no. 2, pp. 460–476, Feb. 2017.

[29] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[30] T. Kirubarajan, Y. Bar-Shalom, and K. R. Pattipati, "Ground target tracking with variable structure IMM estimator," vol. 36, no. 1, pp. 26–46, Jan. 2000.

[31] G. Kravaritis and B. Mulgrew, "Multitarget ground tracking with road maps and particle filters," in *Proc. 5th IEEE Int. Symposium*, Dec. 2005, pp. 253–257.

[32] X. R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," vol. 41, no. 4, pp. 478–493, Apr. 1996.

[33] X. R. Li and V. Jilkov, "Survey of maneuvering targettracking part I: Dynamic models," vol. 39, no. 4, pp. 1333–1364, Jul. 2003.

[34] Y. Li and D. Sun, "Microscopic car-following model for the traffic flow: The state of the art," *Journal of Control Theory and Applications*, vol. 10, no. 2, pp. 133–143, Apr. 2012.

[35] B. Pannetier, V. Nimier, and M. Rombaut, "Multiple ground target tracking with a GMTI sensor," in *IEEE Int. Conf. on MFI*, Sep. 2006, pp. 230–236.

[36] T. H. Rockwell, R. L. Ernst, and A. Hanken, "A sensitivity analysis of empirically derived car-following models," *Transp. Res.*, vol. 2, no. 4, pp. 363–373, Dec. 1968.

[37] S. R. Rogers, "Tracking multiple targets with correlated measurements and maneuvers," vol. 24, no. 3, pp. 313–315, May 1988.

[38] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. Journal of Comput. Vis.*, vol. 104, no. 2, pp. 198–219, Sep. 2013.

[39] P. J. Shea, T. Zadra, D. M. Klamer, E. Frangione, and R. Brouillard, "Improved state estimation through use of roads in ground tracking," in *Proc. SPIE–Signal and Data Processing of Small Targets*, vol. 4048, no. 858, Jul. 2000, pp. 321–332.

[40] Y. Shen, S. Mazuelas, and M. Z. Win, "Network navigation: Theory and interpretation," vol. 30, no. 9, pp. 1823–1834, Oct. 2012.

[41] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization— Part I: A general framework," vol. 56, no. 10, pp. 4956–4980, Oct. 2010.

[42] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," vol. 38, no. 1, pp. 128–136, Jan. 2002.

[43] R. Tharmarasa, T. Kirubarajan, M. Hernandez, and A. Sinha, "PCRLB-based multisensor array management for multitarget tracking," vol. 43, no. 2, pp. 539–555, Apr. 2007.

[44] P. Tichavskỳ, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," vol. 46, no. 5, pp. 1386–1396, May 1998.

[45] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin Heidelberg, Springer-Verlag, 2013.

[46] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[47] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, and J. A. Chambers, "Multi-target tracking and occlusion handling with learned variational bayesian clusters and a social force model," vol. 64, no. 5, pp. 1320–1335, Mar. 2016.

[48] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, 2004.

The following chapter is a reproduction of an Institute of Electrical and Electronics Engineers (IEEE) copyrighted, published paper:

Dan Song, Ratnasingham Tharmarasa, Gongjian Zhou, Mihai C. Florea, Nicolas Duclos-Hindie and Thiagalingam Kirubarajan Multi-Vehicle Tracking Using Microscopic Traffic Models, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 149–161, Jan. 2019. (doi: 10.1109/TITS.2018.2804894)

Chapter 4

Multiple Vehicle Tracking on Multi-lane Roads Using UKF-MHT

4.1 Abstract

In this paper, the multi-vehicle tracking problem is revisited, with greater consideration being given to the interactions between vehicles. Traditionally, algorithms for tracking multiple vehicles in the multi-lane case assume that vehicles move independently of one another and that longitudinal and lateral vehicle dynamics are mutually independent. However, due to traffic volume, limited lane resources, and traffic heterogeneity, vehicles have to interact with neighboring vehicles for the purposes of maintaining a safe distance from the leading vehicle or improving their navigability by passing slower vehicles. To address the limitations in the literature, this

paper proposes a novel multi-vehicle tracking algorithm that integrates the microscopic traffic models (MTM) for modeling interaction behaviors among vehicles in a two-dimensional road coordinate system. Due to the dependence between the longitudinal and later motions, their corresponding estimates are updated sequentially in a recursive manner. An adaptive deferred decision logic is proposed to improve the accuracy of lateral state estimates and thus improve overall performance. Simulation results show that the proposed MTM-based tracking algorithm can achieve better performance than a conventional multi-lane vehicle tracking algorithm with extension to multi-vehicle tracking, which does not consider interactions among vehicles but updates the longitudinal and lateral motion estimates independently.

4.2 Introduction

Vehicle tracking is an important task in intelligent transportation systems (ITS) [7], ground surveillance [30], traffic monitoring [6] and advanced driver assistance systems [12]. This paper aims to address the problem of tracking multiple on-road vehicles.

The central objective in vehicle tracking research is to exploit any extra prior information in order to improve tracking performance. As vehicles move along roads, the on-road constraint or road-map information can be regarded as prior information for tracking; The on-road constraint is considered to be a soft constraint in [17, 18], directly imposed on the filter by introducing directional process noise. By modeling roads as sequences of linear segments, the problem of vehicle tracking with on-road constraints is treated in [26] as state estimation with linear equality constraints and it has been demonstrated that the optimal solution is the projection of Kalman filter

estimates onto the road segments [26]. Other optimization-based filters [24, 10] are also capable of addressing the on-road target tracking problem. Recently, the road-map information has been incorporated into vehicle tracking in road coordinates. The constrained state estimation problem then becomes the standard unconstrained problem: In [30], a single vehicle is tracked in a one-dimensional road coordinate system in which only the longitudinal motion of the vehicle is considered; thus, this approach is only applicable in the single-lane case. This method is extended to multi-lane case in [8] by adding an extra dimension to model multiple lanes. Then, the longitudinal and lateral motions of the vehicle are separately estimated by the adaptive mileage estimator (AME) and the lane filter (LF), respectively.

Recently, interactions between targets while moving have been considered as another source of prior information. For example, the social force model proposed in [13] to model the interactions among pedestrians has been used in pedestrian tracking [31, 11]. This approach is modified to model interactions among vehicles and used in vehicle tracking in [10]. However, using a virtual force to model interactions lacks direct physical meaning and thus it may not always be appropriate. In contrast, the microscopic traffic model (MTM) [28], specifically developed to describe the motion of a vehicle in the presence of interactions with surrounding traffic, has been extensively studied in the area of traffic theory for decades and has been widely used in many microscopic traffic simulators and in adaptive cruise control (ACC) [23]. The MTM consists of car-following and lane-changing models that describe the longitudinal and lateral dynamics of vehicles, respectively. In particular, the car-following model describes vehicle dynamics while driving along a single lane by maintaining a safe distance from the leading vehicle. The most well-known type of car-following

model is the General Motors (GM) model [20], in which the acceleration of each vehicle is given as a function of its motion relative to the leading vehicle. Methods based on this model include the Gazis-Herman-Rothery (GHR) [22] model and the intelligent drive model (IDM) [29], among others. The collision avoidance (CA) model, another type of car-following model, describes the longitudinal motion of vehicles by defining a safe distance for avoiding collisions. The most common CA model is the Gipps model [9]. A comprehensive literature review on the car-following model is available in [20].

The lane-changing model describes vehicle dynamics across multiple lanes for making a lane change in order to improve navigability by surpassing slower vehicles or to evade obstructions. A lane change is typically considered a two-step process: first, lane-change decision-making and then the consequent actions, such as steering and acceleration. At the level of a microscopic traffic model, only the operational decision process is considered and it is assumed that the lane change takes place instantaneously. Various lane-changing models have been presented to model the process of rational lane-changing decision-making, such as the discrete-choice-based Ahmed's model [1] in which the lane-changing decision-making is based on a dynamic discrete choice model, and the incentive-based MOBIL (minimizing overall braking induced by lane change) model, in which two criteria (incentive criterion and safety criterion) are defined and lane-changing is executed once both criteria are met. A systematic comparison of lane-changing models can be found in [23].

Although comprehensive studies, in which road-map information is used in tracking, have been carried out, most vehicle tracking algorithms still assume that vehicles move independently of one another. To the best of our knowledge, the interactions

between vehicles are considered in [25], in which the GHR model is used for candidate track probabilities evaluation in the multiple hypothesis tracking (MHT) framework. In [10], the social force model is modified and used to describe vehicle motions for multi-vehicle tracking. However, as mentioned above, using the social force model to describe vehicle motion is not appropriate. The car-following model is used in [27] to track multiple vehicles on single-lane roads. Given this gap in the research, this paper aims to comprehensively consider the interactions between vehicles by incorporating microscopic traffic models in a two-dimensional (2-D) road representation framework for tracking multiple vehicles in the multi-lane case. To achieve this, the longitudinal motion estimation is treated as a continuous-state estimation problem, with the IDM describing longitudinal vehicle dynamics. The lateral motion estimation, however, is treated as a discrete-state estimation problem, with the MOBIL model describing lateral vehicle dynamics. Since the IDM and MOBIL models are mutually dependent, two estimation problems are sequentially updated in a recursive manner: In each scan, K-best lane-changing hypotheses are generated based on the lane-changing probability of each vehicle evaluated by the MOBIL model. Then each hypothesis is predicted and a data association algorithm is applied to associate measurements to tracks under each hypothesis. After evaluating the quality of these hypotheses, the most likely one is selected and updated as the estimate for the current scan. An adaptive deferred decision logic is also proposed to defer hard-decision making in anticipation that subsequent measurements will resolve the uncertainty, thus improving overall performance.

The rest of the paper is structured as follows: Section 4.3 introduces the representation of vehicle kinematics in the 2-D road coordinate system and the corresponding

measurement model. In Section 4.4, the IDM and MOBIL models are introduced. The vehicle tracking algorithm for multi-lane case proposed in [8] is briefly reviewed and extended to track multiple vehicles in Section 4.5. A new multi-vehicle tracking algorithm, with integrated IDM and MOBIL models, is proposed in Section 4.6. Numerical results are provided in Section 4.7 to show the merits of the proposed multi-vehicle tracking algorithm over the algorithm that does not consider interactions between vehicles. Conclusions are presented in Section 4.8.

4.3 Representation of Vehicle Kinematics in 2-D Road Coordinates

4.3.1 2-D Road Coordinate System

Roads are modeled by a sequence of connected linear segments in most geographical information systems (GIS) [3] as well as in vehicle tracking [30]. This road model is extended to 2-D [33] in that another dimension is added to take road width information into account. For simplicity, altitude information and road errors are not considered in this paper. Given these conditions, any point V on the road can be uniquely defined by $V = (l, d)^{\mathrm{T}}$, where $(\cdot)^{\mathrm{T}}$ denotes the transpose. Here, l refers to the mileage coordinate, which is the arc length of the projection of V onto the center line of the road from the reference starting point of the road, and d refers to the displacement coordinate, which is the signed displacement of V from the center line of the road. Thus the coordinate of V in the Cartesian coordinate system $V = (x, y)^{\mathrm{T}}$

can be obtained through the road-to-Cartesian coordinate transformation as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{s}_k + (l - l_k)\mathbf{r}_k + d\mathbf{d}_k \tag{4.1}$$

where the kth segment is the segment where the projection of V onto the center line is located, and $\mathbf{s}_k$ is the coordinate of the starting point of the kth segment. The term l_k denotes the arc length of the starting point of the kth segment from the reference starting point of the road, which is given by $l_k = \sum_{j=1}^{k-1} \epsilon_j$, where ϵ_j is the length of the jth segment. We define $\mathbf{r}_k$ as the unit direction vector and and $\mathbf{d}_k$ as the unit displacement vector from the left lane to the right lane of the kth segment. Given $\mathbf{r}_k = (x_{r_k}, y_{r_k})^{\mathrm{T}}$, $\mathbf{d}_k = (y_{r_k}, -x_{r_k})^{\mathrm{T}}$. Finally, k is determined by the following inequality:

$$l_k \leq l \leq l_k + \epsilon_k. \tag{4.2}$$

4.3.2 Representation of Vehicle Kinematics in 2-D Road Coordinates

The kinematic state of vehicles can be decomposed into the longitudinal state and the lateral state, corresponding to the vehicle motion in two dimensions of the road coordinate system. The longitudinal state is defined by its position $l(t)$ and speed $v(t) \triangleq \dot{l}(t)$ in the mileage coordinate, i.e.,

$$x_l(t) \triangleq [l(t), v(t)]^{\mathrm{T}}. \tag{4.3}$$

Assuming that the lane change itself takes place instantaneously, the lateral state, the state of a vehicle in the displacement coordinate, can be defined by the lane in which the vehicle is currently traveling, $I(t)$. By stacking the longitudinal state and the lateral state, the kinematic state of vehicles in 2-D road coordinates is given by

$$x(t) \triangleq [l(t), v(t), I(t)]^{\mathrm{T}}. \tag{4.4}$$

4.3.3 Measurement Model

A wide range of sensors are used in the literature and in real systems for target localization and tracking. Radar and cameras are the primary sensors in uncooperative applications, while the Global Positioning System (GPS) and Inertial Navigation System (INS) are generally used in cooperative applications. No matter which sensor is used, position data is the most common information it provides. Without loss of generality and for simplicity, we assume that measurements only depend on target position and have already been projected [26] onto the road coordinate system before being fed into the tracker. Since sensors may be influenced by various factors in harsh environment, such as noise, multipath and clutter, the received measurement are inaccurate and may be contaminated by mis-detections and false alarms. Thus, target-originated measurements are assumed to be detected with probability P_D and the measurement of a vehicle with state vector x is given by

$$z \triangleq \begin{bmatrix} z_l \\ z_d \end{bmatrix} = \begin{bmatrix} l \\ d(I) \end{bmatrix} + \begin{bmatrix} w_l \\ w_d \end{bmatrix} \tag{4.5}$$

where z_l and z_d are the mileage and displacement measurements, respectively; $w_l \sim \mathcal{N}(0, \sigma_l^2)$ and $w_d \sim \mathcal{N}(0, \sigma_d^2)$ are the Gaussian measurement noise components that perturb z_l and z_d, respectively. Assuming that vehicles drive along the center-line of a lane, $d(I)$ denotes the displacement of the center of the Ith lane (from left to right) from the center of the road given by

$$d(I) = (2I - L - 1)\Delta \tag{4.6}$$

where the road segment has L lanes, each with width 2Δ. It is assumed that false alarms are uniformly distributed in the field of the view and their cardinality follows Possion distribution with parameter λ. We denote the set of all measurements received at time step k as

$$Z(k) \triangleq \{z^m(k),\ m = 1, 2, ..., M(k)\} \tag{4.7}$$

and the cumulative set of measurements available up to time step k as

$$Z^k \triangleq \{Z(i),\ i = 1, 2, ..., k\}. \tag{4.8}$$

4.4 Microscopic Traffic Models

Vehicle dynamics can also be decomposed into longitudinal dynamics and lateral dynamics, which are modeled by the car-following model and lane-changing model, respectively, in microscopic traffic models.

Longitudinal dynamics — Car-following model

Table 4.1: Model parameters of the IDM on highways [16]

Parameter	Car	Truck
Desired speed v_0	120 km/h	80 km/h
Time gap T	1.0s	1.5s
Minimum gap s_0	2.0m	4.0m
Acceleration exponent δ	4	4
Maximum acceleration $a_{\max}$	1.5 m/s^2	0.7 m/s^2
Desired deceleration b	2.0 m/s^2	2.0 m/s^2

The longitudinal dynamics refers to the motion of a vehicle that moves along the current lane by following a leading vehicle. In this paper, the longitudinal dynamics is described by the IDM car-following model [29], in which the acceleration of the vehicle $a(t) \triangleq \ddot{l}(t)$ is defined as a function of its longitudinal state $x_l(t)$ and that of its leader $x_l^l(t)$ as

$$a(t) = \text{IDM}\left[x_l(t), x_l^l(t)\right] = a_{\max}\left\{1 - \left[\frac{v(t)}{v_0}\right]^{\delta} - \left[\frac{s^\star(t)}{s(t)}\right]^2\right\} \tag{4.9}$$

where $s(t) = l_l(t) - l(t)$ and $s^\star(t)$ is the desired distance given by

$$s^\star(t) = s_0 + \max\left[0, v(t)T + \frac{v(t)\Delta v(t)}{2\sqrt{a_{\max}b}}\right] \tag{4.10}$$

with $\Delta v(t) = v(t) - v_l(t)$. The definition of parameters in (4.9) and (4.10) and the typical values of these parameters for two different types of vehicles on highways [15, 16] are shown in Table 4.1.

Note that when there is no leading vehicle or an obstruction (in cases of accidents or where the lane comes to an end) ahead, the IDM is still applicable by introducing

a virtual leader with the static longitudinal state $x_l^{no} = [\infty\,\mathrm{m}, 0\,\mathrm{m/s}]^{\mathrm{T}}$ for no leading vehicle and $x_l^{ob} = [l_{ob}\,\mathrm{m}, 0\,\mathrm{m/s}]^{\mathrm{T}}$ for obstruction, respectively, where l_{ob} is the mileage coordinate of where the obstruction is.

Lateral dynamics — Lane-changing model

The lateral dynamics refers to the motion of a vehicle for lane changes. Assuming that lane changes take place instantaneously, only the operational decision process needs to be considered and is modeled by the MOBIL lane-changing model [15] in this paper since it can be seamlessly unified with longitudinal dynamics and the IDM.

A rational lane-change decision is made when a prospective lane change brings the anticipated advantages more than disadvantages, subject to ensuring safety. The MOBIL method models advantages, disadvantages and the safety criterion as functions of the (longitudinal) acceleration of three vehicles: the vehicle considering a lane change, the current following vehicle and the new following vehicle in the neighboring target lanes, denoted as vehicle c, f, and $\tilde{f}$, respectively. Denote the (longitudinal) acceleration difference of these vehicles after possible lane-changing as Δa_c, Δa_f, and $\Delta a_{\tilde{f}}$, respectively, which can be written as

$$\Delta a_r = a_{\max}\left[\left(\frac{s_r^\star}{s_r}\right)^2 - \left(\frac{\tilde{s}_r^\star}{\tilde{s}_r}\right)^2\right], \quad r = c, f, \tilde{f} \tag{4.11}$$

if the IDM given by (4.9) and (4.10) is used. Here, the time symbol t is omitted and the tilde represents variables after the possible lane change. If vehicle f or $\tilde{f}$ does not exist, $\Delta a_f = 0$ or $\Delta a_{\tilde{f}} = 0$, respectively. Then, the incentive criterion is given by

$$\Delta a \triangleq \Delta a_c + p\left(\Delta a_f + \Delta a_{\tilde{f}}\right) > \Delta a_{\mathrm{th}} \tag{4.12}$$

where Δa is the weighted overall acceleration difference, with p being the politeness factor that reflects the degree of altruism of the driver and Δa_{th} is the switching threshold that prevents a lane change if it brings only a marginal advantage. The safety criterion checks the possibility of lane-changing by limiting the deceleration imposed on the new following vehicle in the target lane to be smaller than a given safe limit b_{safe} after lane-changing. That is,

$$\tilde{a}_{\tilde{f}} \geq -b_{\text{safe}}. \tag{4.13}$$

A lane change will be executed if both the incentive criterion and the safety criterion are met. However, (4.12) has already considered the potential deceleration of $\tilde{a}_{\tilde{f}}$ as long as p is not too small, and thus a lane change that violates the safety criterion would be automatically excluded by (4.12). In this case, no additional safety constraint is needed. When left lane-changing and right lane-changing are feasible in the meantime, the change is executed for the lane with a larger Δa.

4.5 AMELF-based Multi-vehicle Tracking in Road Coordinates

In [8], a single vehicle tracking algorithm for multi-lane case is proposed. It can be easily extended to track multiple vehicles by combining it with a data association algorithm and a track maintenance scheme [4]. This algorithm is referred to as the AMELF-based tracking algorithm in this paper and is briefly reviewed and extended to multi-vehicle tracking in this section.

4.5.1 Longitudinal Motion Estimator

The AME, which implements the interacting multiple model (IMM) algorithm with the nearly constant velocity (NCV) model and the mean-adaptive acceleration (MAA) model, is used to estimate the longitudinal state of vehicle over time.

With the augmented longitudinal state $x_{al}(t) \triangleq [l(t), v(t), a(t)]^{\mathrm{T}}$, where $a(t)$ is the longitudinal acceleration, both the NCV model and the MAA model can be expressed as linear discrete-time kinematic models, whose generic form [2] is

$$x_{al}(k) = Fx_{al}(k-1) + Gu(k-1) + \Gamma v(k-1) \tag{4.14}$$

where k is the discrete time step, $u(k-1)$ is the input vector, and $v(k-1)$ is the zero-mean white Gaussian process noise with covariance Q. The details on the NCV model and the MAA model can be found in [8, 19].

Based on the NCV model and MAA model, two Kalman filters (KF) are used within the framework of IMM estimation for maneuvering target tracking [2]. Let M_1 denote the NCV model and M_2 denote the MAA model.

4.5.2 Lateral Motion Estimator

The lateral motion of vehicles in [8] is modeled as a lane sequence $I(k)_{k=0,1,...}$ with the characteristics of a homogeneous Markov chain, whose transition probability matrix (TPM) $\Pi^I = [p_{ij}^I]_{i,j=1}^L$ and initial probability vector $\mu^I(0) = [\mu_1^I(0), ..., \mu_L^I(0)]^{\mathrm{T}}$ are assumed to be known, where

$$p_{ij}^I \triangleq P\left\{I(k) = j | I(k-1) = i\right\},\ i, j = 1, ..., L \tag{4.15}$$

$$\mu_i^I(0) \triangleq P\left\{I(0)=i\right\},\ i=1,...,L. \tag{4.16}$$

Then, the probability of the vehicle on lane j at time step k conditioned on Z^k,

$$\mu_j^I(k) \triangleq P\left\{I(k)=j|Z^k\right\}, \tag{4.17}$$

is updated recursively by lane filter (LF) as

$$\mu_j^I(k) = \frac{1}{c}\Lambda_j^I(k)\mu_j^I(k|k-1) \tag{4.18}$$

where c is the normalization factor

$$c = \sum_{j=1}^{L}\Lambda_j^I(k)\mu_j^I(k|k-1), \tag{4.19}$$

$\mu_j^I(k|k-1)$ is the predicted probability defined as

$$\mu_j^I(k|k-1) \triangleq P\left\{I(k)=j|Z^{k-1}\right\} = \sum_{i=1}^{L} p_{ij}^I\mu_i^I(k-1),\ j=1,\ldots,L \tag{4.20}$$

and $\Lambda_j^I(k)$ is the likelihood function corresponding to $I(k)=j$, defined as

$$\Lambda_j^I(k) \triangleq p\left[z_d(k)|I(k)=j\right] \tag{4.21}$$

where $z_d(k)$ is the displacement measurement used to update the lateral state at time step k. Based on (4.5) and (4.6), $\Lambda_j^I(k)$ is given by

$$\Lambda_j^I(k) = \mathcal{N}\left[z_d(k); d(j), \sigma_d^2\right]. \tag{4.22}$$

The LF chooses the lane with the largest $\mu_i^I(k)$ as the optimal lane estimate $\hat{I}(k)$ at time step k. That is,

$$\hat{I}(k) = \arg \max_{1 \leq i \leq L} \mu_i^I(k). \tag{4.23}$$

4.5.3 AMELF-Based Multi-vehicle Tracking Algorithm

The measurement origin uncertainty needs to be resolved for multi-vehicle tracking with non-unity probability of detection and non-zeros false alarms. The two-dimensional assignment algorithm is a widely-used data association mechanism to handle the measurement origin uncertainty. For details about the 2-D assignment algorithm, the reader is referred to [4].

Let $\Lambda(k, m, n)$ denote the likelihood function of the nth track being associated with the mth measurement, $z^m(k) = [z_l^m(k), z_d^m(k)]^{\mathrm{T}}$, at time step k. Due to the simplified measurement model and the assumed independence between longitudinal and lateral motion estimators, it can be evaluated by two independent parts, respectively. That is,

$$\Lambda(k, m, n) = \Lambda_l(k, m, n)\Lambda_d(k, m, n) \tag{4.24}$$

where $\Lambda_l(k, m, n)$ and $\Lambda_d(k, m, n)$, the likelihood function in the mileage and displacement coordinate, respectively, can be expressed as

$$\Lambda_l(k, m, n) = \sum_{j=1,2} \mu_{n,j}^l(k|k-1)\mathcal{N}\left[z_l^m(k); \hat{z}_l^{n,j}(k|k-1), S_l^{n,j}(k)\right] \tag{4.25}$$

$$\Lambda_d(k,m,n) = \sum_{j=1}^{L} \mu_{n,j}^{I}(k|k-1)\mathcal{N}\left[z_d^m(k); d(j), \sigma_d^2\right], \tag{4.26}$$

where $\hat{z}_l^{n,j}(k|k-1)$ is the predicted mileage measurement for the nth track conditioned on mode M_j with the variance being $S_l^{n,j}(k)$, and $\mu_{n,j}^{l}(k|k-1)$ is the predicted mode probability of M_j for the nth track [2]. Substituting $\Lambda(k,m,n)$ (as defined in (4.24)) into the cost function, the 2-D assignment algorithm obtains the best measurement-to-track association with the minimum global cost [4].

Before applying 2-D assignment, a gating technique [4] is used to eliminate false alarms and reduce the number of candidate assignments. The validation gate for the nth track is typically an ellipsoid defined as

$$[z(k) - \hat{z}^n(k|k-1)]^{\mathrm{T}} S^n(k)^{-1} [z(k) - \hat{z}^n(k|k-1)] \leq \gamma \tag{4.27}$$

where γ is the probability threshold that determines the gate volume. $\hat{z}^n(k|k-1) = [\hat{z}_l^n(k|k-1), \hat{z}_d^n(k|k-1)]^{\mathrm{T}}$ is the predicted measurement for the nth track at time step k with covariance matrix $S^n(k) = \mathrm{diag}(S_l^n(k), S_d^n(k))$, where $\mathrm{diag}(\cdot)$ denotes the (block) diagonal matrix construction operator, $\hat{z}_l^n(k|k-1)$ and $\hat{z}_d^n(k|k-1)$ are the predicted mileage and displacement measurements for the nth track given by

$$\hat{z}_l^n(k|k-1) = \sum_{j=1,2} \hat{z}_l^{n,j}(k|k-1)\mu_{n,j}^{l}(k|k-1) \tag{4.28}$$

and

$$\hat{z}_d^n(k|k-1) = \sum_{j=1}^{L} d(j)\mu_{n,j}^{I}(k|k-1), \tag{4.29}$$

respectively, with corresponding variances being

$$S_l^n(k) = \sum_{j=1,2} \mu_{n,j}^l(k|k-1) \left\{ \left[\hat{z}_l^{n,j}(k|k-1) - \hat{z}_l^n(k|k-1) \right] \right.$$
$$\left. \times \left[\hat{z}_l^{n,j}(k|k-1) - \hat{z}_l^n(k|k-1) \right]^{\mathrm{T}} + S_l^{n,j}(k) \right\} \tag{4.30}$$

and

$$S_d^n(k) = \sigma_d^2 + \sum_{j=1}^{L} \mu_{n,j}^l(k|k-1) \left[d(j) - \hat{z}_d^n(k|k-1) \right] \left[d(j) - \hat{z}_d^n(k|k-1) \right]^{\mathrm{T}}, \tag{4.31}$$

respectively. Only those measurements falling within the validation gate of the nth track are considered for association with this track.

The AME and LF combined with the 2-D assignment algorithm is now applicable for multi-vehicle tracking in the multi-lane case.

4.6 MTM-based Multi-vehicle Tracking in Road Coordinates

Although the AMELF-based vehicle tracking algorithm is extended to handle the multi-vehicle tracking problem, it assumes that vehicles move independently and thus cannot exploit motion dependence information during tracking. The MTM-based multi-vehicle tracking algorithm proposed in this section explicitly considers the interactions among vehicles by integrating the IDM and MOBIL models into estimators.

4.6.1 Longitudinal Motion Estimator

The longitudinal dynamics of a vehicle evolves based on the IDM model. However, the desired speed of vehicles are usually different in practical scenarios and unknown to the estimator. Thus, it is considered as another parameter that needs to be estimated, and is inserted into the augmented longitudinal state, $x_{al}(t) = [l(t), v(t), v_0]^{\mathrm{T}}$. The evolution of the augmented longitudinal state from time step $k-1$ to k can also be described by the generic form (4.14) with

$$F = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad G = \Gamma = \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix} \tag{4.32}$$

where T is the update interval, $u(k-1) = a(k-1)$ given by the IDM (4.9) and $v(k-1)$ represents the mismatch in the acceleration between the IDM and the real vehicle motion. Due to the high nonlinearity of (4.9) and (4.10), the unscented transformation (UT) [14, 32] is applied to predict longitudinal motion by the steps shown below.

Generate sigma points

Sigma points are generated as the following:

$$\begin{aligned}
&\mathcal{X}_0(k-1) = \hat{x}_l^s(k-1) \\
&\mathcal{X}_i(k-1) = \hat{x}_l^s(k-1) + \left[\sqrt{(N_x^s + \lambda)P_l^s(k-1)}\right]_i, i = 1, ..., N_x^s \\
&\mathcal{X}_i(k-1) = \hat{x}_l^s(k-1) + \left[\sqrt{(N_x^s + \lambda)P_l^s(k-1)}\right]_{i-N_x^s}, i = N_x^s + 1, ..., 2N_x^s
\end{aligned} \tag{4.33}$$

where $\hat{x}_l^s(k-1) = \left[\hat{x}_{al}(k-1|k-1)^{\mathrm{T}}, \hat{x}_l^l(k-1|k-1)^{\mathrm{T}}\right]^{\mathrm{T}}$ with dimension N_x^s and its associated approximate covariance matrix being $P_l^s(k-1) = \mathrm{diag}(P_{al}(k-1|k-1), P_l^l(k-1|k-1))$, which ignores the cross-correlation between $\hat{x}_{al}(k-1|k-1)$ and $\hat{x}_l^l(k-1|k-1)$. $\hat{x}_l^l(k-1|k-1)$, the longitudinal state estimate of the leader, and its associated covariance matrix $P_l^l(k-1|k-1)$ can be found as a part of $\hat{x}_{al}^l(k-1|k-1)$ and $P_{al}^l(k-1|k-1)$, respectively. In (4.33), $\lambda = \alpha^2(N_x^s + \kappa) - N_x^s$ and $\left[\sqrt{(N_x^s+\lambda)P_l^s(k-1)}\right]_i$ is the ith column of the matrix square root of $(N_x^s+\lambda)P_l^s(k-1)$. We decompose $\mathcal{X}_i(k-1)$, the ith sigma point, into a form similar to $\hat{x}_l^s(k-1)$ denoting as $\mathcal{X}_i(k-1) = \left[\hat{x}_{al}^i(k-1|k-1)^{\mathrm{T}}, \hat{x}_l^{l,i}(k-1|k-1)^{\mathrm{T}}\right]^{\mathrm{T}}$.

Predict sigma points

Instantiating each sigma point using the discrete-time model in (4.14) with (4.32), the transformed sigma point $\mathcal{Y}_i(k-1)$ is given by

$$\mathcal{Y}_i(k-1) = F\hat{x}_{al}^i(k-1|k-1) + G \times \mathrm{IDM}\left[\hat{x}_{al}^i(k-1|k-1), \hat{x}_l^{l,i}(k-1|k-1)\right] \quad (4.34)$$

Calculate the predicted longitudinal state $\hat{x}_{al}(k|k-1)$ and the associated covariance $P_{al}(k|k-1)$

$$\hat{x}_{al}(k|k-1) = \sum_{i=0}^{2N_x^s} W_i^{(m)} \mathcal{Y}_i(k-1) \quad (4.35)$$

$$P_{al}(k|k-1) = \Gamma Q \Gamma^{\mathrm{T}} + \sum_{i=0}^{2N_x^s} W_i^{(c)} \left[\mathcal{Y}_i(k-1) - \hat{x}_{al}(k|k-1)\right]\left[\mathcal{Y}_i(k-1) - \hat{x}_{al}(k|k-1)\right]^{\mathrm{T}} \quad (4.36)$$

where $W_i^{(m)}$ and $W_i^{(c)}$ are the weights associated with $\mathcal{X}_i(k-1)$ for estimating the mean and covariance, respectively, which are given by

$$W_0^{(m)} = \frac{\lambda}{N_x^s + \lambda}, W_0^{(c)} = \frac{\lambda}{N_x^s + \lambda} + (1 - \alpha^2 + \beta)$$
$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(N_x^s + \lambda)}, \quad i = 1, ..., 2N_x^s. \tag{4.37}$$

For the special case of no leader or an obstruction ahead, sigma points are then generated based on $\hat{x}_l^s(k-1) = \hat{x}_{al}(k-1|k-1)$ and $P_l^s(k-1) = P_{al}(k-1|k-1)$; the set of transformed sigma points is given by

$$\mathcal{Y}_i(k-1) = F\hat{x}_{al}^i(k-1|k-1) + \ G \times \text{IDM}\left[\hat{x}_{al}^i(k-1|k-1), x_l^l\right] \tag{4.38}$$

where x_l^l is given by x_l^{no} or x_l^{ob}.

One potential issue is the premature convergence of $\hat{v}_0(k)$, which is caused by identifying v_0 as the static parameter during estimation. To address this, an "artificial process noise" is added in (4.36) to increase the variance of $\hat{v}_0(k)$ in $P_{al}(k|k-1)$. After $\hat{x}_{al}(k|k-1)$ and $P_{al}(k|k-1)$ are obtained, $\hat{x}_{al}(k|k)$ can be obtained by employing the normal KF equations [2]. However, since the IDM classifies vehicles into two types (i.e., cars and trucks) with different model parameters (see Table 4.1), two KFs are set up corresponding to the two vehicle types, respectively, resulting in the static multiple model estimator [4]. The posterior probability of model j being correct given Z^k can be recursively evaluated as

$$P\{M_j|Z^k\} = \frac{p\left[z(k)|Z^{k-1}, M_j\right] P\{M_j|Z^{k-1}\}}{\sum_{i=1,2} p\left[z(k)|Z^{k-1}, M_j\right] P\{M_j|Z^{k-1}\}} \tag{4.39}$$

where $p\left[z(k)|Z^{k-1}, M_j\right]$ is the likelihood function of model j at time step k, which is approximated by

$$p\left[z(k)|Z^{k-1}, M_j\right] = \mathcal{N}\left[z_l(k); \hat{z}_l^j(k|k-1), S_l^j(k)\right]. \tag{4.40}$$

Once $P\{M_j|Z^k\}$ is greater than a pre-defined threshold P_{th}^l, a hard-decision that model j is correct is made and the other model-matched filter is deleted.

4.6.2 Lateral Motion Estimator

The MOBIL model is used to predict lateral vehicle dynamics. However, due to the uncertainty in the longitudinal state estimate, the test static of the MOBIL model (4.12) is a random variable and thus only the lane-changing probability can be evaluated. Based on this, the k-best lane-changing hypotheses are generated and maintained. Once the measurements become available, each lane-changing hypothesis is evaluated. An adaptive deferred decision logic as hypothesis-oriented multiple hypothesis tracking (MHT) [5] is proposed to choose the most likely hypothesis.

Lane-changing probability evaluation

The lane-changing probability is evaluated based on the incentive criterion (4.12) of the MOBIL model. Assuming all lane changes are performed at the beginning of each time step, the test statistic at time step k, $\Delta a(k)$, is determined by the updated longitudinal state estimate at time step $k-1$ for the following five vehicles: vehicles c, f, $\tilde{f}$, l and $\tilde{l}$ where l and $\tilde{l}$ denote the current leading vehicle of c and the new leading vehicle in the neighboring target lane, respectively.

The distribution of $\Delta a(k)$ is approximated by a Gaussian probability density function with mean $\Delta\bar{a}(k)$ and variance $\sigma^2_{\Delta a}(k)$ via second-order moment matching [30]. The terms $\Delta\bar{a}(k)$ and $\sigma^2_{\Delta a}(k)$ can be estimated by UT as described in Section 4.6.1–4.6.1, but $\hat{x}^s_l(k-1)$ in (4.33) is replaced by stacking the longitudinal state estimates of vehicles $c, l, f, \tilde{l}$ and $\tilde{f}$ (if existent), in that order respectively; Its associated covariance $P^s_l(k-1)$ is approximated by ignoring the cross-correlation between the estimates of different vehicles. The transformed sigma point $\mathcal{Y}_i(k-1)$ now is Δa_i obtained by instantiating $\mathcal{X}_i(k-1)$ into (4.12); $\Delta\bar{a}(k)$ and $\sigma^2_{\Delta a}(k)$ are then given by (4.35) and (4.36) without the term $\Gamma Q \Gamma^{\mathrm{T}}$, respectively. For example, if there is an obstruction ahead, no following vehicle in the current lane and no leading vehicle in the target lane, 9 sigma points are generated by (4.33) with $\hat{x}^s_l(k-1) = [\hat{x}^c_l(k-1|k-1)^{\mathrm{T}}, \hat{x}^{\tilde{f}}_l(k-1|k-1)]^{\mathrm{T}}$ and $P^s_l(k-1) \approx \mathrm{diag}(P^c_l(k-1|k-1), P^{\tilde{f}}_l(k-1|k-1))$. In $\mathcal{Y}_i(k-1)$, $\Delta a_f = 0$, Δa_c and $\Delta a_{\tilde{f}}$ are obtained based on (4.10) and (4.11), with $\hat{x}^{c,i}_l(k-1|k-1)$, x^{ob}_l and x^{no}_l, and with $\hat{x}^{\tilde{f},i}_l(k-1|k-1)$, x^{no}_l and $\hat{x}^{c,i}_l(k-1|k-1)$, respectively.

Finally, the probability of vehicle c executing a lane change at time step k, $P\{C|k\}$, is given by

$$P\{C|k\} = P\left\{\Delta a(k) > \Delta a_{th}\right\} = 1 - \Phi\left[\frac{\Delta a_{th} - \Delta\bar{a}(k)}{\sigma_{\Delta a}(k)}\right] \tag{4.41}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian distribution. When the left lane and the right lane are both available to change into, the test statistic for these two cases, $\Delta a_L(k)$ and $\Delta a_R(k)$, are evaluated, respectively, and the corresponding probabilities are approximated by

$$P\{UC|k\} \approx \min_{i=L,R} P\left\{\Delta a_i(k) < \Delta a_{th}\right\}$$

$$= \min_{i=L,R} \Phi \left[\frac{\Delta a_{th} - \Delta \bar{a}_i(k)}{\sigma_{\Delta a_i}(k)} \right] \tag{4.42}$$

$$P\{LC|k\} = (1 - P\{UC|k\})\, P\{\Delta a_R(k) < \Delta a_L(k)\}$$

$$\approx (1 - P\{UC|k\})\, \Phi \left[\frac{\Delta \bar{a}_L(k) - \Delta \bar{a}_R(k)}{\sqrt{\sigma^2_{\Delta a_L}(k) + \sigma^2_{\Delta a_R}(k)}} \right] \tag{4.43}$$

$$P\{RC|k\} = 1 - P\{UC|k\} - P\{LC|k\} \tag{4.44}$$

where $P\{UC|k\}, P\{LC|k\}$ and $P\{RC|k\}$ are the probabilities of no lane-change, left lane-change, and right lane-change, respectively. Assuming that each vehicle can only change one lane at a time, $\mu_j^I(k|k-1)$, defined in (4.20), is now given by

$$\mu_j^I(k|k-1) = \begin{cases} P\{LC|k\}, & j = \hat{I}(k-1) - 1 \\ P\{UC|k\}, & j = \hat{I}(k-1) \\ P\{RC|k\}, & j = \hat{I}(k-1) + 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.45}$$

K-best lane-changing hypotheses generation

The lane-changing hypothesis at time step k, $\theta_i(k)$, is defined as a combination of possible lane-changing decisions made by each vehicle

$$\theta_i(k) = \left\{ \hat{I}_n(k), n = 1, ..., N \right\}. \tag{4.46}$$

To avoid the potential combinatorial explosion in the number of hypotheses, given the state estimates in the previous scan, only the K-best hypotheses are extracted in each scan by Algorithm 1 without enumerating all possible hypotheses, under

Algorithm 1: K-best lane-changing hypotheses generation algorithm

Initialization:

Initialize current hypothesis $\theta_0(k)$ with $P\{\theta_0(k)\} = 1$;
Initialize number of formed hypotheses $\theta_i(k), k_h = 0$;
$P_{\max} \triangleq \max_{i=1,...,k_h} P\{\theta_i(k)\} = 0$;
Find sorted index of tracks TI by mileage coordinate estimates in descending order;
$n = 1$;

Hyotheses generation:

Identify the leading and following tracks in the current and neighboring lanes for track TI(n);
Evaluate lane-changing probability and corresponding $\mu^I_{\mathrm{TI}(n),j}(k|k-1), j = 1, ..., L$;
Find sorted indices of lanes LI by $\mu^I_{\mathrm{TI}(n),j}(k|k-1)$ in descending order;
for $j = 1$ **to** L **do** `/* traversing all possible lanes for each track */`
  $\hat{I}_n(k) = \mathrm{LI}(j)$;
  $P\{\theta_0(k)\} = P\{\theta_0(k)\}\mu^I_{\mathrm{TI}(n),\mathrm{LI}(j)}(k|k-1)$;
  if $P\{\theta_0(k)\} > \eta P_{max}$ && $(k_h < K || P\{\theta_0(k)\} > P\{\theta_K(k)\})$ **then**
    if $n < N$ **then**
      $n = n + 1$ and **go to** the beginning of **Hypothesis generation**;
    else
      if $k_h < K$ **then**
        $k_h = k + 1, \theta_{k_h}(k) = \theta_0(k)$;
      else
        $\theta_K(k) = \theta_0(k)$;
      end
      update $P_{\max}$ and sort $\theta_i(k), i = 1, ..., k_h$ by $P\{\theta_i(k)\}$ in descending order;
    end
  end
end

the assumption that the lane-changing of a given vehicle does not influence leading vehicles that are making lane-changing decisions ahead of it.

Lane-changing hypothesis evaluation

The longitudinal state estimates for tracks are then predicted based on their respective lane-changing decisions in $\theta_i(k)$. Once new measurements become available, a data association algorithm, e.g., the 2-D assignment algorithm described previously, is applied to each hypothesis and the probability of hypothesis is then evaluated by

$$P\{\theta_i(k)|Z^k\} = \frac{1}{c} p\left[Z(k)|\theta_i(k), Z^{k-1}\right] P\{\theta_i(k)|Z^{k-1}\} \tag{4.47}$$

where c is the normalization factor given by

$$c = \sum_{i=1}^{K} p\left[Z(k)|\theta_i(k), Z^{k-1}\right] P\{\theta_i(k)|Z^{k-1}\}, \tag{4.48}$$

$P\{\theta_i(k)|Z^{k-1}\}$ is $P\{\theta_i(k)\}$ in Algorithm 1 and $p\left[Z(k)|\theta_i(k), Z^{k-1}\right]$ is the likelihood function for $\theta_i(k)$, which can be expressed as

$$p\left[Z(k)|\theta_i(k), Z^{k-1}\right] \propto \prod_{n=1}^{N} (1 - P_D)^{1-\tau_n} \left[\frac{P_D \Lambda^n(k|\theta_i(k))}{\lambda}\right]^{\tau_n} \tag{4.49}$$

In the above, λ is the spatial density of false alarms and τ_n is an indicator as to whether the nth track is associated with a measurement in the data association result under hypothesis $\theta_i(k)$. Also, $\Lambda^n(k|\theta_i(k))$, the likelihood function for the nth track given $\theta_i(k)$, is evaluated by

$$\Lambda^n(k|\theta_i(k)) = \Lambda_l^n(k|\theta_i)\Lambda_d^n(k|\theta_i)$$

$$= \mathcal{N}\left[z_l(k); \hat{z}_l^n(k|k-1,\theta_i), S_l^n(k|\theta_i)\right] \times \mathcal{N}\left[z_d(k); d\left[\hat{I}_n(k|\theta_i(k))\right], \sigma_d^2\right] \quad (4.50)$$

where $\hat{I}_n(k|\theta_i(k))$ and $\hat{z}_l^n(k|k-1,\theta_i)$ with variance $S_l^n(k|\theta_i)$ are the assumed lane-changing decision and the predicted mileage measurement for the nth track given $\theta_i(k)$, respectively.

Adaptive deferred decision logic

The lateral state estimate strongly influences the longitudinal state prediction and thus the performance of the longitudinal motion estimator. Therefore, rather than only retaining the hypothesis with the largest probability in each scan, an adaptive deferred decision logic is proposed: all formed hypotheses are updated and propagated, and the decision-making is deferred into the future in anticipation that subsequent measurements will resolve the uncertainty and thus improve the accuracy of lateral state estimate. Then, each hypothesis will be expanded into new K-best hypotheses in the next scan.

Denote a new hypothesis formed at time step $k+1$ as Θ_i^{k+1}, which is composed of

$$\Theta_i^{k+1} = \left\{\Theta_{p(i)}^k, \theta_{q(i)}(k+1)\right\} \quad (4.51)$$

where $\Theta_{p(i)}^k$ is the parent hypothesis from which Θ_i^{k+1} is derived based on $\theta_{q(i)}(k+1)$, the $q(i)$th hypothesis formed from $\Theta_{p(i)}^k$ using Algorithm 1. The probability of $\Theta_i(k+1)$ is then evaluated by

$$P\{\Theta_i^{k+1}|Z^{k+1}\} = \frac{1}{c} P\{Z(k+1)|\Theta_i^{k+1}, Z^k\} P\{\Theta_i^{k+1}|Z^k\} \quad (4.52)$$

where $P\{Z(k+1)|\Theta_i^{k+1}, Z^k\}$ is the likelihood function defined in (4.49) and

$$P\{\Theta_i^{k+1}|Z^k\} = P\{\theta_{q(i)}(k+1)|\Theta_{p(i)}^k, Z^k\}P\{\Theta_{p(i)}^k|Z^k\} \tag{4.53}$$

with $P\{\theta_{q(i)}(k+1)|\Theta_{p(i)}^k, Z^k\}$ being $P\{\theta_i(k)\}$ in Algorithm 1, derived from $\Theta_{p(i)}^k$ at time step $k+1$. The term $P\{\Theta_{p(i)}^k|Z^k\}$, the probability of parent hypothesis $\Theta_{p(i)}^k$ given Z^k, is available from the previous iteration. The probability of the previously formed hypothesis is then updated as

$$\begin{aligned} P\{\Theta_j^k|Z^{k+1}\} &= \sum_i P\{\Theta_j^k, \Theta_i^{k+1}|Z^{k+1}\} \\ &= \sum_{p(i)=j} P\{\Theta_i^{k+1}|Z^{k+1}\}. \end{aligned} \tag{4.54}$$

Once the updated probability of a hypothesis is greater than the pre-defined threshold P_{th}^I, all other hypotheses generated at the same scan and their descendant hypotheses are pruned. Another threshold η may also be set to prune those hypotheses and their descendant hypotheses if the ratios of their probabilities to that of the best hypothesis generated in the same scan are too small. Other techniques in MHT to further reduce computation cost, such as clustering, can also be used [5].

4.6.3 MTM-Based Multi-vehicle Tracking Algorithm

By combining the proposed MTM-based longitudinal and lateral motion estimators with data association and track maintenance as previously outlined, the MTM-based multi-vehicle tracking algorithm is proposed now. The algorithm's block diagram is presented in Fig. 4.1.

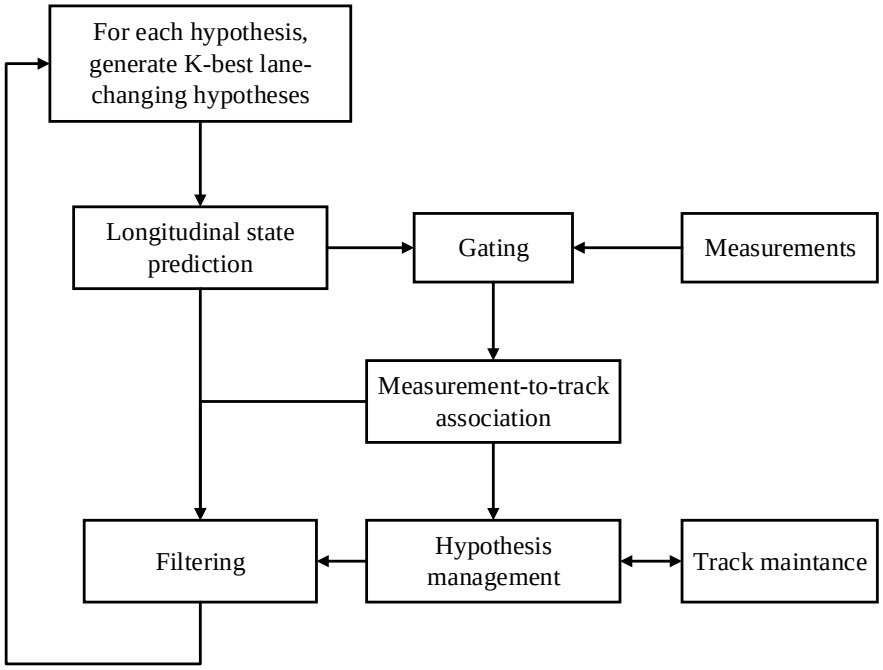


Fig. 4.1: Block diagram of the proposed MTM-based multi-vehicle tracking algorithm.

Since the accuracy of both the longitudinal and lateral state estimates given by the MTM-based tracking algorithm are sensitive to whether there is indeed a leader ahead, in order to improve tracker stability, the MTM-based tracking algorithm is only considered for confirmed tracks: the tentative tracks are still initialized and maintained by the AMELF-based tracking algorithm.

4.7 Simulation Results

In this section, the proposed MTM-based multi-vehicle tracking algorithm is compared with the AMELF-based tracking algorithm using simulations.

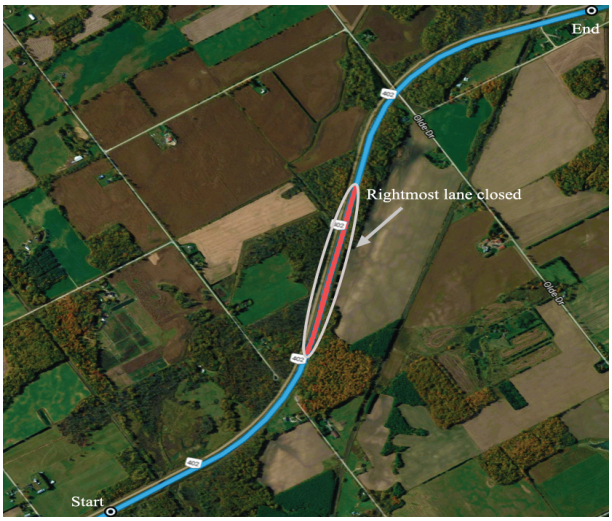


Fig. 4.2: The road map with geographic information.

4.7.1 Simulation Scenario

Six vehicles are simulated to move along a part of Hwy 402 (near London, Ontario, Canada) as shown in Fig. 4.2 with geographic information being obtained from Google Maps. The road has $L = 3$ lanes except for a one-km long stretch marked in red in Fig. 4.2, where the rightmost lane is closed, and each lane has a width of $2\Delta = 4$m. The vehicle motion is generated based on the IDM and MOBIL models with white Gaussian process noise modeling disturbances in driving behaviors. The numerical update step sizes of two models are $\Delta t_{\text{IDM}} = 1$s and $\Delta t_{\text{MOBIL}} = 2$s, respectively. The IDM parameters are set based on Table 4.1 and, since each vehicle may have different desired speeds in practical scenarios, the desired speed of each individual vehicle is assumed normally distributed but non-negative, with the mean being the desired speed of the corresponding vehicle type and the standard deviation $\sigma_{v_0} = 10\,\text{km/h}$. The following MOBIL parameters are used in the simulation: $p = 0.5$, $\Delta a_{\text{th}} = 0.3\,\text{m/s}^2$. The type and initial state of the vehicles are shown in Table 4.2. The

Table 4.2: Types and Initial states of vehicles

Vehicle No.	Type	Mileage coordinate (m)	Lane	Desired speed (km/h)
1	Car	100	2	125
2	Car	150	3	120
3	Car	175	1	110
4	Car	200	2	115
5	Truck	500	2	80
6	Truck	600	1	80

Table 4.3: Lane changes in the simulated scenario

Time (s)	Vehicle No.	Lane change	Time (s)	Vehicle No.	Lane change
14	1	2→3	62	4	1→2
42	1	3→2	70	1	1→2
42	4	2→1	74	2	2→1
48	2	3→2	80	4	2→1
54	1	2→1	96	6	1→2
54	3	1→2			

initial speed of each vehicle is assumed identical to its desired speed. The simulated scenario duration is 100s, in which lane changes occur 8 times in total as shown in detail in Table 4.3. Measurements are generated every $T = 2$s according to (4.5), with $\sigma_l = 10$m and $\sigma_d = 2$m. The target detection probability $P_D = 0.95$, and false alarms are assumed to be uniformly distributed over the whole region with the spatial density $\lambda = 2.0 \times 10^{-5}/\text{m}^2$.

4.7.2 Configuration of Estimators

AMELF-based tracking algorithm

The standard deviation of the process noise for the NCV model is $\sigma_{\text{NCV}} = 0.1\,\text{m/s}^2$. The parameters for the MAA model are set as $\alpha = 1/15$, $a_{\max} = 4\,\text{m/s}^2$, and $a_{\max} = -4\,\text{m/s}^2$. The initial probability of the NCV model and the MAA model are $\mu^l(0) = [0.5, 0.5]^{\text{T}}$, and the corresponding TPM is given by

$$\Pi^l = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}. \tag{4.55}$$

The lane filter (LF) uses $\mu^I(0) = [1/3, 1/3, 1/3]^{\text{T}}$ and due to the assumption that vehicles can only change to one neighboring lane at a time,

$$\Pi^I = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.05 & 0.9 & 0.05 \\ 0 & 0.1 & 0.9 \end{bmatrix} \tag{4.56}$$

is used for tracks whose mileage coordinate estimate is on road segments with 3 lanes and

$$\Pi^I = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.1 & 0.9 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.57}$$

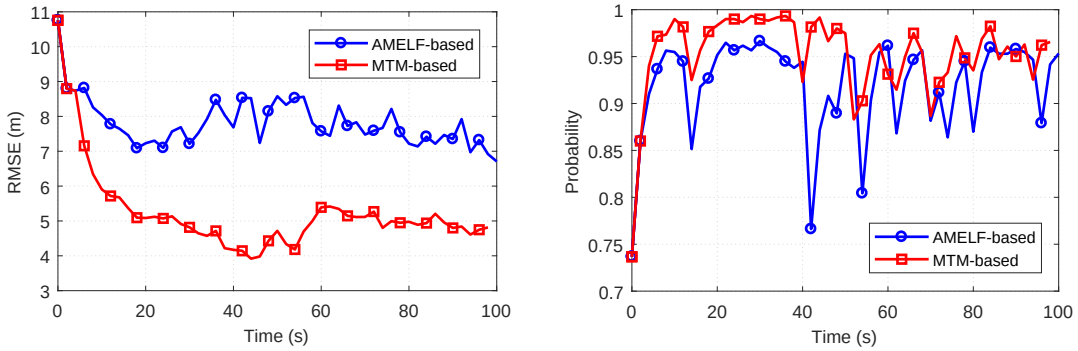
for other tracks on segments with 2 lanes.

MTM-based tracking algorithm

The parameters of UT, $\alpha = 0.5$, $\beta = 2$, and $\kappa = 0$. Each maintained lane-changing hypothesis is expanded into at most $K = 4$ new hypotheses with $\eta = 0.1$ and $P_{\text{th}}^{I} = 0.85$. N-scan pruning [5] with $N = 3$ is also used, combined with the adaptive deferred decision logic to further reduce computation cost.

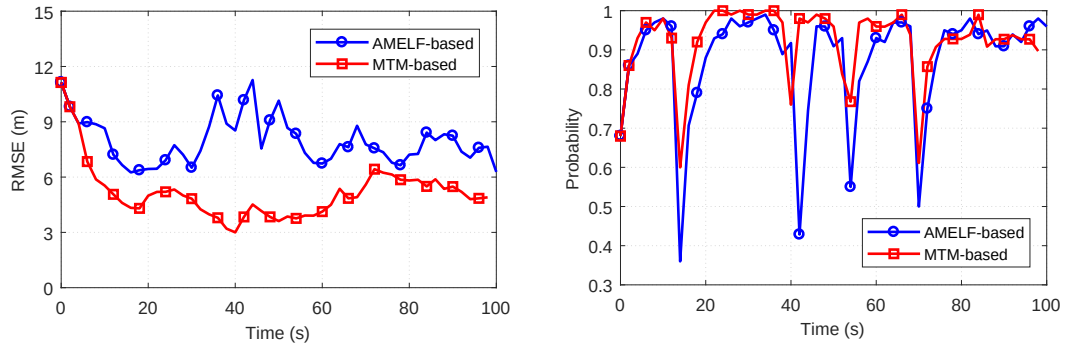
4.7.3 Simulation Results

The performances of two algorithms are evaluated in 100 Monte Carlo runs. The root mean square error (RMSE) of the mileage position estimate is used for measuring the accuracy of the longitudinal state estimator and the probability of the correct lane, defined as the ratio of the number of correct lane identifications to the number of runs in which a confirmed track is associated with this vehicle at time step k, is used for measuring the accuracy of the lateral state estimator. Fig. 4.3 shows the mileage position RMSE and the probability of the correct lane averaged over six vehicles. It can be seen that the MTM-based algorithm yield better performance than the AMELF-based algorithm with respect to both metrics. This is because, with the integration of the IDM and MOBIL models exploiting the motion dependence information, MTM-based estimators can accurately predict longitudinal maneuvers for maintaining a safe distance from the leading vehicle as well as lane changes to improve navigability, and thus provide better prediction. Subsequent measurements are used by the MTM-based algorithm with the adaptive deferred decision logic, with the result that the probability of the correct lane of the MTM-based algorithm is generally better than that of the AMELF-based algorithm. In particular, the valleys in the graphs, corresponding to lane-changing instances summarized in Table 4.3, of



(a) Mileage position RMSE (b) Probability of correct lane

Fig. 4.3: RMSE of mileage coordinate estimates and probability of correct lane averaged over six vehicles.

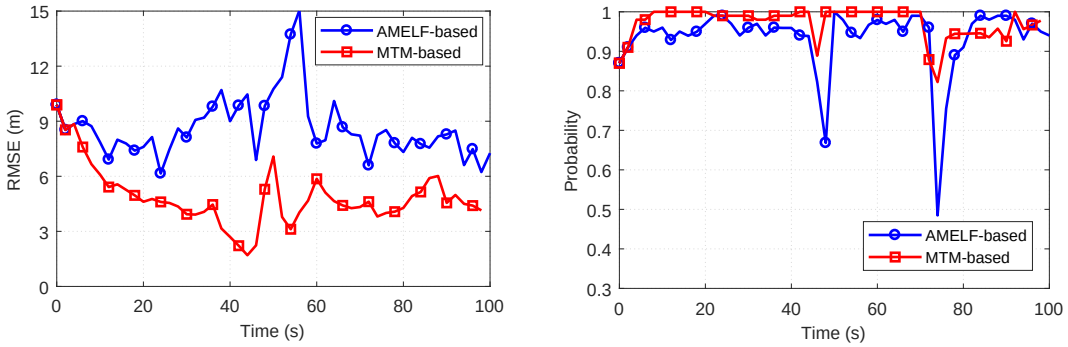


(a) Mileage position RMSE of vehicle 1 (b) Probability of correct lane of vehicle 1

Fig. 4.4: RMSE of mileage coordinate estimates and probabilities of correct lane averaged for vehicle 1.

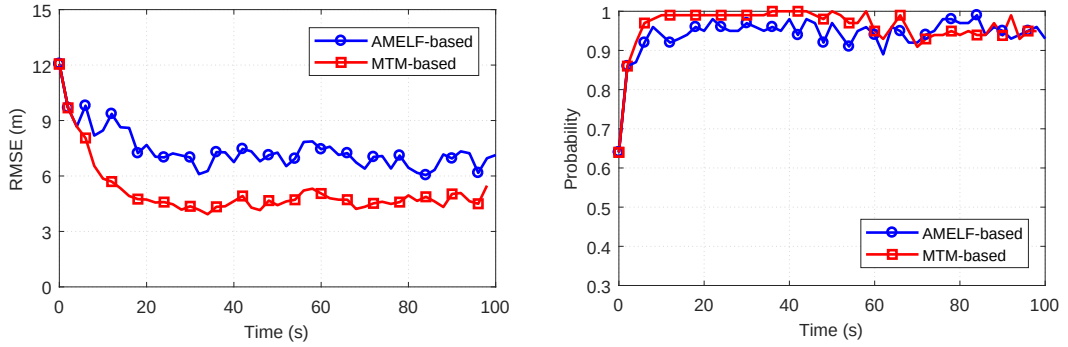
the MTM-based algorithm are much shallower than those of the AMELF-based algorithm. This illustrates that the MTM-based algorithm can make timely predictions of lane changes.

To be more specific, Fig. 4.4,Fig. 4.5 and Fig. 4.6 show the two metrics for vehicles $1, 2$ and 5, respectively. The estimates provided by the MTM-based algorithm for the three vehicles all have better accuracy than those given by the AMELF-based algorithm. Though vehicle 1 exhibits more maneuvers and lane changes than the



(a) Mileage position RMSE of vehicle 2 (b) Probability of correct lane of vehicle 2

Fig. 4.5: RMSE of mileage coordinate estimates and probabilities of correct lane averaged for vehicle 2.



(a) Mileage position RMSE of vehicle 5 (b) Probability of correct lane of vehicle 5

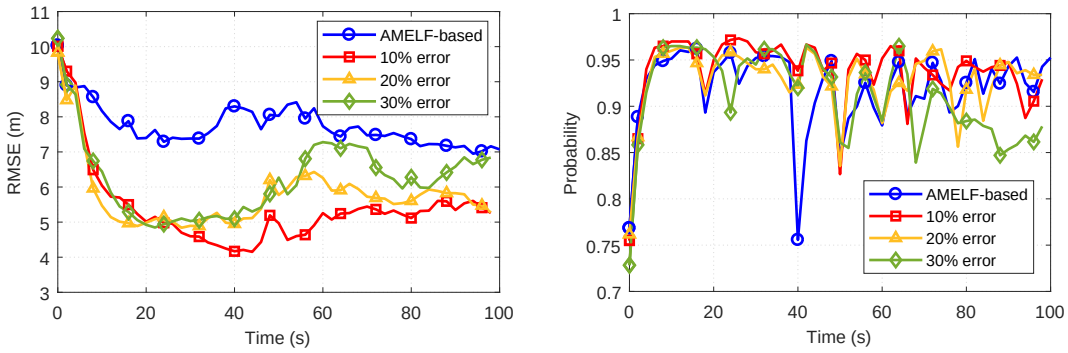
Fig. 4.6: RMSE of mileage coordinate estimates and probabilities of correct lane averaged for vehicle 5.

other vehicles, both algorithms track it successfully but the MTM-based algorithm provides better results. The AMELF-based algorithm gives poor mileage coordinates and lane estimates for vehicle 2 at its first instance of lane-changing. This is because that vehicle is driving to where the road narrows from 3 lanes to 2 lanes and thus has to decelerate in order to merge safely. Although the MAA model is set up to deal with maneuvers, it cannot predict this deceleration in time and thus the mileage position estimate enters the segment with two lanes and forces the lane estimate to

leave the rightmost lane in advance. In contrast, the longitudinal prediction based on the IDM is more accurate, thus this lane merging is also accurately predicted by the MTM-based lateral estimator. Vehicle 5 moves with a nearly constant velocity and no lane changes in the whole duration. The NCV model in the AMELF-based algorithm matches its motion perfectly, but the AMELF-based algorithm still yields a larger mileage position RMSE than the MTM-based algorithm. This is because within the IMM scheme, the MAA model, in this case, has an negative impact on the overall estimate [2].

The robustness of the MTM-based tracking algorithm against mismatch between estimator-used parameters of the MTM and true parameter values is evaluated in the same scenario except that the IDM parameters, T, s_0 and $a_{\max}$, for each vehicle are randomly selected from a uniform distribution centered at the typical values shown in Table 4.1 when generating the ground truth while the typical values are used in the MTM-based tracking algorithm. The length of the uniform distribution determines the maximum relative error of parameters. Fig. 4.7 shows the resulting metrics for the MTM-based algorithm with the maximum relative error of each parameter being 10%, 20% and 30%, respectively, with the comparison of the metrics for the AMELF-based algorithm averaged over these three scenarios. It can be observed that with these three different levels of parameter error, the MTM-based tracking algorithm still tracks all of vehicles successfully. Although its performance declines, it is still better than that of the AMELF-based algorithm.

The computational times of the tracking algorithms are compared on a 2.5 GHz Intel i7 PC with 16 GB RAM. The average computation times per run over 100 Monte Carlo runs are shown in Table 4.4. In comparison with the AMELF-based algorithm,



(a) Mileage position RMSE (b) Probability of correct lane

Fig. 4.7: RMSE of mileage coordinate estimates and probability of correct lane averaged over six vehicles in the scenario with parameter error.

Table 4.4: Average computation times over 100 Monte Carlo runs

	AMELF-based	MTM-based
Average computation time per run (s)	0.57	2.89

the MTM-based algorithm has a higher computational cost (nearly 5 times as much) mainly because multiple lane-changing hypotheses are generated in each scan, and each hypothesis is maintained and expanded to more hypotheses thereafter if the adaptive deferred decision logic is used. However, the higher computational load of the proposed approach does not preclude its application in real systems in view of the computational resources in today's computers. In particular, the proposed algorithm is amenable to parallelization with multicore CPU systems and massively parallel graphical processing units (GPU) [21, 5], which are common in computers today.

4.8 Summary and Conclusions

A new MTM-based tracking algorithm was proposed to track multiple on-road vehicles in the multi-lane case. Unlike existing tracking algorithms in the literature that model motion between vehicles independently and the longitudinal and lateral motion of each vehicle independently, the proposed tracking algorithm considers interactions among vehicles by integrating the IDM car-following model and the MOBIL lane-changing model to model the interactions among vehicles in longitudinal and lateral motions, respectively. The proposed tracking algorithm then estimates the longitudinal and lateral motions sequentially using a recursive approach. A conventional vehicle tracking algorithm in the multi-lane case that does not consider interactions among vehicles was extended to multi-vehicle tracking and was compared with the proposed MTM-based tracking algorithm using simulations. The results indicate that the proposed MTM-based tracking algorithm performs better with respect to both longitudinal and lateral state estimates, albeit at a higher computational cost, due to better prediction of maneuvers the vehicle executes while trying to keep a safe distance from other vehicles and lane changes.

Bibliography

[1] K. I. Ahmed, "Modeling drivers' acceleration and lane changing behavior," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, Feb. 1999.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software.* John Wiley & Sons, 2001.

[3] D. Betaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 786–798, Dec. 2010.

[4] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems.* Boston, MA: Artech House, 1999.

[5] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.

[6] X. Cao, X. Jiang, X. Li, and P. Yan, "Correlation-based tracking of multiple targets with hierarchical layered structure," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 90–102, Jan. 2018.

[7] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[8] Y. Chen, V. P. Jilkov, and X. R. Li, "Multilane-road target tracking using radar and image sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 65–80, Jan. 2015.

[9] B. Ciuffo, V. Punzo, and M. Montanino, "Thirty years of gipps' car-following model," *Transp. Res. Rec.: J. Transp. Res. Bd*, vol. 2315, pp. 89–99, Oct. 2012.

[10] R. Ding, M. Yu, H. Oh, and W. H. Chen, "New multiple-target tracking strategy using domain knowledge and optimization," *IEEE Trans. Syst. Man. Cybern. Syst.*, vol. PP, no. 99, pp. 1–12, 2016.

[11] P. Feng, W. Wang, S. Dlay, S. M. Naqvi, and J. Chambers, "Social force model based MCMC-OCSVM particle PHD filter for multiple human tracking," *IEEE Trans. Multimedia*, vol. PP, no. 99, pp. 1–15, Dec. 2016.

[12] C. Hasberg, S. Hensel, and C. Stiller, "Simultaneous localization and mapping for path-constrained motion," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 541–552, Jun. 2012.

[13] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, May 1995.

[14] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *IEEE Proc.*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[15] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec.*, vol. 1999, pp. 86–94, 2007.

[16] ——, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Phil. Trans. R. Soc. A*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[17] T. Kirubarajan, Y. Bar-Shalom, and K. R. Pattipati, "Ground target tracking with variable structure IMM estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 26–46, Jan. 2000.

[18] G. Kravaritis and B. Mulgrew, "Multitarget ground tracking with road maps and particle filters," in *Proc. IEEE 5th Int. Symp. Signal Process. & Inf. Tech.*, Dec. 2005, pp. 253–257.

[19] X. R. Li and V. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Jul. 2003.

[20] Y. Li and D. Sun, "Microscopic car-following model for the traffic flow: The state of the art," *IET Control Theory Appl.*, vol. 10, no. 2, pp. 133–143, May 2012.

[21] H. Lin, J. Rushing, S. J. Graves, S. Tanner, and E. Criswell, "Real time target tracking with binary sensor networks and parallel computing," in *IEEE Int. Conf. GrC*, May 2006, pp. 112–117.

[22] S. Ossen and S. Hoogendoorn, "Car-following behavior analysis from microscopic trajectory data," *Transp. Res. Rec.: J. Transp. Res. Bd*, vol. 1934, pp. 13–21, Jul. 2005.

[23] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, "Review of microscopic lane-changing models and future research opportunities," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1942–1956, Dec. 2013.

[24] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 246–258, Feb. 2003.

[25] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 198–219, Sep. 2013.

[26] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 1, pp. 128–136, Jan. 2002.

[27] D. Song, R. Tharmarasa, T. Kirubarajan, and X. N. Fernando, "Multi-vehicle tracking with road maps and car-following models," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–12, Aug. 2017.

[28] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin Heidelberg, Springer-Verlag, 2013.

[29] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000.

[30] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[31] A. Ur-Rehman, S. M. Naqvi, L. Mihaylova, and J. A. Chambers, "Multi-target tracking and occlusion handling with learned variational bayesian clusters and a social force model," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1320–1335, Mar. 2016.

[32] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Oct. 2000, pp. 153–158.

[33] C. Yang, M. Bakich, and E. Blasch, "Nonlinear constrained tracking of targets on roads," in *in Proc. 7th Int. Conf. Inf. Fusion*, vol. 1, Jul. 2005, pp. 235–242.

The following chapter is a reproduction of a peer-reviewed article submitted to the IEEE:

Dan Song, Ratnasingham Tharmarasa, Thiagalingam Kirubarajan and Xavier N. Fernando Multi-vehicle Tracking with Microscopic Traffic Flow Model-based Particle Filtering, Submitted to *Automatica* (second round revision), March 2018.

Chapter 5

Multiple Vehicle Tracking on Multi-lane Roads with Particle Filtering

5.1 Abstract

This paper addresses the problem of tracking multiple vehicles on multi-lane roads with consideration for interactions among vehicles. Due to limited lane resources and traffic heterogeneity, vehicles have to interact with neighboring vehicles while moving along roads to improve their navigability and safety, resulting in highly dependent motions. However, multitarget tracking algorithms generally assume that targets move independently of one another. To address this limitation, using the microscopic traffic flow (MTF) model for modeling vehicle dynamics in the presence of interactions with surrounding traffic, an MTF-based tracking algorithm is proposed under the particle filter (PF) framework. The recursive maximum likelihood (RML) method

is integrated into the PF to estimate unknown parameters in the MTF model. The posterior Cramer-Rao lower bound (PCRLB) is also derived for this problem. The performance of the proposed MTF-PF algorithm is compared with those of existing algorithms for multi-vehicle tracking on multi-lane roads. Numerical results show that the proposed algorithm requires less prior information while yielding more accurate and consistent tracks.

5.2 Introduction

Multi-vehicle tracking is a key component in various applications such as ground surveillance [22] and traffic monitoring [3]. Unlike in the case of general multitarget tracking (MTT) [2], the special characteristics of ground vehicle motion can be exploited as prior information to improve tracker performance. The motions of on-road vehicles are subjected to various constraints imposed by roads. Thus, the road-map information can help reduce the uncertainty in vehicle motion estimates and improve track accuracy and consistency. Comprehensive studies have been conducted on road-map aided vehicle tracking in recent years [22, 4].

Due to limited lane resources and traffic heterogeneity, vehicles have to interact with surrounding traffic, which adds another layer of motion constraints. The microscopic traffic flow (MTF) models have been developed in the area of traffic theory to describe the dynamics of an individual vehicle in the presence of interactions with surrounding traffic [21]. Vehicle dynamics can be decomposed into longitudinal dynamics and lateral dynamics, which are described by car-following models [12] and lane-changing models [14], respectively. Recently, the interactions between vehicles have been considered using the MTF models in tracking scenarios and they have led

to significant improvement in tracking performance [15, 17, 18]. The car-following model is used to evaluate the probability of candidate tracks in the multiple hypothesis tracking (MHT)-based multi-vehicle tracking algorithm proposed in [15]. In [18], the problem of tracking multiple vehicles on single-lane roads using the car-following model as motion model for vehicles is studied. This problem has been extended to multi-lane roads by considering the lateral motion of vehicles modeled by the lane changing model [17]. However, several challenges such as data association, online estimation of unknown model parameters and the algorithm stability have not been fully resolved. Also, there is no theoretical bound to evaluate the improvement in tracking performance with consideration for interactions among vehicles.

Therefore, the problem of tracking multiple vehicles on multi-lane roads is revisited in this paper, in which a novel tracking algorithm is proposed and a performance bound for this problem is derived. The intelligent drive model (IDM) [11] and the minimizing overall braking induced by lane change (MOBIL) model [10] are used to model the longitudinal and lateral dynamics of vehicles, respectively, because of their amenability for integration into a probabilistic estimation framework. Due to the high non-linearity of the IDM model and the discrete lateral state, estimating vehicle states is a highly nonlinear and hybrid estimation problem [1]. Thus, due to its ability to handle highly nonlinear and hybrid estimation problems [6, 8], the particle filter (PF) [5] is preferred. However, the main challenge in using the PF for MTT is the curse of dimensionality [23]. Then, it is critical to design an efficient sampling strategy for PF. The main contributions of this paper are: 1) The posterior of vehicle states evolving with the IDM and MOBIL models is derived and a computationally-efficient joint sampling method for PF is proposed; 2) The recursive maximum likelihood (RML)

method [13] is integrated into the PF framework with the particle approximation of the likelihood function to estimate the unknown parameters in the MTF model online; 3) A posterior Cramer-Rao lower bound (PCRLB) [20] quantifying achievable estimation accuracy is derived for this problem.

The rest of the paper is structured as follows: The vehicle dynamics is described in a 2-D road coordinate system and the measurement model is given in Section 5.3. In Section 5.4, the proposed MTF-based PF (MTF-PF) tracking algorithm is described in detail. The PCRLB is derived in Section 5.5. Section 5.6 presents numerical results and comparisons with existing algorithms for tracking multiple vehicles on multi-lane roads. Conclusions are given in Section 5.7.

5.3 Review of Vehicle Kinematics

This paper applies the model proposed in [17], which is briefly summarized in this section, to describe the vehicle kinematics in 2-D road coordinates.

5.3.1 Vehicle Dynamics Model

By modeling roads as a sequence of connected linear segments, a 2-D road coordinate system, which consists of the mileage and displacement coordinates, is defined to represent on-road points [4]. The mileage coordinate of an on-road point is the accumulated arc length from the reference starting point to the projection of the point onto the center line of the road while the displacement coordinate is the signed displacement from its projection to the point.

Decomposing vehicle dynamics into the 2-D road coordinate system, the longitudinal dynamics is the motion of a vehicle driving along the current lane by following

the leading vehicle with a safe distance. The longitudinal dynamics of a vehicle, modeled by the IDM, can be expressed using a discrete-time kinematic equation as follows:

$$x_k = F x_{k-1} + \Gamma \left(a_{k-1} + \nu_{k-1} \right). \tag{5.1}$$

In the above, $x_k \triangleq [s_k, v_k]^{\mathrm{T}}$ is the longitudinal state of the vehicle at time step k, where $(\cdot)^{\mathrm{T}}$ denotes the transpose operator, s_k and v_k are the position and speed of the vehicle in the mileage coordinate, respectively. Also, a_{k-1} is the deterministic acceleration component of the vehicle defined by the IDM model as a function of its motion relative to its leader [11] and it is given by

$$a_k = a_{\max} \left\{ 1 - \left[\frac{v_k}{v_0} \right]^{\delta} - \left[\frac{\Delta s_k^{\star}}{\Delta s_k} \right]^2 \right\}, \tag{5.2}$$

while ν_{k-1} is a zero-mean white Gaussian process noise with variance σ_ν^2 representing the noisy acceleration component to model disturbances of driving behaviors. Thus, $F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ and $\Gamma = [T^2/2, T]^{\mathrm{T}}$, where T denotes the update interval for the IDM model. The definition of the variables in (5.2) can be found in [11, 17]. The component in the displacement coordinate is referred to as lateral dynamics by a vehicle performing lane changes to improve its navigability. By assuming that lane changes take place instantaneously, the lateral dynamics of a vehicle is modeled by the MOBIL lane-changing model [10] and it can be expressed as

$$l_k = l_{k-1} + \psi_{k-1} \tag{5.3}$$

where l_k is the lateral state of the vehicle defined as the lane (from left to right) in

which it is currently traveling and ψ_{k-1} denotes the lane-changing decision made by the vehicle at time step $k-1$. By assuming that vehicles can only change one lane at a time, $\psi_{k-1,j}$ takes values in $\{-1, 0, 1\}$, corresponding to left lane-changing, no change and right lane-changing, respectively, which is determined by the incentive criterion and the safety criterion in the MOBIL model. For more details on the MOBIL model, the reader is referred to [10, 17]. Note that the MOBIL model does not consider random lane changes made by drivers. It can be extended by assigning a small probability to perform those lane changes that meet the safety criterion while not meeting the incentive criterion.

5.3.2 Measurement Model

A simplified model for position-only measurement [17] is assumed in this paper. Due to the limited accuracy of the sensor, the resulting measurements are contaminated with errors, missed detections and false alarms. False alarms are assumed to be uniformly distributed over the whole region with spatial density λ and their cardinality follow the Poisson distribution [2]. Vehicles are detected with detection probability P_D in each scan. A measurement originating from a vehicle with states x_k and l_k can be expressed as

$$z_k \triangleq [z_{s_k}, z_{d_k}]^{\mathrm{T}} = [Hx_k, d(l_k)]^{\mathrm{T}} + [w_{s_k}, w_{d_k}]^{\mathrm{T}} \tag{5.4}$$

where w_{s_k} and w_{d_k} are zero-mean Gaussian noises with variances σ_s^2 and σ_d^2 that perturb z_{s_k} and z_{d_k}, respectively. In the above, $H = [1\ 0]^{\mathrm{T}}$ and $d(l_k)$ denotes the displacement coordinate of the vehicle given by $d(l_k) = (2l_k - L - 1)\Delta$, where L is the number of lanes in a road segment, each with width 2Δ. The set of all measurements

received at time step k is denoted as $Z_k = \{z_{k,i},\ i = 1, ..., M_k\}$ and the cumulative set of Z_k available up to time step k is denoted as $Z^k = \{Z_i,\ i = 1, ..., k\}$. Note that the update rates of the IDM and MOBIL models may be different from each other and from that of measurements. Although the same time step symbol k is used in (5.1), (5.3) and (5.4), they need to be clarified further later.

5.4 MTF-based Particle Filter

Considering the presence of r vehicles, the stacked longitudinal and lateral states, constructed as the concatenation of the longitudinal state and the lateral state of these vehicles, i.e., $X_k = [x_{k,1}^{\mathrm{T}}, ..., x_{k,r}^{\mathrm{T}}]^{\mathrm{T}}$ and $L_k = [l_{k,1}, ..., l_{k,r}]^{\mathrm{T}}$, respectively, need to be inferred at each time step. Due to the heterogeneity in realistic traffic, some parameters in the IDM and the MOBIL models may be driver-dependent and unknown to the tracker. It is necessary to estimate them online during tracking. Without loss of generality and for simplicity, in this paper, we take the desired speed as an unknown parameter and assume that other parameters take the typical values given in [11, 17]. Stacking the desired speed of each vehicle, the parameter vector, $u = [v_{0,1}, ..., v_{0,r}]$, is estimated together with X_k and L_k.

5.4.1 Multitarget Particle Filter

The auxiliary PF [5] is applied to approximate the Bayes formula [2] to recursively update the posteriors of X_k and L_k given Z^k. A more detailed overview of the auxiliary PF can be found in [5]. Since the measurement-origin uncertainty has to be resolved in MTT, the measurement-to-track association hypothesis vector, $\theta =$

$[\theta_1, ..., \theta_r]$, that indicates which measurement each target contributes to, is introduced to marginalize the measurement density and jointly sampled with the auxiliary index i', X_k and L_k from an importance density $q(X_k, L_k, i', \theta|Z^k)$. The main challenge here is to find an importance density to efficiently handle the curse of dimensionality [23].

Design of Importance Density

An efficient importance density is designed by factorizing

$$q(X_k, L_k, i', \theta|Z^k) = p(\theta|Z^k)p(i'|\theta, Z^k)p(X_k, L_k|\theta, i', Z^k). \tag{5.5}$$

In the above, $p(\theta|Z^k)$ is the posterior probability mass of the association hypothesis vector θ given by

$$p(\theta|Z^k) \propto p(\theta|Z^{k-1})p_u(Z_k|\theta, Z^{k-1}) \tag{5.6}$$

where $p(\theta|Z^{k-1})$ is the prior of θ given by [23]

$$p(\theta|Z^{k-1}) \propto \left[\frac{P_D}{(1-P_D)\lambda}\right]^{r(\theta)} \tag{5.7}$$

with $r(\theta)$ being the number of targets associated with measurement under hypothesis θ. In (5.6), $p_u(Z_k|\theta, Z^{k-1})$, where the subscript u represents the probability given the parameter vector u, i.e., $p_u(Z_k|\theta, Z^{k-1}) = p(Z_k|\theta, u, Z^{k-1})$, can be approximated by

$$p_u(Z_k|\theta, Z^{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^i p_u(Z_k|i, \theta, Z^{k-1}) \tag{5.8}$$

with

$$p_u(Z_k|i,\theta,Z^{k-1}) = \prod_{\{j:\theta_j>0\}} p_u(z_{k,\theta_j}|X^i_{k-1},L^i_{k-1}) = \prod_{\{j:\theta_j>0\}} \mathcal{N}\left(z_{k,\theta_j};\hat{z}^i_{k,j},S\right) \tag{5.9}$$

given the particle set $\left\{X^i_{k-1},L^i_{k-1},w^i_{k-1}\right\}_{i=1}^N$ in the previous time step. In the above, $\mathcal{N}(z_{k,\theta_j};\hat{z}^i_{k,j},S)$ denotes the Gaussian probability density function evaluated at z_{k,θ_j} with mean vector $\hat{z}^i_{k,j}$ and covariance matrix S, where $\hat{z}^i_{k,j} = [H\hat{x}^i_{k,j}, d(\hat{l}^i_{k,j})]^{\mathrm{T}}$. Also, $\hat{x}^i_{k,j}$ and $\hat{l}^i_{k,j}$ are the predicted longitudinal and lateral states of the jth vehicle at time step k given the ith particle and the parameter vector u based on (5.1) and (5.3), respectively. The covariance matrix S is given by $S = \mathrm{diag}(S_s,\sigma_d^2)$, where $S_s = HQH^{\mathrm{T}} + \sigma_s^2$ and $Q = \Gamma\sigma_\nu^2\Gamma^{\mathrm{T}}$. In (5.5), $p(i'|\theta,Z^k)$ is the probability for the i'th particle propagating from time step $k-1$ to k given by

$$p(i'|\theta,Z^k) \propto p(i'|\theta,Z^{k-1})p_u(Z_k|i',\theta,Z^{k-1}) \tag{5.10}$$

where $p(i'|\theta,Z^{k-1}) = w^{i'}_{k-1}$ and $p_u(Z_k|i',\theta,Z^{k-1})$ is given in (5.9). Finally, X_k and L_k are sampled by the optimal importance densitys (OID) [24], which is given by

$$p(X_k,L_k|\theta,i',Z^k) = p_u(L_k|\theta,i',Z^k)p_u(X_k|L_k,\theta,i',Z^k). \tag{5.11}$$

From the lateral state equation (5.3),

$$p_u(L_k|\theta,i',Z^k) = p_u(L_k|X^{i'}_{k-1},L^{i'}_{k-1}) = \delta(L_k - \hat{L}^{i'}_k). \tag{5.12}$$

where $\hat{L}_k^{i'} = [\hat{l}_{k,1}^{i'}, ..., \hat{l}_{k,r}^{i'}]^{\mathrm{T}}$. $p_u(X_k|L_k, \theta, i', Z^k)$ can be written as

$$\begin{aligned} p_u(X_k|L_k, \theta, i', Z^k) &= p_u(X_k|X_{k-1}^{i'}, L_k, \theta, Z_k) \\ &= \prod_{j=1}^{r} p_u(x_{k,j}|X_{k-1}^{i'}, L_k, z_{k,\theta_j}) \end{aligned} \tag{5.13}$$

$$p_u(x_{k,j}|X_{k-1}^{i'}, L_k, z_{k,\theta_j}) = \begin{cases} \mathcal{N}\left(x_{k,j}; \tilde{x}_{k,j}^{i'}, \Sigma\right), \theta_j > 0 \\ \mathcal{N}\left(x_{k,j}; \hat{x}_{k,j}^{i'}, Q\right), \theta_j = 0 \end{cases} \tag{5.14}$$

Here, $\tilde{x}_{k,j}^{i'} = \hat{x}_{k,j}^{i'} + W(z_{s_{k,\theta_j}} - H\hat{x}_{k,j}^{i'})$ and $\Sigma = Q - WS_sW^{\mathrm{T}}$, where $W = QH^{\mathrm{T}}S_s^{-1}$.

Sampling of Data Association Hypothesis and Weight Calculation

The number of feasible association hypothesis vectors θ exponentially increases with the number of targets and measurements. To reduce the computation load, only the hypothesis vectors with large probabilities are selected as candidates for sampling. However, the evaluation of $p(\theta|Z^k)$ given in (5.6)–(5.9) is still not computationally efficient. Hence, by using an alternative approximation of $p_u(Z_k|\theta, Z^{k-1})$,

$$\tilde{p}_u(Z_k|\theta, Z^{k-1}) = \prod_{\{j:\theta_j>0\}} \sum_{i=1}^{N} w_{k-1}^i p_u(z_{k,\theta_j}|X_{k-1}^i, L_{k-1}^i), \tag{5.15}$$

$p(\theta|Z^k)$ can be refactorized as

$$p(\theta|Z^k) \propto \prod_{\{j:\theta_j>0\}} \frac{P_D}{(1-P_D)\lambda} p_u(z_{k,\theta_j}|Z^{k-1}). \tag{5.16}$$

Define a cost for associating the nth track with the mth measurement as $c(k, m, n) = -\ln \Lambda(k, m, n)$ and a cost function for each hypothesis vector as $C(k, \theta) = \sum_{j=1}^{r} c(k, \theta_j, j)$,

where $\Lambda(k,m,n)$ is the likelihood function given by

$$\Lambda(k,m,n) = \begin{cases} \frac{P_D}{\lambda} \sum_{i=1}^{N} w_{k-1}^{i} \mathcal{N}\left(z_{k,m}; \hat{z}_{k,n}^{i}, S\right), m > 0 \\ 1 - P_D, \qquad m = 0. \end{cases} \tag{5.17}$$

It can be verified that $p(\theta|Z^k) \propto \exp[-C(k,\theta)]$. Then, the K-best Auction algorithm [2] can be used to efficiently select the hypothesis vectors with the top K largest probabilities. Denoting the resulting hypothesis set as $\Theta_k = \{\theta_1^*, ..., \theta_K^*\}$, the hypothesis vector θ is sampled from Θ_k with the normalized probability

$$p(\theta_i^*|Z^k) = \frac{\exp[-C(k,\theta_i^*)]}{\sum_{j=1}^{K} \exp[-C(k,\theta_j^*)]}. \tag{5.18}$$

After new particles are sampled from $q(X_k, L_k, i', \theta|Z^k)$, based on (5.5) and (5.16), their weights are calculated by

$$w_k^i \propto \frac{p(X_k^i, L_k^i, i'^i, \theta^i|Z^k)}{q(X_k^i, L_k^i, i'^i, \theta^i|Z^k)} = \frac{p_u(Z_k|\theta^i, Z^{k-1})}{\tilde{p}_u(Z_k|\theta^i, Z^{k-1})} \tag{5.19}$$

where the superscript i denotes the weight and samples for the ith particle. Finally, the estimates of X_k and L_k can be obtained by $\hat{X}_k = \sum_{i=1}^{N} w_k^i X_k^i$ and $\hat{L}_k = [\hat{l}_{k,1}, ..., \hat{l}_{k,r}]^{\mathrm{T}}$, where $\hat{l}_{k,j} = \arg\max_{1 \le l \le L} \sum_{\{i: l_{k,j}^i = l\}} w_k^i$.

5.4.2 RML Based on Particle Approximations

The MTF-based PF is derived given the parameter vector u. However, u is unknown and thus needs to be estimated online. A straightforward approach is to augment the state-space with u and estimate it together with X_k and L_k in the PF.

However, this method may yield poor performance even in some simple scenarios [9]. Thus, the RML method is integrated into the PF to estimate u separately. When $p_u(Z^k)$ is differentiable with respect to u, the RML estimate of u can be recursively updated using the steepest ascent algorithm as

$$\hat{u}_k = \hat{u}_{k-1} + \gamma_k \nabla_u \log p_{\hat{u}_{k-1}}(Z^k) \tag{5.20}$$

where ∇_Θ denotes the gradient with respect to Θ and γ_k is the step size with the constraints $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$ to ensure the convergence of the algorithm [13]. In our case, there is no closed-form expression for $\nabla_u \log p_{\hat{u}_{k-1}}(Z^k)$ and thus its particle-based approximation is derived. However, it is still computationally expensive to evaluate the particle approximations of $\nabla_u \log p_{\hat{u}_{k-1}}(Z^k)$ because the entire history of observations has to be revisited. To overcome this problem, $\hat{u}_k$ is updated in the direction of $\nabla_u \log p_{\hat{u}_{k-1}}(Z_k|Z^{k-1})$ instead of $\nabla_u \log p_{\hat{u}_{k-1}}(Z^k)$. Define $\alpha_k^i \triangleq \nabla_u \log p_{\hat{u}_{k-1}}(X_k^i, Z^k)$ for each particle. Instead of a single value, each α_k^i is represented by a Gaussian kernel with mean m_k^i and covariance $h^2 V_k$ to move α_k^i to their mean while adding a noise to overcome particle degeneracy [13]. Then, $\nabla_u \log p_{\hat{u}_{k-1}}(Z^k) \approx \sum_{i=1}^N w_k^i m_k^i$ and

$$\begin{aligned} \nabla_u \log p_{\hat{u}_{k-1}}(Z_k|Z^{k-1}) &\approx \nabla_u \log p_{\hat{u}_{k-1}}(Z^k) - \nabla_u \log p_{\hat{u}_{k-2}}(Z^{k-1}) \\ &= \sum_{i=1}^N w_k^i m_k^i - \sum_{i=1}^N w_{k-1}^i m_{k-1}^i. \end{aligned} \tag{5.21}$$

Thus, only m_k^i needs to be maintained and propagated by

$$m_k^i = \eta m_{k-1}^{i'^i} + (1-\eta) \sum_{j=1}^N w_{k-1}^j m_{k-1}^j + \nabla_u \log p_{\hat{u}_{k-1}}(Z_k, X_k^i | X_{k-1}^{i'^i}) \tag{5.22}$$

where η is the shrinkage parameter and should satisfy the condition that $\eta^2 + h^2 = 1$ [13]. Since the measurement Z_k is conditionally-independent of u given X_k,

$$\nabla_u \log p_{\hat{u}_{k-1}}(Z_k, X_k^i | X_{k-1}^{i'^i}) = \nabla_u \log p_{\hat{u}_{k-1}}(X_k^i | X_{k-1}^{i'^i}). \tag{5.23}$$

Based on (5.1), we have

$$\begin{aligned} \nabla_u \log p_{\hat{u}_{k-1}}(X_k^i | X_{k-1}^{i'^i}) &= \sum_{j=1}^{r} \nabla_u \log p_{\hat{u}_{k-1}}(x_{k,j}^i | X_{k-1}^{i'^i}) \\ &= \left[\nabla_{v_0} \log p_{\hat{u}_{k-1,1}}(x_{k,1}^i | X_{k-1}^{i'^i})^{\mathrm{T}}, ..., \nabla_{v_0} \log p_{\hat{u}_{k-1,r}}(x_{k,r}^i | X_{k-1}^{i'^i})^{\mathrm{T}} \right]^{\mathrm{T}} \end{aligned} \tag{5.24}$$

where $\nabla_{v_0} \log p_{\hat{u}_{k-1,j}}(x_{k,j}^i | X_{k-1}^{i'^i}) = \nabla_{v_0} \hat{x}_{k-1,j}^{i'^i} Q^{-1}(x_{k,j}^i - \hat{x}_{k-1,j}^{i'^i})$. Here, $\nabla_{v_0} \hat{x}_{k-1,j}^{i'^i}$ denotes the derivative of $\hat{x}_{k-1,j}^{i'^i}$ with respect to v_0 evaluated at $\hat{u}_{k-1,j}$. Based on (5.2), it is given by

$$\nabla_{v_0} \hat{x}_{k-1,j}^{i'^i} = \sum_{\kappa=1}^{n} F^{\kappa-1} \Gamma \nabla_{v_0} a_{k-1,j}^{i'^i,\kappa} \tag{5.25}$$

$$\nabla_{v_0} a_{k-1,j}^{i'^i,\kappa} = \frac{a_{\max}\delta}{v_0} \left(v_{k-1,j}^{i'^i,\kappa} v_0 \right)^{\delta} \Big|_{v_0 = \hat{u}_{k-1,j}} \tag{5.26}$$

where n is the number of times the longitudinal state evolves between time step $k-1$ and k and $v_{k-1,j}^{i'^i,\kappa}$ denotes the speed of the jth vehicle before the κth evolution. Note that Q becomes singular when $n = 1$. For this case, $p_{\hat{u}_{k-1,j}}(x_{k,j}^i | X_{k-1}^{i'^i})$ collapses to $\mathcal{N}(s_{k,j}^i; \hat{s}_{k-1,j}^{i'^i}, Q_{11})$ and thus $\nabla_{v_0} \log p_{\hat{u}_{k-1,j}}(x_{k,j}^i | X_{k-1}^{i'^i})$ can be evaluated as

$$\nabla_{v_0} \log p_{\hat{u}_{k-1,j}}(x_{k,j}^i | X_{k-1}^{i'^i}) = \frac{\nabla_{v_0} \hat{s}_{k-1,j}^{i'^i} (s_{k,j}^i - \hat{s}_{k-1,j}^{i'^i})}{Q_{11}} \tag{5.27}$$

Algorithm 2: The proposed MTF-based particle filter

Assume that the estimate of u, $\hat{u}_{k-1}$, and the particle set $\{X_{k-1}^i, L_{k-1}^i, m_{k-1}^i, w_{k-1}^i\}_{i=1}^N$ are available at time step $k-1$.

Sequential importance resampling:

- Predict $\hat{X}_k^i$ and $\hat{L}_k^i$ using (5.1) and (5.3) given X_{k-1}^i, L_{k-1}^i and $\hat{u}_{k-1}$.
- Generate the candidate association hypothesis vector set Θ_k using the K-best Auction algorithm and calculate $p(\theta^*|Z^k)$ based on (5.18).
- Sample the hypothesis vector θ_i from Θ_k with $p(\theta^*|Z^k)$.
- Sample the auxiliary index i'^i with probability $p(i'|\theta_i, Z^k)$ calculated by (5.10).
- Propose L_k^i and X_k^i according to (5.12) and (5.13), respectively, given θ_i and i'^i.
- Calculate the weights w_k^i using (5.19).
- Update $\hat{X}_k$ and $\hat{L}_k$.
- Resample the particle set if the effective sample size is below a pre-defined threshold.

Recursive maximum likelihood estimation:

- Calculate $\nabla_u \log p_{\hat{u}_{k-1}}(Z_k, X_k^i | X_{k-1}^{i'})$ based on (5.24)–(5.26) and update m_k^i according to (5.22).
- Calculate $\nabla_u \log p_{\hat{u}_{k-1}}(Z_k|Z^{k-1})$ using (5.21) and update the estimate $\hat{u}_k = \hat{u}_{k-1} + \gamma_k \nabla_u \log p_{\hat{u}_{k-1}}(Z_k|Z^{k-1})$.

where $\nabla_{v_0} \hat{s}_{k-1,j}^{i'^i} = \Gamma_{11} \nabla_{v_0} a_{k-1,j}^{i'^i}$, Γ_{11} and Q_{11} denote the first element of Γ and Q, respectively.

5.4.3 Algorithm Summary

Algorithm 2 summarizes the proposed MTF-based PF, which is derived given the number of vehicles. In order to handle an unknown or a time-varying number of vehicles, a track management method, in which each measurement that is not associated with any tracks in candidate association hypothesis vectors in Θ_k initializes

a tentative track and an M-out-of-N logic [2] is applied to confirm tentative tracks and terminate dead ones, is integrated with the MTF-PF. Since the MTF-based vehicle dynamics model depends on surrounding traffic, the proposed MTF-PF is effective provided vehicles within a short range are detected with high probability and to improve tracker stability, the MTF-PF is only considered for confirmed tracks while tentative tracks are separately maintained by the conventional vehicle tracking algorithm in [4]. Once a tentative track is confirmed, N samples of its longitudinal and lateral states, $\{x_k^i, l_k^i\}_{i=1}^N$, are drawn from their posteriors maintained by the conventional algorithm and concatenated into the particles, i.e., $X_k^i = [{X_k^i}^{\mathrm{T}}, {x_k^i}^{\mathrm{T}}]^{\mathrm{T}}$ and $L_k^i = [{L_k^i}^{\mathrm{T}}, l_k^i]^{\mathrm{T}}$. The desired speed estimate that can be initialized by its speed estimate is concatenated into $\hat{u}_k$ and m_k^i is augmented as $m_k^i = [{m_k^i}^{\mathrm{T}}, 0]^{\mathrm{T}}$. Once a confirmed track is terminated, its corresponding parts in $\{X_{k-1}^i, L_{k-1}^i, m_{k-1}^i\}_{i=1}^N$ and $\hat{u}_k$ are removed.

5.5 PCRLB for Multi-Vehicle Tracking

The PCRLB is a commonly used performance bound to quantify the best possible accuracy due to its computationally efficient formulation for recursive evaluation [20]. However, the regularity conditions required by the PCRLB [16] are not satisfied in the combined continuous and discrete state estimation problem. One trade-off is to assume that the lateral dynamics of vehicles is known and derive the PCRLB for the longitudinal dynamics. Although this would result in an optimistic bound, it is still acceptable as long as the lateral estimate from the algorithm is reasonable.

The PCRLB is defined by the inverse of the Fisher information matrix (FIM), J_k^{-1}, which provides a lower bound on the error covariance matrix for any unbiased

estimates. The FIM J_k can be recursively evaluated as

$$J_k = D_{k-1}^{22} - D_{k-1}^{21}(J_{k-1} + D_{k-1}^{11})^{-1} D_{k-1}^{12} + J_{Z_k} \tag{5.28}$$

The expressions of the terms in (5.28) can be found in [20]. In our problem, the PCRLB is derived for longitudinal state x_k and unknown desired speed v_0 of vehicles. Thus, denoting $\bar{X}_k = [\bar{x}_{k,1}^{\mathrm{T}}, ..., \bar{x}_{k,r}^{\mathrm{T}}]^{\mathrm{T}}$, where $\bar{x}_{k,j} = [x_{k,j}^{\mathrm{T}}, v_{0,j}]^{\mathrm{T}}$, the evolution of $\bar{X}_k$ can be written as

$$\bar{X}_k = f(\bar{X}_{k-1}, L_{k-1}) + \nu_{k-1} \tag{5.29}$$

where $f = [f_1, ...f_r]^{\mathrm{T}}$ and $\nu_{k-1} = [\bar{\Gamma}^{\mathrm{T}}\nu_{k-1,1}, ..., \bar{\Gamma}^{\mathrm{T}}\nu_{k-1,r}]^{\mathrm{T}}$. Here, f_j denotes the function for $\bar{x}_{k,j}$, which, based on (5.1), is given by $f_j(\bar{X}_{k-1}, L_{k-1}) = \bar{F}\bar{x}_{k-1,j} + \bar{\Gamma}a_{k-1,j}$, where $\bar{F} = \mathrm{diag}(F, 1)$ and $\bar{\Gamma} = [\Gamma^{\mathrm{T}}, 0]^{\mathrm{T}}$. Denoting the covariance matrix of ν_{k-1} as $\bar{Q}$, the terms in (5.28) are given by

$$D_{k-1}^{11} = \mathcal{F}^{\mathrm{T}}\bar{Q}^{-1}\mathcal{F} \tag{5.30}$$

$$D_{k-1}^{12} = {D_{k-1}^{21}}^{\mathrm{T}} = -\mathcal{F}^{\mathrm{T}}\bar{Q}^{-1} \tag{5.31}$$

$$D_{k-1}^{22} = \bar{Q}^{-1} \tag{5.32}$$

where $\mathcal{F} = \nabla_{\bar{X}_{k-1}}^{\mathrm{T}} f(\bar{X}_{k-1}, L_{k-1})$ with the following form

$$\mathcal{F} = \begin{bmatrix} \bar{F} + \bar{\Gamma}\nabla_{\bar{x}_1} a_{k-1,1} & \bar{\Gamma}\nabla_{\bar{x}_2} a_{k-1,1} & \dots & \bar{\Gamma}\nabla_{\bar{x}_r} a_{k-1,1} \\ \bar{\Gamma}\nabla_{\bar{x}_1} a_{k-1,2} & \bar{F} + \bar{\Gamma}\nabla_{\bar{x}_2} a_{k-1,2} & \dots & \bar{\Gamma}\nabla_{\bar{x}_r} a_{k-1,2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\Gamma}\nabla_{\bar{x}_1} a_{k-1,r} & \bar{\Gamma}\nabla_{\bar{x}_2} a_{k-1,r} & \dots & \bar{F} + \bar{\Gamma}\nabla_{\bar{x}_r} a_{k-1,r} \end{bmatrix} \tag{5.33}$$

and based on (5.2), $\nabla_{\bar{x}_j} a_{k-1,i}$ is given by

$$\nabla_{\bar{x}_j} a_{k-1,i} = \begin{cases} \begin{bmatrix} -\frac{2a_{\max}}{s_{k-1,j}} \left(\frac{\Delta s^*_{k-1,i}}{\Delta s_{k-1,i}}\right)^2 \\ -a_{\max}\left[\frac{\delta}{v_{0,i}}\left(\frac{v_{k-1,j}}{v_{0,i}}\right)^{\delta-1} + \tau(\Delta s^*_{k-1,i})\frac{2\Delta s^*_{k-1,i}}{\Delta s^2_{k-1,i}}(T + \frac{v_{k-1,j}+2\Delta v_{k-1,i}}{2\sqrt{ab}})\right] \\ \frac{a_{\max}\delta}{v_{0,j}}\left(\frac{v_{k-1,j}}{v_{0,i}}\right)^{\delta} \end{bmatrix}, & j = i \\ \begin{bmatrix} \frac{2a_{\max}}{s_{k-1,i}}\left(\frac{\Delta s^*_{k-1,i}}{\Delta s_{k-1,i}}\right)^2 \\ -a_{\max}\tau(\Delta s^*_{k-1,i})\frac{2\Delta s^*_{k-1,i}}{\Delta s^2_{k-1,i}}\frac{v_{k-1,i}}{2\sqrt{ab}} \\ 0 \end{bmatrix}, & j = i^l_{k-1} \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{5.34}$$

where i^l_{k-1} denotes the leading vehicle of the ith vehicle at time step $k-1$ and $\tau(\Delta s^*)$ is an indicator that is equal to 0 when $\Delta s^* = \Delta s_0$ otherwise is 1. Substituting (5.30)–(5.32) into (5.28), we have

$$J_k = (\mathcal{F} J_{k-1} \mathcal{F}^{\mathrm{T}} + \bar{Q})^{-1} + J_{Z_k}. \tag{5.35}$$

Note that though $\bar{Q}$ in our problem is singular, it can be verified that (5.35) yields the same result with the method proposed for the singular case in [20]. The derivation of J_{Z_k} in our problem is identical to the general MTT problems. In this paper, J_{Z_k} is evaluated using the method in [19].

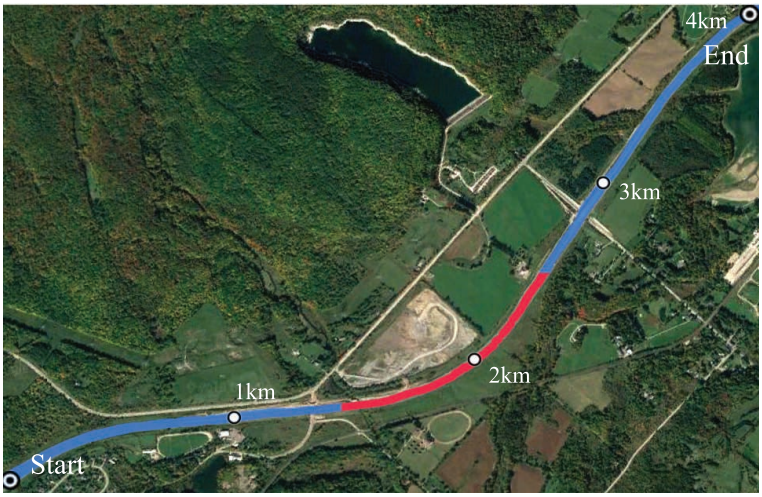


Fig. 5.1: The road map with geographic information being extracted from the Google Maps.

5.6 Simulation Results

The performance of the proposed MTF-PF tracking algorithm is evaluated and compared with those of the algorithms proposed in [4] and [17] through simulations. To the best of our knowledge, these are the only two algorithms for tracking multiple vehicles on multi-lane roads reported so far. The algorithm presented in [4], referred to as AMELF algorithm, assumes that the motions of vehicles are mutually independent. The other algorithm presented in [17], referred to as MTF-UKF algorithm, uses the same MTF models while employing the unscented Kalman filter (UKF) under the MHT framework to track vehicles.

5.6.1 Traffic Scenario

A scenario consisting of six vehicles over 100s is simulated on a 4 km long part of Highway 401 near Milton, Ontario, Canada, as shown in Fig. 5.1. This highway has

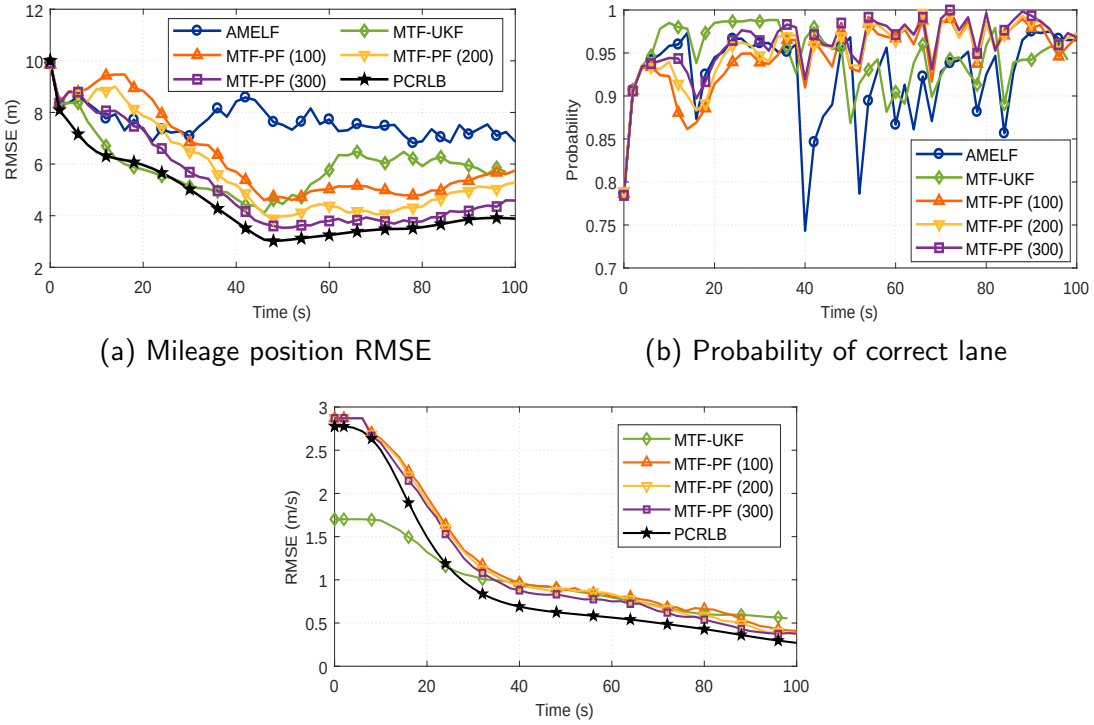
Table 5.1: Types and initial states of vehicles

Vehicle No.	Type	Mileage (m)	Lane	Desired speed (km/h)
1	Car	0	2	125
2	Car	50	3	120
3	Car	100	1	110
4	Car	100	2	115
5	Truck	400	2	80
6	Truck	500	1	80

Table 5.2: Lane changes in the simulated scenario

Time (s)	Vehicle No.	Lane change	Time (s)	Vehicle No.	Lane change
16	1	2→3	60	4	1→2
40	1	3→2	64	2	2→1
40	4	2→1	68	1	1→2
46	2	3→2	78	4	2→1
52	1	2→1	84	2	1→2
52	3	1→2			

$L = 3$ lanes while the far right lane is closed on a 1 km long stretch starting at 1.43 km away from the reference starting point of the road, which is marked by red in Fig. 5.1. The width of each lane is $2\Delta = 4$m. The motions of vehicles are generated based on state equations (5.1) and (5.3) with $\sigma_\nu = 0.1$ m/s^2 and initial states as shown in Table 5.1. The numerical update step sizes of the two equations are $T_{\text{lon}} = 1$s and $T_{\text{lat}} = 2$s, respectively. The IDM and MOBIL parameters are set according to [17]. The lane changing occurs 11 times in the simulation and the details are shown in Table 5.2. The measurements are generated every $T = 2$s with spatial false alarm density $\lambda = 2.0 \times 10^{-5}/\text{m}^2$, detection probability $P_D = 0.95$, $\sigma_s = 10$m and $\sigma_d = 2$m.



(a) Mileage position RMSE

(b) Probability of correct lane

(c) Desired speed RMSE

Fig. 5.2: RMSEs of mileage position and desired speed estimates and probability of correct lane averaged over six vehicles

5.6.2 Numerical Results

The configurations of the MTF-PF algorithm are as follows: The size of candidate hypothesis vector set Θ_k is $K = 5$ and the shrinkage parameter η is set to 0.95 [13]. The AMELF and MTF-UKF algorithms are configured according to [17]. Note that a potential issue in the MTF-UKF algorithm is that its stability is sensitive to the initialization of desired speed estimates. The types of vehicles are assumed to be known in the MTF-UKF algorithm in order to initialize their desired speed estimates, accurately.

The root mean square error (RMSE) and the probability of correct lane [4] are

calculated over one hundred Monte Carlo runs to measure the accuracies of state estimates. Fig. 5.2 shows the RMSEs of mileage position estimates, desired speed estimates and the probability of correct lane averaged over six vehicles for AMELF, MTF-UKF and MTF-PF running with 100, 200 and 300 particles, respectively. The PCRLB values of mileage position and desired speed estimation errors under the assumption that the lateral state is known are also shown in Fig. 5.2. It can be seen that the MTF-based algorithms perform better than the AMELF. This shows the improvement resulting from considering interactions among vehicles. As expected, the MTF-PF with more particles yields better performance, especially in the beginning of scenario. This is because the initial motions of vehicles have large uncertainties, which requires more particles to counter. The AMELF has a delayed response to lane changes, resulting in deep valleys in the probability of correct lane. In contrast, the MTF-based algorithms accurately predicts these lane changes and the MTM-PF with 300 particles even maintains the probability of correct lane at around 0.96. With such an accuracy of lateral state estimates, the RMSEs of mileage position and desired speed estimates for MTM-PF with 300 particles are close to the corresponding curves of the PCRLB. It is worth noting that the MTF-UKF accurately initializes the desired speed estimates given the types of vehicles, while the PCRLB is derived and the MTF-PFs are run without this prior information. This makes the RMSE of desired speed estimates for MTF-UKF significantly lower than the PCRLB and that of the MTF-PF, and thus the MTF-UKF initially yields better mileage position RMSE and probability of correct lane than the MTF-PF. However, the subsequent accuracies of the MTF-UKF becomes worse than those of the MTF-PF.

Besides the accuracy, the track consistency is another important performance

Table 5.3: Statistics of track swaps over 100 Monte Carlo runs

Algorithm	Number of runs with swap	Total number of swaps	Swaps per run	
			Max	Mean
AMELF	81	472	17	5.83
MTF-UKF	25	202	20	8.08
MTF-PF (100)	49	265	17	5.41
MTF-PF (200)	40	195	15	4.88
MTF-PF (300)	35	153	12	4.37

metric in MTT problems [7], which is assessed by track swaps in this paper. A track swap is deemed to occur if a target is associated with a track that is different from the track associated in the previous scan but is the same as the track associated in the subsequent scan. Table 5.3 gives the statistics of track swaps over 100 Monte Carlo runs. The number of runs with track swaps and the total number of track swaps for AMELF are significantly more than those for the MTF-based algorithms. This is because, with the consideration for motion dependent information, the IDM ensures that the tracks keep a distance from their leading tracks, avoiding unrealistic sudden jumps ahead of their leaders. Also, the MOBIL helps correctly identify tracks surpassing slower tracks. Though the number of runs with track swaps for MTF-PF is more than that for MTF-UKF, the MTF-PF results in fewer track swaps in a single run than the MTF-UKF. Moreover, most of the track swaps for the MTF-PF take place in the beginning of the scenario. It can be expected that the number of runs with track swaps and the number of track swaps in these runs for MTF-PF will be less if the desired speed estimates can be accurately initialized as in the MTF-UKF.

Finally, the computational cost of the algorithms are evaluated on a 2.5 GHz Intel i7-4710HQ PC with 16 GB RAM. The computation times per run averaged over 100 Monte Carlo runs are given in Table 5.4. As expected, the MTF-PF has a higher

Table 5.4: Computation Times Averaged over 100 Monte Carlo Runs

Algorithm	Computation time per run (s)
AMELF [4]	0.56
MTF-UKF [17]	3.05
MTF-PF (100)	3.86
MTF-PF (200)	6.40
MTF-PF (300)	9.43

computational cost than the AMELF and the MTF-UKF due to the use of the PF. However, the computational demand of the proposed MTF-PF is still acceptable in real time applications in view of the computational capability of today's computers. Moreover, the computational cost of the MTF-PF shows a linear increase with the size of particle set. A trade-off between performance and computational efficiency of the proposed algorithm can be made for its applications in real systems.

5.7 Conclusions

The problem of multi-vehicle tracking on multi-lane roads was considered in this paper. The microscopic traffic flow model was used to model the dynamics of vehicles in the presence of interactions with surrounding traffic. The vehicle states that evolve according to the intelligent drive car-following model and the MOBIL lane-changing model were sequentially inferred using the particle filter with a computationally-efficient joint sampling method. A particle approximation of the likelihood function of model parameters was derived and the recursive maximum likelihood method was integrated into the particle filter to estimate unknown parameters in the dynamics model. The posterior Cramer-Rao lower bound for this problem was also derived. The experiments show that the proposed algorithm requires less prior information

while yielding better performance with respect to both accuracy and consistency of resulting tracks than existing algorithms for multi-vehicle tracking on multi-lane roads.

Bibliography

[1] A. Alessandri, M. Baglietto, and G. Battistelli, "A maximum-likelihood Kalman filter for switching discrete-time linear systems," *Automatica*, vol. 46, no. 11, pp. 1870–1876, Nov. 2010.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA: Artech House, 1999.

[3] X. Cao, X. Jiang, X. Li, and P. Yan, "Correlation-based tracking of multiple targets with hierarchical layered structure," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 90–102, Jan. 2018.

[4] Y. Chen, V. P. Jilkov, and X. R. Li, "Multilane-road target tracking using radar and image sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 65–80, Jan. 2015.

[5] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of Nonlinear Filtering*, vol. 12, no. 656-704, p. 3, Dec. 2009.

[6] A. D. Freitas, L. M., A. Gning, D. Angelova, and V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering," *Automatica*, vol. 69, pp. 380–394, Jul. 2016.

[7] A. Gorji, R. Tharmarasa, and T. Kirubarajan, "Performance measures for multiple target tracking problems," *Proc. 14th Int. Conf. Inf. Fusion*, pp. 1560–1567, Jul. 2011.

[8] M. Imani and U. M. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," *Automatica*, vol. 87, pp. 238–250, Jan. 2018.

[9] N. Kantas, A. Doucet, J. M. S. S. Singh, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statist. Sci.*, vol. 30, no. 3, pp. 328–351, Aug. 2015.

[10] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec.*, vol. 1999, pp. 86–94, 2007.

[11] ——, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Phil. Trans. R. Soc. A*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[12] Y. Li and D. Sun, "Microscopic car-following model for the traffic flow: The state of the art," *IET Control Theory Appl.*, vol. 10, no. 2, pp. 133–143, May 2012.

[13] C. Nemeth, P. Fearnhead, and L. Mihaylova, "Particle approximations of the score and observed information matrix for parameter estimation in statespace

models with linear computational cost," *J. Comp. Graph. Stat*, vol. 25, no. 4, pp. 1138–1157, Nov. 2016.

[14] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, "Review of microscopic lane-changing models and future research opportunities," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1942–1956, Dec. 2013.

[15] I. Saleemi and M. Shah, "Multiframe many-many point correspondence for vehicle tracking in high density wide area aerial videos," *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 198–219, Sep. 2013.

[16] T. Sathyan, M. L. Hernandez, A. Sinha, and T. Kirubarajan, "Weiss–Weinstein lower bound for maneuvering target tracking," in *Proc. SPIE–Signal Data Process. of Small Targets*, vol. 6236, Jun. 2006, pp. 1–9.

[17] D. Song, R. Tharmarasa, T. Kirubarajan, and X. Fernando, "Multi-vehicle tracking using microscopic traffic models," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–13, Mar. 2018.

[18] D. Song, R. Tharmarasa, T. Kirubarajan, and X. N. Fernando, "Multi-vehicle tracking with road maps and car-following models," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1375–1386, May 2018.

[19] R. Tharmarasa, T. Kirubarajan, M. Hernandez, and A. Sinha, "PCRLB-based multisensor array management for multitarget tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 539–555, Apr. 2007.

[20] P. Tichavskỳ, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds

for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[21] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin Heidelberg, Springer-Verlag, 2013.

[22] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[23] J. Vermaak, S. J. Godsill, and P. Perez, "Monte Carlo filtering for multi target tracking and data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, Jan. 2005.

[24] W. Yi, M. R. Morelande, L. Kong, and J. Yang, "A computationally efficient particle filter for multitarget tracking using an independence approximation," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 843–856, Feb. 2013.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The multiple on-road vehicle tracking using domain knowledge of on-road vehicle motion was studied in this thesis.

The bias correction is an indispensable step to mitigate bias in converted measurements of target position before using them for vehicle tracking. In this thesis, the determination of the bias by errors in sensor measurements was mathematically formulated. It was then decoupled from the true target position and approximately expressed by a linear function of the errors in sensor measurements, by assuming the errors are small. Based on the decoupled results, two computationally-efficient methods to estimate the errors and thus the bias were proposed. It was demonstrated by numerical results that the proposed bias correction method achieves similar performance to the previous optimization-based method while the proposed method is simpler and substantially faster.

In the context of the multiple on-road vehicle tracking problem, a novel tracking

algorithm, which integrates a car-following model as the motion model of vehicles on single-lane roads, was proposed. The notion of car-following cluster was defined to handle the motion dependence among vehicles in a group. To handle motion deviations from the integrated car-following model, a variable structure interacting multiple model algorithm was also integrated into the proposed tracking algorithm.

The proposed tracking algorithm was then extended to track multiple vehicles on multi-lane roads by integrating a lane-changing model for modeling the lateral motion of vehicles, i.e., lane-changing behaviors. The longitudinal and lateral states were estimated sequentially in a recursive manner using the unscented Kalman filter under the multiple hypothesis tracking framework. An adaptive deferred decision logic was proposed to further improve the accuracy of lateral state estimates with the cost of an increasing computation load. In order to better handle the dependence between longitudinal and lateral motions of a vehicle, the longitudinal and lateral states were jointly estimated under a unified particle filter framework with a specifically designed computationally-efficient joint sampling method. The recursive maximum likelihood method was integrated into the particle filter framework to online estimate unknown parameters in the integrated microscopic traffic flow models. Finally, the posterior Cramer-Rao lower bound was derived for tracking multiple vehicle on single-lane and multi-lane roads, respectively. Numerical results demonstrated that with the use of domain knowledge of on-road vehicle motion, the proposed tracking algorithms outperform conventional multiple target tracking algorithms. It was also shown that the improved algorithm based on the particle filter framework requires less prior information while generating more accurate and consistent tracks than the proposed

algorithm based on the unscented Kalman filter combined with the multiple hypothesis tracking framework.

6.2 Future Works

There are still a few limitations found in this work, which can be addressed in future works. Most importantly, the algorithms proposed in this thesis are only validated through simulations. Real experiments should be conducted in the future to further validate the proposed algorithms. Then, it is required to better handle mismatch between microscopic traffic flow models and real vehicle motions, especially for modeling human aspects of driving behavior, such as reaction time and imperfect driving. Furthermore, more domain knowledge can be exploited in the multiple on-road vehicle tracking. One typical example is that by analyzing the lane change history of a vehicle combined with the lane information (i.e., straight, left-turn and right-turn lanes), it helps to predict longitudinal maneuvers, upcoming lane-changing decision and driving direction in the intersection. Other domain knowledge that can be incorporated includes traffic lights, speed limit and even soft data.