12-2016

Traffic Surveillance and Automated Data Extraction from Aerial Video Using Computer Vision, Artificial Intelligence, and Probabilistic Approaches

Xi Zhao
Clemson University, muderx@gmail.com

TRAFFIC SURVEILLANCE AND AUTOMATED DATA EXTRACTION FROM AERIAL VIDEO USING COMPUTER VISION, ARTIFICIAL INTELLIGENCE, AND PROBABILISTIC APPROACHES

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Civil Engineering

---

by
Xi Zhao
December 2016

Accepted by:
Dr. Wayne Sarasua, Committee Chair
Dr. Ronnie Chowdhury, Committee Member
Dr. Jennifer Ogle, Committee Member
Dr. Joshua Levine, Committee Member

ABSTRACT

In transportation engineering, sufficient, reliable, and diverse traffic data is necessary for effective planning, operations, research, and professional practice. Using aerial imagery to achieve traffic surveillance and collect traffic data is one of the feasible ways that is facilitated by the advances of technologies in many related areas. A great deal of aerial imagery datasets are currently available and more datasets are collected every day for various applications. It will be beneficial to make full and efficient use of the attribute rich imagery as a resource for valid and useful traffic data for many applications in transportation research and practice.

In this dissertation, a traffic surveillance system that can collect valid and useful traffic data using quality-limited aerial imagery datasets with diverse characteristics is developed. Two novel approaches, which can achieve robust and accurate performance, are proposed and implemented for this system. The first one is a computer vision-based approach, which uses convolutional neural network (CNN) to detect vehicles in aerial imagery and uses features to track those detections. This approach is capable of detecting and tracking vehicles in the aerial imagery datasets with a very limited quality. Experimental results indicate the performance of this approach is very promising and it can achieve accurate measurements for macroscopic traffic data and is also potential for reliable microscopic traffic data. The second approach is a multiple hypothesis tracking (MHT) approach with innovative kinematics and appearance models (KAM). The implemented MHT module is designed to cooperate with the CNN module in order to

extend and improve the vehicle tracking system. Experiments are designed based on a meticulously established synthetic vehicle detection datasets, originally induced scale-agonistic property of MHT, and comprehensively identified metrics for performance evaluation. The experimental results not only indicate that the performance of this approach can be very promising, but also provide solutions for some long-standing problems and reveal the impacts of frame rate, detection noise, and traffic configurations as well as the effects of vehicle appearance information on the performance. The experimental results of both approaches prove the feasibility of traffic surveillance and data collection by detecting and tracking vehicles in aerial video, and indicate the direction of further research as well as solutions to achieve satisfactory performance with existing aerial imagery datasets that have very limited quality and frame rates.

This traffic surveillance system has the potential to be transformational in how large area traffic data is collected in the future. Such a system will be capable of achieving wide area traffic surveillance and extracting valid and useful traffic data from wide area aerial video captured with a single platform.

DEDICATION

This dissertation is dedicated to my parents — Dr. Zaili Zhao and Mrs. Xiaolan Huang

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Dr. Wayne Sarasua, for giving me an opportunity to work with him and pursue my doctorate. His intelligence, passion and optimism have been a great source of inspiration and encouragement for me in all facets of my life. He encouraged and gave me plenty of freedom to be an independent and active thinker as well as an open-minded collaborator.

My appreciation goes to Dr. Ronnie Chowdhury, Dr. Jennifer Ogle and Dr. Joshua Levine for being on my advisory committee and for their priceless contributions towards attaining my doctoral degree. I would also like to thank Dr. Stanley Birchfield for his insightful advices and input in my research. I am very fortunate and grateful to learn from the most intelligent and thoughtful minds.

I would like to thank a fellow doctoral student from the Department of Electrical and Computer Engineering, Mr. Douglas Dawson, who is always happy to share his knowledge with me and offering help, for excellent collaborating in our research. I am also grateful to all my colleagues in the Department of Civil Engineering.

Last, but most, I would like to thank my parents, Dr. Zaili Zhao and Mrs. Xiaolan Huang. They have always been supportive in my life; their selfless love enabled my life's journey till now.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AdaBoost	Adaptive Boosting
ADAS	Airborne Data Acquisition System
ATSS	Airborne Traffic Surveillance Systems
CLIF	Columbus Large Image Format
CLT	Central Limit Theorem
COMETS	Real Time COordination and control of Multiple heterogeneous unmanned aerial vehiclES
CNN	Convolutional Neural Network
ConvNet	Convolutional Neural Network
DBN	Dynamic Bayesian Network
HOG	Histogram of Oriented Gradients
ITS	Intelligent Transportation System
JPDA	Joint Probabilistic Data Association
KAM	Kinematics and Appearance Models
KLT	Kanade-Lucas-Tomasi Feature Tracker
LOS	Level of Service
MHT	Multiple Hypothesis Tracking
MTT	Multiple Target Tracking
O-D	Origin and Destination
PSS	Persistent Surveillance Systems
RSPA	Research and Special Programs Administration
ROI	Regions of Interest
SDMS	Sensor Data Management System
SURF	Speeded Up Robust Features
TLD	Tracking-Learning-Detection
TRAVIS	Tracking and Registration of Airborne Video Image Sequences

UAV	Unmanned Aerial Vehicle
USDOT	United States Department of Transportation
WAPS	Wide Area Persistent Surveillance

CHAPTER I

INTRODUCTION

1.1 Background

Reliable and sufficient data is the foundation of research and professional practice in science and engineering. In transportation engineering, sufficient, reliable, and diverse traffic data is necessary for effective planning, operations, research, and professional practice related to transportation systems. Many technologies have been developed to collect different types of traffic data. Traditional data collection technologies include bending plates, pneumatic road tubes, piezoelectric sensors, inductive loop detectors, passive and active infrared sensors, magnetometers, microwave radar devices, ultrasonic acoustic devices, video detection systems, and even manual observations (Traffic Monitoring Guide, 2016). These technologies are used to monitor traffic and collect data throughout a traffic network, including traffic volume, time-mean speed, vehicles classification, occupancy, etc. While traditional traffic monitoring technologies have proven to be effective, a major drawback is that they can only collect certain types of traffic data at fixed locations. The traditional video detection systems using virtual detectors are also at fixed locations. With recent advancements of computer vision technology, field of view tracking of vehicles is now possible using specialized algorithms that process video collected from cameras mounted on poles on the side of roads (Kanhere, et al., 2010). Because cameras can cover much wider areas than traditional sensors and the captured video contains a variety of information from which various types of traffic data can be

extracted using specialized vehicle tracking algorithms, some video detection systems can provide traffic data for specialized transportation applications including monitoring driving behavior (Tsai, et al., 2011) and collision detection (Saunier & Sayed, 2007). Compared to most traditional technologies, the video-based systems have enlarged area coverage but this area is still limited to field of view and the performance is not uniform throughout field of view.

Can the coverage be wider, and can the data be even more informative? The answer is YES if the field of view is greatly expanded. High resolution satellite imagery can give a very wide field of view but continues video is impractical for traffic surveillance and data collection. With the advent of high performance optical sensors mounted on aircraft it is now possible to record high resolution video for a relatively wide field of view from aircraft overhead. Figure 1 shows such a camera array (HawkEye II from Persistent Surveillance Systems (PSS)) that is configurable for a variety of aircraft. The area of the coverage varies and depends on the specifications of the optical sensors and the altitude of the platform but wide area persistent coverage is now possible. For example, the aerial imagery in PSS dataset used in this research covers an area of approximately 5 mi by 5 mi, as shown in Figure 2.

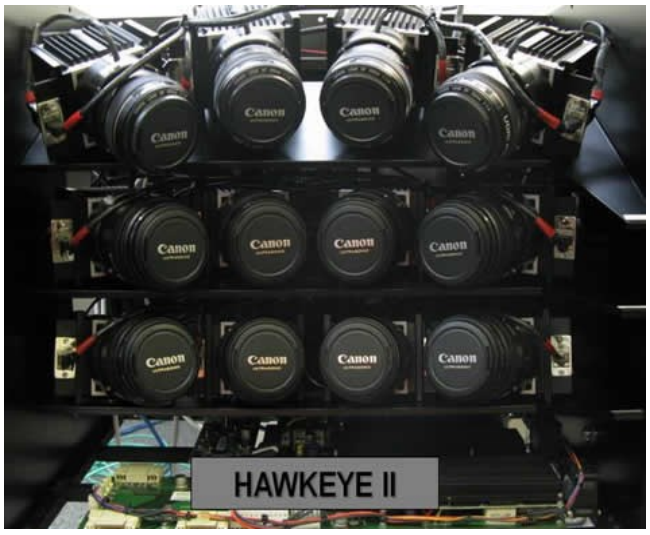


Figure 1 HawkEye II aerial camera array (courtesy of PSS)

Another relatively recent development in platforms for aerial imagery is the proliferation of unmanned aerial vehicle (UAV) usage. The use of UAVs has become so popular that regulations are under development in many countries (Nex & Remondino, 2014). In the United States, it is regulated in all fifty states and there are federal regulations as well. A primary benefit of UAVs is that they are cost-efficient.

Regardless of platform, aerial imagery contains more practical macroscopic traffic information than the output of fixed-location sensors and cameras. For example, density is one of the most important variables used in measuring the performance of roads. Traditional methods used to estimate density involve collecting occupancy, time-mean speed, and volume at discrete locations and then use these data to calculate density. A better estimate of density requires the use of space-mean speed rather than time-mean speed. Collecting accurate space-mean speed requires tracking all vehicles over a distance which cannot be collected by fixed-location sensors/cameras. In contrast to those methods, aerial

imagery can be used to directly determine density by counting the number of vehicles on a mile of roadway. Additionally, counting vehicles passing a point combined with space-mean speed measurements by tracking vehicles over a distance can also give accurate density values. Besides macroscopic traffic data, aerial video also contains more informative microscopic traffic data than the outputs of fixed-location devices. By monitoring aerial image sequences of a wide area overhead, as shown in Figure 2, it is possible to extract accurate location, speed, acceleration/deceleration, travel time, origin and destination (O-D) data, and car-following data as well as other driver behavior characteristics. Some types of these data are difficult and costly to collect by traditional technologies; however, it can potentially be cost-efficient by tracking each vehicle in the network with aerial video.

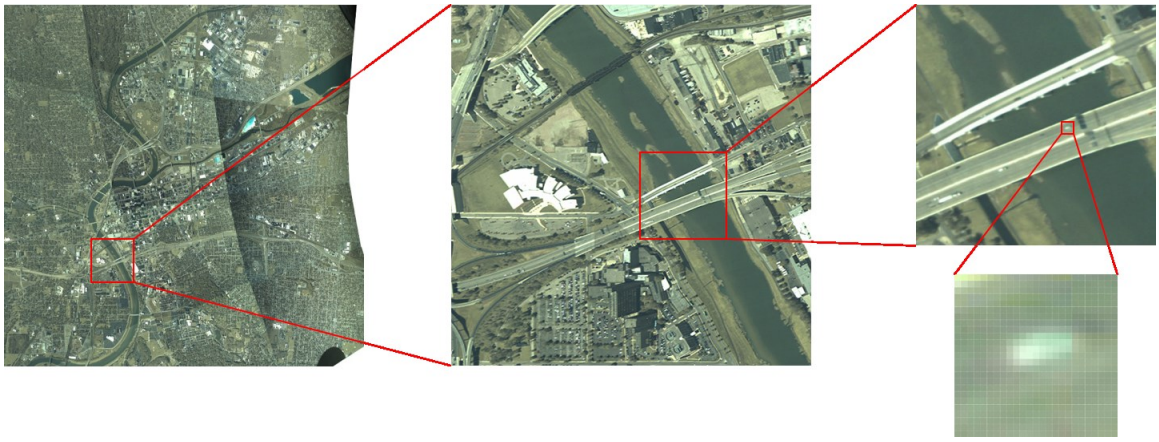


Figure 2 Wide area surveillance using aerial imagery (courtesy of PSS)

1.2 Motivation

While optical sensing and UAV technologies have significantly improved in recent years, the capture of aerial imagery is neither unaffordable nor impractical. Obstacles that can make aerial surveillance impractical relate to challenges on how to extract reliable and sufficient information. Automating the extraction of useful information, such as traffic data, from captured aerial imagery can potentially be transformational with regard to how large amounts of traffic data can be collected in an efficient manner.

Until recently, people analyzed aerial photos with the assistance of digital image processing technology, using methods that were not automated. In recent years, many methods based on computer vision have been developed to monitor traffic by detecting and tracking vehicles in aerial videos but the feasibility of them is limited by the characteristics and the quality of imagery data. For example, feature detection/modeling methods (Moon, et al., 2002; Kim & Malik, 2003; Zhao & Nevatia, 2003; Hinz, 2005; Palaniappan, et al., 2010; Pelapur, et al., 2012; Cao, et al., 2012) require high resolution imagery, while background subtraction/modeling methods (Xiao, et al., 2010; Shi, et al., 2013; Prokaj & Medioni, 2014; Saleemi & Shah, 2013; Chen & Medioni, 2015) require high quality preprocessing including stabilization, rectification, etc. Unfortunately, the characteristics and quality of most available datasets of aerial imagery cannot guarantee promising performance of existing vehicle detection and tracking methods for several reasons:

1. One reason is that most aerial imagery datasets are not specifically collected for the purpose of traffic surveillance, but for their own purposes, such as law enforcement,

critical infrastructure protection, event security, and emergency response after natural disasters (Palaniappan, et al., 2011). Their characteristics and quality make it challenging to process for traffic surveillance. For example, the datasets provided by PSS used in this research were originally intended to "make our families and our neighborhoods safe places to live and work," and the vehicles in each frame are very difficult to distinguish for algorithms, as shown in Figure 2.

2. Another reason is most aerial imagery datasets are proprietary and only a few of them are public with varying characteristics and quality. Many researchers specifically developed their own algorithms to process certain datasets but the transferability of their algorithms to other datasets may not give good results.
3. Furthermore, some vehicles detection and tracking algorithms are not specifically intended to track vehicles in aerial imagery (Kalal, et al., 2012).

In order to make it feasible to monitor traffic and extract traffic data from aerial imagery, a high quality dataset is always preferred. Normally, people use low altitude platforms to record high frame rate video for high resolution imagery data, with which the performance of their algorithms are acceptable. As a result, the area of coverage is inevitably limited. Of course, the advancement of optical sensor technology will continuously improve the quality of aerial imagery in the future which will make the processing for traffic surveillance and data extraction easier. However, numerous datasets of aerial imagery already exist, and more data is being produced and will continue to be produced. It will be beneficial to make full and efficient use of this attribute rich imagery

as a resource for reliable and useful traffic data for many applications in transportation research and practice.

Is it possible to develop an approach that is capable of monitoring traffic and extract traffic data, using aerial imagery of which the quality is limited or the imagery is not specifically collected for the purpose of traffic data collecting? How to automatically, accurately, and efficiently extract reliable and informative traffic data to support transportation research and practice? To answer these questions, this dissertation proposes in-depth research of traffic surveillance and data extraction approaches using quality-limited aerial video.

1.3 Goals and Objectives

The overall goal of this dissertation is to research and implement a reliable, robust, and automated system that is capable of monitoring traffic and extracting a variety of traffic data from wide area aerial imagery datasets of limited quality and frame rate. More specifically, the goal is to design and implement vehicle detection and tracking approaches, evaluate their performance, and explore some critical factors to achieve satisfactory performance for existing aerial imagery datasets. This is a process that requires interdisciplinary literature review, extensive coding, in-depth research, and convincing experiments.

The first objective is to investigate existing research related to traffic surveillance approaches and data extraction from aerial imagery to help guide this research and determine its contributions.

The second objective is to investigate challenges with automatically processing wide area aerial video and extracting useful traffic data.

The third objective is to implement, test, and evaluate computer vision-based vehicle detection and tracking approaches to monitor traffic and collect reliable traffic data using aerial imagery.

The fourth objective is to identify and refine the most promising approach and determine its capabilities, robustness, and limitations that may guide future research.

This research will potentially lead to a traffic surveillance and data collection system that will tap into a data source that has promising big data potential to support traffic monitoring applications as well as academic research.

This research will not address any detailed content about imagery capturing devices (optical sensor and platform), the process of capturing aerial imagery or the preprocessing of the imagery datasets accomplished by the data providers. They are only mentioned briefly in the literature review to provide background knowledge to readers for better understanding of existing research.

1.4 Organization

In Chapter II, a comprehensive and interdisciplinary literature review is conducted in several areas closely related to the traffic surveillance and traffic data collection using aerial imagery. The literature review starts with two most impacting emerging technologies in traffic surveillance: UAV and wide area persistent surveillance (WAPS), which

determine the characteristics and specifications of many existing aerial imagery datasets. Then, a comprehensive literature review of computer vision-based algorithms and approaches is conducted, including feature detection/modeling, background subtraction/modeling, and machine learning. Further, the existing applications of MHT and convolutional neural network (CNN) in related areas of traffic surveillance and vehicle detection and tracking are reviewed.

In Chapter III, several available commercial and public aerial imagery datasets that are potentially capable of providing sufficient visual information for the purpose of traffic surveillance are investigated and analyzed. The challenges identified and solved in this dissertation are not only caused by the limited accessibility of aerial imagery datasets, but also the diverse characteristics and the limited quality due to their specifications.

In Chapter IV, a feature-based tracking framework and a CNN-based detector are presented, and then a novel CNN approach with speeded up robust features (SURF) is proposed and implemented by combining them. The estimation methods of traffic data for different measurements are analyzed. Experiments are implemented to evaluate the performance of the implemented CNN approach, and experimental results are presented and discussed.

In Chapter V, a MHT approach with an innovative kinematics and appearance models (KAM) structure is proposed and implemented to extend and improve the vehicle tracking system by cooperating with the MHT approach presented in Chapter IV. To evaluate the performance of the MHT approach with KAM, explore concerned critical

factors, and solve long-standing problems, experiments are designed based on a meticulously established synthetic detection dataset, originally induced scale-agonistic property of MHT, and comprehensively identified metrics for performance evaluation. The experimental results are presented and discussed.

In Chapter VI, the dissertation goals and objectives are restated along with relevant findings and main contribution of this dissertation is summarized. Future research is proposed.

CHAPTER II

LITERATURE REVIEW

2.1 Aerial Surveillance Systems

The most impacting emerging technologies in the field of aerial surveillance are UAV and WAPS that have drawn significant attention from many relevant fields in recent years. A large proportion of existing datasets investigated in this research were collected using these technologies.

2.1.1 Traffic Surveillance Systems using UAVs

Aerial imagery has been used to achieve a broad vision of a traffic network or a specific facility for decades in the field of remote sensing. Because of drawbacks for systems using manned aircraft, including low cost efficiency, etc., and lack of efficient methods to extract useful traffic data, traditional data collection technologies have been preferred by professionals for a long time. People commonly observe the static view overhead for static information, especially in the case that the public sources, like Google Earth/Map and Bing Maps are free for use. However, with the advent of UAVs in civilian applications, UAVs have demonstrated a great potential to be a feasible platform for traffic surveillance, data collection, and a part of intelligent transportation system (ITS) infrastructure because of easy maneuvering, great flexibility, and promising cost efficiency (McCormack & Trepanier, 2008).

Many institutes and DOTs have presented a great interest in the application of UAVs in traffic surveillance. University of Florida introduced Airborne Traffic Surveillance Systems (ATSS) sponsored by Florida DOT. The research team used a UAV to collect timely video data for the purpose of traffic surveillance in rural areas in Florida, and to evaluate the feasibility of integration of ATSS with Florida DOT's microwave communication system (Puri, 2005). Ohio DOT conducted research in cooperation with Ohio State University to study potential benefits of applications of UAVs in transportation surveillance by monitoring specific transportation facilities (Puri, 2005). A research team led by Research and Special Programs Administration (RSPA) of United States Department of Transportation (USDOT) used a UAV to collect and interpret real-time multi-modal traffic data using its road-following capabilities (Puri, 2005). Virginia DOT demonstrated the feasibility of Airborne Data Acquisition System (ADAS) for the purpose of traffic surveillance (Puri, 2005). Researchers at University of South Florida used a UAV to collect real-time temporal/spatial data for the purpose of monitoring traffic, counting vehicles, and evaluate and assess traffic patterns (Puri, et al., 2007). Researchers at University of Washington conducted research to explore the general capabilities of UAVs and evaluate their potential as an avalanche control tool (McCormack & Trepanier, 2008). This research, sponsored by Washington State DOT, found UAVs are suitable for traffic surveillance and data collection. In Europe, the project named COMETS (Real Time COordination and control of Multiple heterogeneous unmanned aerial vehiclES) focused on the design and implementation of distributed control system using heterogeneous UAVs

for the purpose of monitoring traffic situation, and to identify and track individual vehicles, etc. (Puri, et al., 2007).

Most research of traffic surveillance systems using UAVs focused on the feasibility, framework, or devices of the systems. Very few of them focused on the methods and algorithms of extracting traffic information from captured imagery. In recent years, more and more research explored the methods and algorithms of vehicles detection and tracking using the imagery data collected by UAVs. They will be reviewed in Chapter 2.2. A number of public imagery datasets available in Sensor Data Management System (SDMS) were collected by UAVs. Most of them present the following characteristics: single optical sensor and low altitude. The use of a single optical sensor makes it straight-forward to preprocess the imagery because image stitching is not necessary. Low altitude only allows a limited area of the coverage and the small pixel size makes objects easy to distinguish.

2.1.2 Wide Area Persistent Surveillance

WAPS by aerial imaging is a newly evolving technology that enables persistent coverage of a large geographical region (Palaniappan, et al., 2011). The use of camera arrays and the application of digital image processing and computer vision technologies enable the system to monitor a large region up to multiple square miles depending on the altitude of the platform and the configurations of the camera array. The platforms typically use a circular flight path to cruise at a constant altitude on the targeted region and the camera arrays are continuously adjusted to maintain the orientation fixed around the area of interest. The coverage of the surveillance can remain constant for a considerably long time depending

on the flight time of the platform. One of significant advantages of WAPS compared to conventional aerial surveillance is that the flight plan does not rely on concerned targets, which means the platform does not have to follow specified targets because of the large region of coverage.

A number of potential applications of WAPS exist in many fields. Currently, the most widely accepted applications include urban planning, ecological survey, agricultural survey, law enforcement, and event security. In the applications of law enforcement and event security, individuals of interest are manually tracked and the concerned information is relayed to ground personnel. Many potential applications of WAPS are promising even though numerous challenges exist for the WAPS system developers and users (Palaniappan, et al., 2011). These challenges include the need for improved sensors calibration, better estimation of platform dynamics, accounting for lighting variability, seamless image mosaicking, and image geo-registration.

In this research, several WAPS imagery datasets were investigated. These include the commercial dataset from PSS and public datasets available in SDMS, including Columbus Large Image Format (CLIF) 2006/2007. All of these datasets presented the following characteristics: seamed mosaicking/stitching, stabilization problems, low frame rate and large pixel size that make it extremely challenging to apply them for traffic surveillance, especially for microscopic traffic surveillance by tracking vehicles in the imagery.

2.2 Digital Image Processing/Computer Vision Algorithms

In recent years, many researchers have shown a great interest in applying emerging technologies of digital image processing and computer vision for traffic surveillance and traffic data extraction. A considerable number of approaches and algorithms have been developed for the purpose of detecting and tracking vehicles in aerial imagery. For example, researchers at University of Arizona proposed an approach for collecting and analyzing aerial imagery and outlined the methods to estimate speed, travel time, density and queuing delay using background subtraction (Angel, et al., 2003). The imagery used in this research was captured by a single standard resolution camera (720×480 pixels) mounted on a helicopter flying at an altitude of under 305 m (1000 feet), which provided a field of view of less than 244 m (800 feet) In the following year, they presented an automated method to estimate intersection queue length in aerial imagery using connected component analysis of the region of interest (Agrawal & Hickman, 2004). Their method estimated the queue length by two techniques: counting individual vehicles in the queue and obtaining the area of the polygon containing vehicles in the queue. Their continued research led to the development of a software they named "Tracking and Registration of Airborne Video Image Sequences" (TRAVIS) which can extract vehicles positions by detecting and tracking vehicles through the image sequence to assist the analysis of microscopic traffic behavior (Hickman & Mirchandani, 2006). TRAVIS implemented an algorithm that consisted of image registration, background subtraction, and filtering and tracking blobs to output a sequence of pixel coordinates of vehicles through the image sequence. A follow-up research was conducted to improve vehicle detection and reduce the probability of false

detection and computation time by masking the area outside roadways (Du & Hickman, 2012). They also improved the tracking algorithms to better handle vehicles with little contrast relative to the pavement of roadways. The imagery data used in their recent work was captured by a single camera mounted on a helicopter that provide a coverage ranging from 500 to 1000 feet across and a pixel size ranging from 0.3 to 1.3 pixel/feet at the frame rate of 30 fps.

Most existing digital image processing and computer vision approaches for vehicle detections and tracking using aerial imagery datasets can be categorized in several classes: feature detection/modeling, background subtraction/modeling and machine learning, and they are reviewed in the following sections.

2.2.1 Feature Detection/Modeling

A large proportion of vehicle detection and tracking algorithms are based on distinguishable pixel information. Many feature-based techniques, including Kanade-Lucas-Tomasi feature tracker (KLT) and histogram of oriented gradients (HOG), are widely used to either label or model vehicles for the purpose of vehicle detection or tracking in previous work. Moon et al. presented a vehicle detection algorithm by combining four elongated edge operators to search the sides of a vehicle (Moon, et al., 2002). The performance of this model-based algorithm was affected by camera angles, illuminations, and site information. Kim and Malik presented a model-based 3D vehicle detection and description algorithm (Kim & Malik, 2003). This algorithm modeled vehicles by a probabilistic line features grouping, and the results indicated it outperformed the

algorithm presented by Zhao and Nevatia (Zhao & Nevatia, 2003). Hinz presented a vehicle detection approach by fusing the local model of cars with a 3D wireframe representation of individual vehicle and the global model of the grouping of cars within queues with ribbons (Hinz, 2005). The advantage of this algorithm was that it neither relied on external information nor was limited to very constrained environments. Palaniappa et al. presented a vehicle tracking system based on a set of feature detectors using appearance modeling, saliency estimation and motion prediction (Palaniappan, et al., 2010). The results indicated that fusing feature likelihood maps improved performance, but combining saliency information degraded the performance for low frame rate imagery. In follow-up research, Pelapur et al. presented a tracking system based on feature likelihood maps and adaptive appearance target update model (Pelapur, et al., 2012). The results indicate that this system outperformed other vehicle trackers with several wide area aerial imagery datasets including CLIF dataset. Cao et al. presented a framework for vehicles detection and tracking using aerial imagery collected by UAV platform (Cao, et al., 2012). This method was based on KLT features, and the result indicated that it was more accurate for detection, more efficient and robust to partial occlusion and computationally simpler than other algorithms.

2.2.2 Background Subtraction/Modeling

Background subtraction and modeling are widely used in many algorithms; however, those algorithms require either highly stabilized optic sensors or accurate image registration. Reinartz et al. used background subtraction to find vehicles and image patch correlation to match the vehicles between frames (Reinartz, et al., 2006). Their approach had issues with

mistakenly detecting pedestrians and grouping vehicles that were too close together. Xiao et al. presented a method of probabilistic relation graphing combining a vehicle behavior model with a road network for the purpose of vehicle detection and tracking in wide area aerial imagery (Xiao, et al., 2010). The experiments indicated the method can produce robust long time results. Shi et al. presented a spatio-temporal context model of maximum consistency context for multiple object tracking by leveraging the discriminative power and robustness in wide area traffic scenes (Shi, et al., 2013). Experimental results validated the effectiveness of this approach. Saleemi and Shah presented a framework capable of tracking thousands of vehicles in low frame rate aerial videos by maintaining multiple object-centric associations for each track, and using background subtraction and modeling (Saleemi & Shah, 2013). The results indicated that this method outperformed global, one to one data association methods. Prokaj and Medioni presented a multiple objects tracking approach using two trackers in parallel: one using background subtraction and the other using a regression tracker (Prokaj & Medioni, 2014). The results indicated this approach improved object detection rates and ID-switch rates with limited increases in false alarms comparing to its competitors. Chen and Medioni presented two methods to achieve more accurate background model (Chen & Medioni, 2015). The first methods predicted the image flow and perform pixel-level classification for detection using a dense 3D model of the landscape, and the second method used the epipolar flow constraint to distinguish object motion. The results indicated a significant improvement in detection rate and speedup.

2.2.3 Machine Learning

Machine learning is usually applied in algorithms of vehicle detection and tracking in cooperation with either feature detection/modeling or background subtraction/modeling. Zhao and Nevatia presented a passenger car detection system by modeling passenger cars as 3D objects using a Bayesian network to integrate a set of features (Zhao & Nevatia, 2003). The experiments indicated promising results. Nguyen et al. presented an automated car detection framework using online adaptive boosting (AdaBoost), which was trained with three types of features, and a mean shift clustering method (Nguyen, et al., 2007). Experimental results indicated this framework is superior and applicable for many applications. Tuermer et al. presented a detection approach using a fast preprocessing to limit the search space and Real AdaBoost trained with HOG features for detection (Türmer, et al., 2010). The results indicated high detection rate and reliability of this approach. Cheng et al. presented a pixelwise classification method in which a dynamic Bayesian network (DBN) was constructed based on color and edge detection for the classification purpose (Cheng, et al., 2012). The results indicated the flexibility and generalization capability of this method.

Most vehicle tracking approaches are based on vehicle detections, which use the visual information to initialize the tracker or support the tracking process by matching correspondences between adjacent frames. However, the tracking process and detection process are mutually dependent in some approaches. Kalal et al. proposed Tracking-Learning-Detection (TLD) framework which combined tracking, learning, and detection (Kalal, et al., 2012). The tracker generates training data for improving the detector, and the

detector initializes and re-initializes the tracker simultaneously. Experimental results indicated the superiority of this approach compared to competitors for many different tasks.

2.3 Multiple Hypothesis Tracking

Multiple hypothesis tracking is an algorithm proposal by Reid in 1979 that has been widely preferred for data association in multiple target tracking (MTT) tasks (Reid, 1979). Practical implementation of MHT is very challenging because of its computational complexity. However, since the improved implementation methods (Cox & Hingorani, 1996)and upgraded computing hardware, the practical real-time implementation was proven feasible (Blackman, et al., 2001).

Because most surveillance applications must be capable of tracking multiple targets, MTT is one of the most important tasks for those applications. In many frameworks, modules of sensors, including radar, infrared, and sonar, report measurements to MHT modules for sensor data association. MHT approach has been applied in many areas, including track confirmation, agile beam radar, missile defense systems and ground target tracking, which is probably the most challenging application (Blackman, 2004).

Security restrictions and proprietary policies greatly restrict MHT researchers to publish and share their work (Blackman, 2004), thus very little literature presenting details of applications of MHT were found. Arambel et al. presented a brief report that proposed an automated video-based ground targeting system for UAVs (Arambel, et al., 2004). This system used a module of background subtraction and site modeling to extract features and

report the measurements to a MHT module for tracking multiple ground targets simultaneously.

2.4 Convolutional Neural Network

Deep learning is the state-of-the-art reinvention of artificial neural network and there is clear evidence that for an off-line detection/classification problem no handcrafted techniques, including support vector machine (SVM), principal component analysis (PCA), scale-invariant feature transform (SIFT), HOG., can compete with deep learning. Convolutional neural network (CNN or ConvNet) is a class of multiple layers neural networks, which are specifically designed for two-dimensional data (LeCun, et al., 1998). It is the first truly successful deep learning approach (Arel, et al., 2010).

Few literatures were found about the application of CNN in vehicle detection or tracking using aerial imagery. Chen, et al. presented a hybrid deep CNN to extract variable-scale features in high resolution satellite imagery (Chen, et al., 2014). The results indicated that this network outperformed other traditional machine learning approaches on vehicle detection. The static satellite imagery used in their work has higher resolution than many aerial imagery datasets investigated in this research.

CHAPTER III

AERIAL IMAGERY DATA

3.1 Characteristics and Specifications of Datasets

Several commercial and public aerial imagery datasets from various sources, including PSS, Skycomp, and SDMS, were investigated in this research. These datasets were collected by different systems with differing preprocessing techniques. Thus, they present diverse characteristics and all of them have limited quality for some specifications which make them challenging for automated vehicle detection and tracking to extract reliable traffic data. Characteristics and specifications of these dataset are summarized in Table 1.

Table 1 Aerial imagery datasets and their characteristics

Dataset	Camera Type	Image Type	Coverage	Frame Rate	Resolution*	Rectification	Aligned	Mosaicking	Stabilized	Luminance
PSS	Array	Color	Wide (5×5 mi^2)	1 fps	0.5 m/pixel	Yes	Yes	Seamed	No	Inconsistent
Skycomp 18mm	Single	Color	Moderate (1.8×1.2 mi^2)	1 fps	about 0.4 m/pixel	No	No	NA	No	Consistent
Skycomp 50mm	Single	Color	Small (0.6×0.4 mi^2)	1 fps	about 0.2 m/pixel	No	No	NA	No	Consistent
Skycomp 1 aligned	Singe	Color	Small	1 fps	about 0.1 m/pixel	No	Yes	NA	Yes	Consistent
Skycomp 2 aligned	Single	Color	Moderate	1 fps	about 0.4 m/pixel	No	Yes	NA	Yes	Consistent
CLIF 2006	Array	Grayscale	Moderate	2 fps	about 0.2 m/pixel	No	No	Seamed	No	Inconsistent
CLIF 2007	Array	Grayscale	Moderate	2 fps	about 0.2 m/pixel	No	No	Seamed	No	Inconsistent
WPAFB 2009	Array	Grayscale	Moderate	about 1.3 fps	0.5m /pixel	Yes	Yes	Seamed	No	Inconsistent

*Only rectified datasets have unified resolution.

3.2 Challenges for Quality-Limited Aerial Imagery

In this research, multiple aerial imagery datasets were investigated and numerous challenges were identified that exist for all of them. These challenges are caused by the limited quality and diverse characteristics of those datasets. They have to be overcome for automated processing and reliable traffic data to be possible. These challenges are identified as follows:

1. The imagery is not completely stabilized since shaking is inevitable during flight. Notice how the image in Figure 3 (b) is shifted to the left from Figure 3 (a), even though they were rectified and aligned in preprocessing.
2. The sub-images from different cameras in the array are not exactly aligned in the case that the imagery is captured by camera arrays, and the stitched images are seamed, as shown in Figure 3 (b).
3. The illumination of sub-images from different cameras in the array is not consistent, as shown in Figure 3 (d). Even the illumination in a single sub-image from a camera in the array is not consistent.
4. Most imagery data used in this research was preprocessed by their providers. The details and the quality of the preprocessing vary depending on the dataset. For example, the dataset from PSS is well rectified but aligned poorly, while some datasets from Skycomp are well aligned but not rectified. The CLIF 2007 dataset is neither rectified nor stitched well but it provides the source code used for

stitching. The critical issue is that most datasets are proprietary and the author does not have access to the details how the datasets were preprocessed.

5. Most datasets have low resolutions and the resolutions vary depending on the dataset. For example, the dataset from PSS is well rectified and the unified resolution is 0.5 m/pixel. Thus, there are very few distinguishable features available to detect vehicles from a static image as shown in Figure 3 (c). The resolution of the dataset from Skycomp is relatively high (approximately 0.2 m/pixel from a 50 mm camera and 0.4 m/pixel from an 18mm camera). However, the imagery is not rectified, so it does not have a unified resolution and the approximate values cannot be applied to the whole image.
6. On the other hand, all datasets have a large image size. For example, a frame from PPS is 16384×16384 pixels for 5×5 mi^2. A frame from Skycomp is 5616×3744 for about 0.6×0.4 mi^2 from a 50mm camera; and about 1.8×1.2 mi^2 from an 18mm camera. Some processing toolkits cannot fully support imagery data of this resolution.
7. All the datasets have low frame rates that make vehicle tracking very challenging. The dataset from PSS and Skycomp are only 1 Hertz, while the CLIF 2007 and 2008 datasets are 2 Hertz. For the PSS dataset, a vehicle traveling at 60 mph on a freeway will travel 26.8 m in a second (53.6 pixels) making it very challenging to match correspondences between frames.
8. The amount of data is huge. For the PSS dataset, a single compressed frame is 40-50 MB and a single uncompressed frame is nearly 1 GB, thus proposed algorithms

require high computational and memory efficiency to achieve practical or near real-time execution.

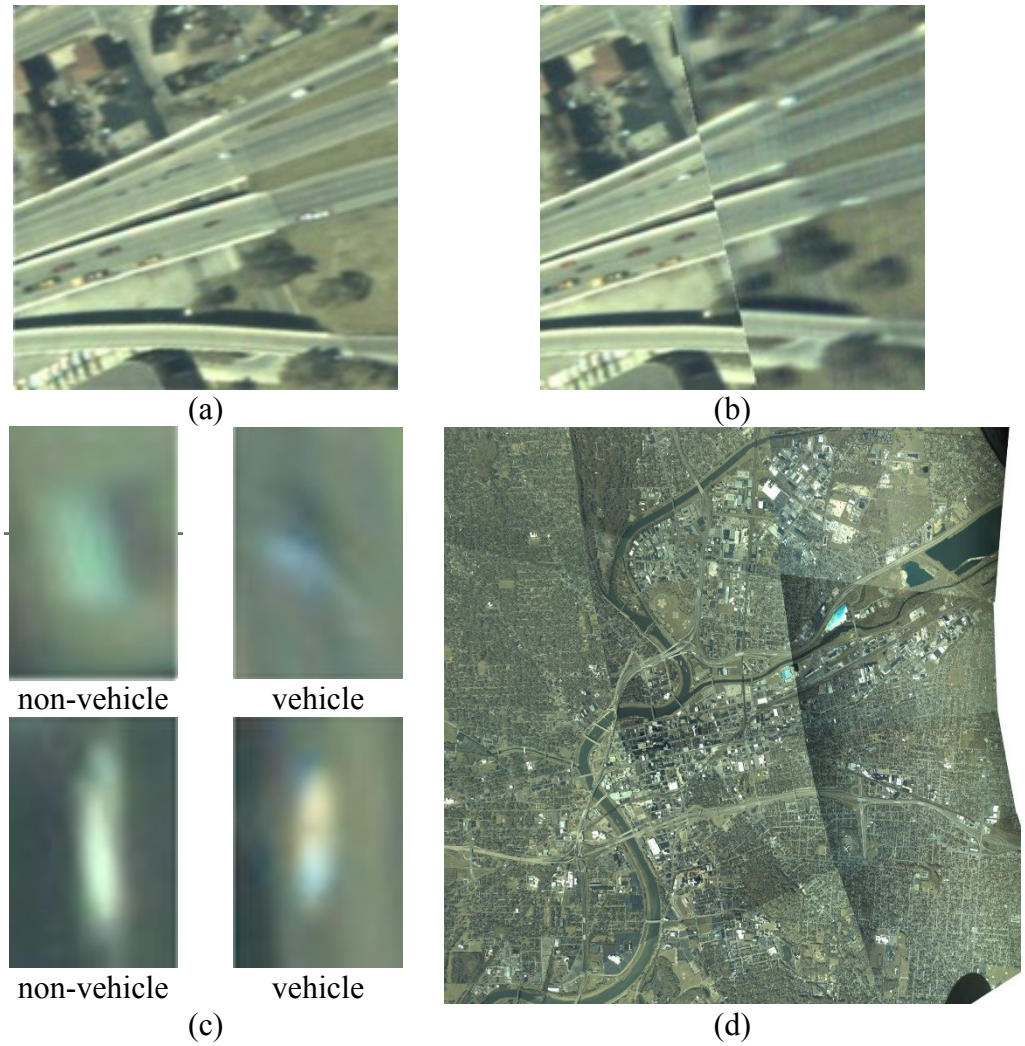


Figure 3 (a) A section of an image frame; (b) the same section in the next frame, showing a seam; (c) zoomed portions of a frame showing vehicles and other objects; (d) a frame of the input data (courtesy of PSS)

CHAPTER IV

COMPUTER VISION VEHICLE DETECTION AND TRACKING APPROACH

4.1 Methodology

To explore the possibility of vehicle detection and tracking for aerial imagery, the author tried several different heuristic and machine learning approaches of computer vison technology. Those approaches include a pixel-based approach, a feature-based approach, a SVM-based approach, and a CNN-based approach. Simple tests and comparisons were used to identify their basic performance to process wide area aerial imagery data and their potential capabilities to overcome the challenges identified in Chapter III. In this chapter, the author proposes and combines the two most promising approaches to achieve the computer vision-based approach for macroscopic traffic data extraction from wide area aerial video. The first approach is a feature-based vehicle tracking framework, whereas the second approach is a vehicle detector based on CNN. The implementation of both approaches was tested with the PSS dataset; however, the developed module combining them is scalable to other aerial imagery datasets.

4.1.1 A Feature-Based Vehicle Tracking Framework

The approach presented in this section is a general framework for vehicle tracking with a variety of handcrafted features that have been widely used in digital image processing and computer vision.

This vehicle tracking framework is dependent on feature detection, and matching the corresponding features in consecutive frames in order to label the corresponding vehicles in a sequence of frames. It is a heuristic approach based on the fact that the cluster of features representing the tracked object does not significantly morph appearance between consecutive frames. Thus, the sample vehicle can be identified in a sequence of frames by explicitly matching the specific representative features of the vehicle with adequate methods (detection, matching and filtering). The basic procedure is shown in Figure 4 and as follows:

1. Manually specify the vehicle to track in the initial frame.
2. For each pair of consecutive frames:
    a. Create regions of interest (ROIs) in both frames. The size of ROI was predefined (100×100 pixels in the implementation).
    b. Extract background information in both ROIs.
    c. Detect (SURF) features in both ROIs.
    d. Identify the cluster of the features representing the tracked vehicle in the first frame.
    e. Extract descriptors for the representative features in both frames.
    f. Match descriptors between two ROIs.
    g. Filter matches based on background information and vehicle data.
    h. Identify the cluster of the features representing the tracked vehicle in the second frame.
    i. Collect and update the vehicle data.

j. Predict and update ROIs for the next frame pair using a constant acceleration model.

3. Output vehicle data.

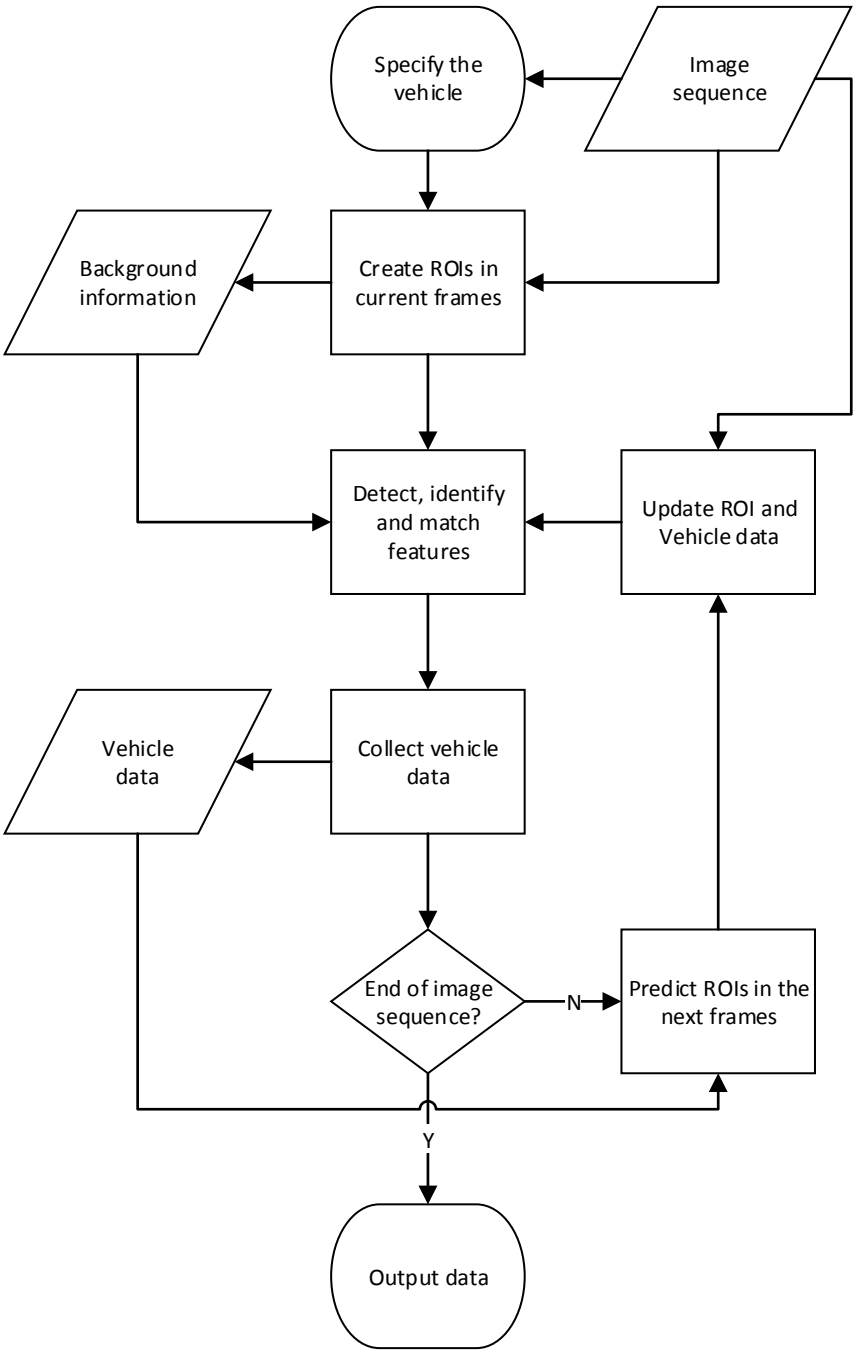
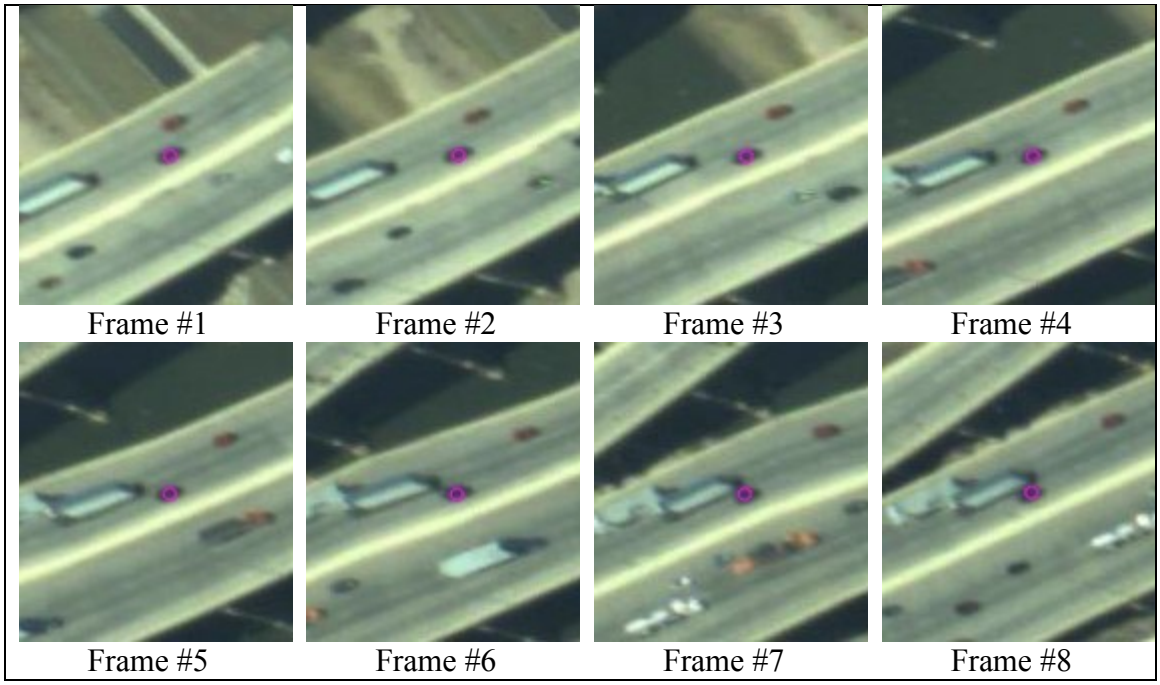


Figure 4 The basic procedure of the feature-based vehicle tracking framework

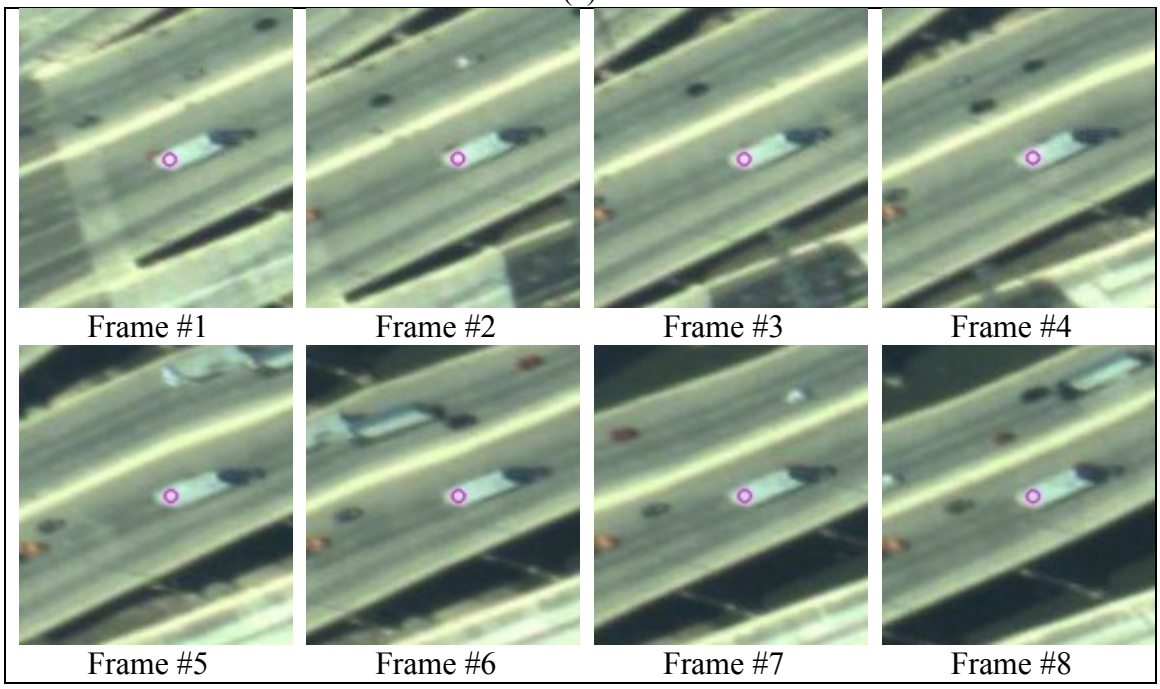
This framework was implemented in C/C++using OpenCV Version 3.0.0. The experimental results indicated that this approach was capable of tracking a specified

vehicle in the frame sequence of the PSS dataset. A simplified constant acceleration model was also employed in this implementation to estimate speed, acceleration and orientation for the purpose of predicting a vehicle's future location and ROIs to reducing the feature search and matching range. This framework can apply different methods for the feature detector, the feature descriptor, the feature matcher and the feature matcher filter in the framework, detailed configurations of which are provided in Appendix B. The performance depended on the detailed methods applied in this framework.

The tracking examples illustrated in Figure 5 (a) and (b) were based on a SURF detector and descriptor. As shown in Figure 5 (a), the black car (marked by a magenta circle) was tracked in the traffic flow on a bridge. The tracking lost as the track lost at the frame (not included in the figure) when the car reached a segment with different pavement and incorrectly jumped on another car in the opposite direction with similar appearance. In Figure 5 (b), the white truck was also successfully tracked, and it lost for pavement change as well. The performance of this specific application depends on numerous factors, including the contrast of vehicles and disruptors, among which the occlusion and seams are still challenging. Using SURF, this approach is capable of tracking most vehicles with stable appearance and background in many instances.



(a)



(b)

Figure 5 (a) Tracking a black car (8 adjacent frames are shown); (b) tracking a white truck (courtesy of PSS)

4.1.2 A Convolutional Neural Network Vehicle Detector

A supervised deep learning approach using CNN is presented in this section. The state-of-the-art CNN have already significantly outperformed existing handcrafted machine leaning approaches in many detection and classification problems. The purpose of this approach is to design and train a robust ConvNet that can accurately detect vehicles and output detections for tracking vehicles in aerial imagery.

The ConvNet was designed to detect whether an image patch contained a vehicle or not. The input image patches are 100×100 pixels. The ConvNet implemented in this research consists of eight layers, as shown in Figure 6:

1. The first layer is a convolutional layer. The input of the layer is a three-channel image (100×100×3). 25 different kernels (7×7×3) are used for convolution, respectively, and stride is set to 1. The outputs are 25 feature maps.
2. The second layer is a max-pooling layer. The kernel size is 3×3 and stride is set to 1, thus this layer doesn't subsample the input. The outputs are 25 pooled feature maps.
3. The third is a convolutional layer. 50 different kernels (5×5×25) are used to compute the convolution, respectively, and stride is set to 1. The outputs are 50 feature maps.
4. The fourth layer is a max-pooling layer. The kernel size is 4×4 and stride is set to 2, thus this layer subsamples the inputs. The outputs are 50 subsampled and pooled feature maps.

5. The fifth layer is a fully connected (inner product) layer. The output is a vector of 100 elements.
6. The sixth layer is a leaky rectified-linear unit layer.
7. The seventh layer is a fully connected (inner product) layer. The output a vector of 2 elements. One element represents the how likely the input image patch is a vehicle; and the other represents the how likely it is not a vehicle
8. The eighth layer is a leaky rectified-linear unit layer.

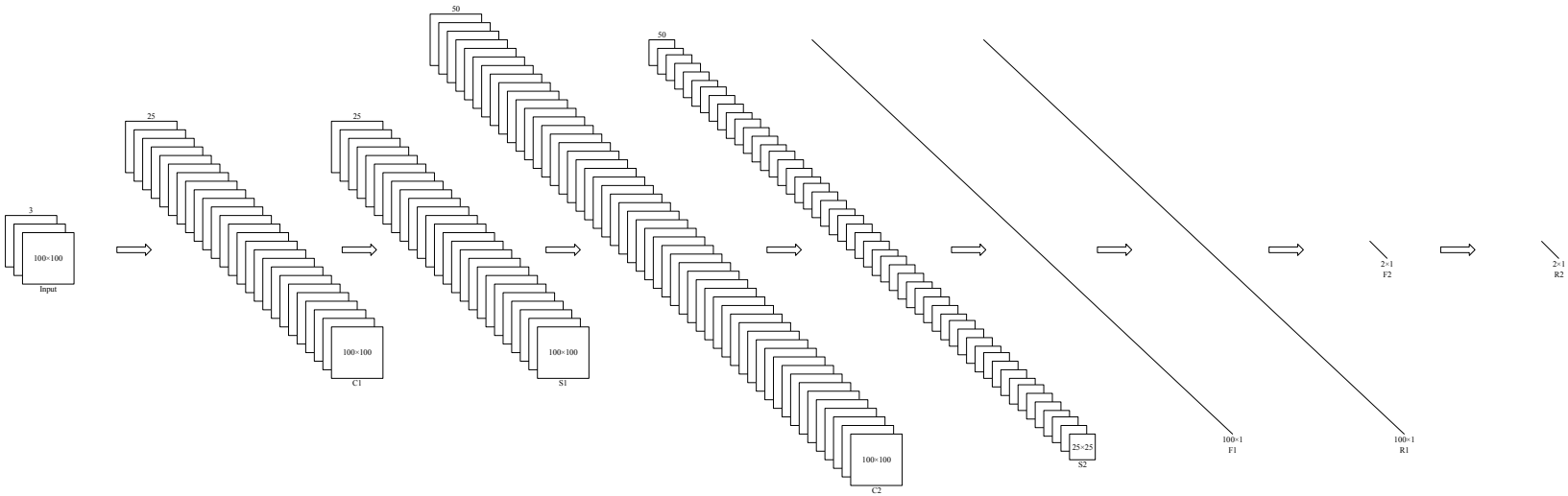


Figure 6 Design of the ConvNet used to detect vehicles in an image patch

The designed ConvNet was implemented with the Caffe library developed at University of California, Berkeley. The detailed design of the ConvNet is provided in Appendix C. The implementation procedure of this supervised approach consists of four basic steps:

1. Create training dataset. 1002 vehicle points and 604 non-vehicle points were manually labeled in the PSS dataset. The training data consisted of image patches generated from them. These image patches were randomly separated into training and test sets (80% training, 20% test).
2. Train the network with a training dataset. The training of the network took about 1 hour on an AMD 2.8 GHz processor, and resulted with the following performance on the test set: 95.7% accuracy, 94.5% precision, 96.0% recall (sensitivity), and 95.5% specificity.
3. Apply the trained ConvNet for vehicle detection. Each frame of the imagery data was split in multiple regions. The trained ConvNet run through each region and produce a score for each pixel indicating how likely a vehicle centered at the pixel.
4. Compute detections for each frame. Non-maximum suppression was used to find the peak value in the output score map. The final detection was determined by scores which were calculated by the local maximum in the score map. For the example region shown in Figure 7, all eight vehicles were detected, with three false positives. These false positives are removed in the subsequent tracking.

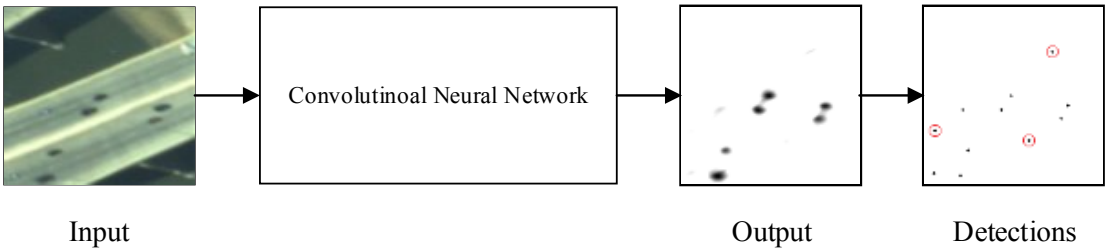


Figure 7 Trained ConvNet detects vehicles in a region from aerial imagery (courtesy of PSS)

4.1.3 Convolutional Neural Network with SURF

By combining the CNN-based detection with the feature-based tracking, the computer vision-based approach can achieve promising performance for vehicle detection and tracking in aerial imagery. This novel approach uses the trained ConvNet to detect vehicles in each frame and matches corresponding vehicles in a sequence of consecutive frames using SURF.

Experimental results indicated this approach can maintain tracks even the vehicles crossed seams or luminance changes, as shown in Figure 8. The accuracy of this approach is promising if the scenario is not complicated. However, the performance is dependent on false positives and false negatives in each pair of consecutive frames. As shown in Figure 9, the vehicles in blue circles were detected in a frame (top) but failed to be detected in the next frame (bottom), thus the tracks of these vehicles lost. On the other hand, false positives, like shadows or blots on pavement, were detected as moving vehicles so they lead to false tracks which are labeled with red circles.

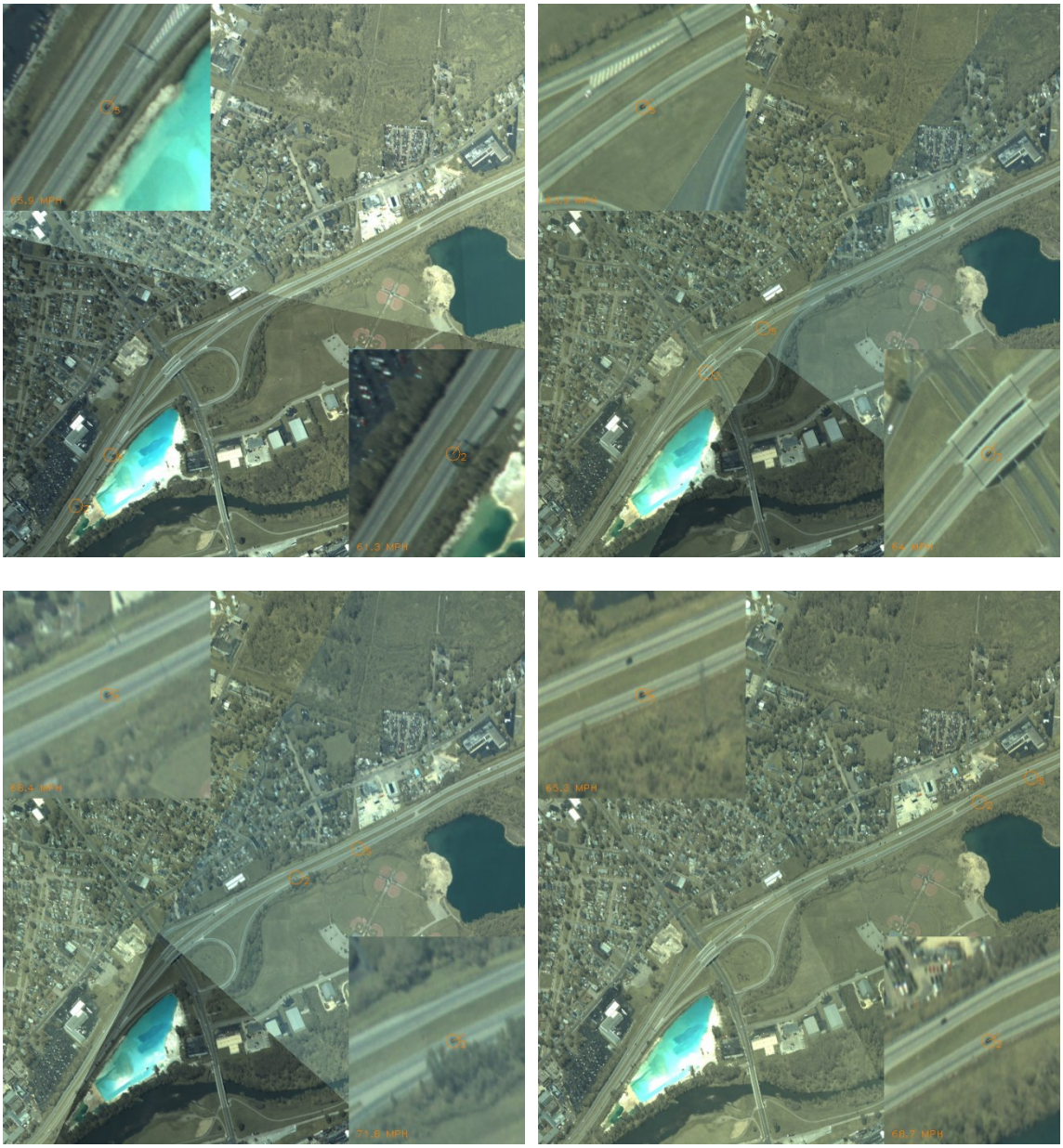


Figure 8 Tracking two vehicles across the seams and luminance changes (courtesy of PSS)

Figure 9 Detections from a frame (top) are filtered using the next frame (bottom) (courtesy of PSS)

4.2 Macroscopic Traffic Data Extraction

Diverse traffic data can be extracted from adequately processed (rectified or geo-registered) aerial imagery datasets. Macroscopic traffic data, including density, volume, average speed, and travel time, can be extracted by explicitly detecting and tracking each vehicle in the traffic flow. Sampling is an acceptable alternative to approximate macroscopic traffic data for those traffic data following normal distribution because of the central limit theorem (CLT). Thus, the implemented computer vision-based approach can theoretically provide reasonable estimates for macroscopic traffic data even it cannot guarantee 100% accuracy. Microscopic traffic data, including location, speed, acceleration (or deceleration), and trajectory of individual vehicles, requires highly accurate vehicle detection and tracking. Some complex traffic data, like car-following data, requires identifying and recording the

associations of interacting vehicles. This section will discuss the estimation of three macroscopic traffic measurements: density, speed and volume by processing the PSS dataset with the computer vision-based approach.

4.2.1 Density

Density data of a road segment requires a mask where the segment is defined and attributed. Underlying vehicles are detected using the trained ConvNet in the segment, and then the density data can be achieved based on the amount of detections in each frame. The trained ConvNet is capable of identifying vehicles with a promising accuracy in a single frame; however, the characteristics and the limited quality of the imagery in the PSS dataset makes it impossible to correctly identify 100% vehicles. By matching SURF features of detections in adjacent frames, the influence of false positives and false negatives can be significantly reduced. An example of this is shown in Figure 9, the trained ConvNet detected the vehicles in this segment for the current frame (top) and all detections are labeled as circles. SURF features of the detections are used to match them with corresponding detections in the next frame (bottom). Not all detections are able to be matched with SURF features (labeled with blue circles). Many kinds of false positives like stationary objects on roadside or pavement blots can be identified by the movement of the corresponding tracks (labeled with red circles). True positives can be used to measure speed (labeled with green circles). For the small segment in Figure 9, the CNN outputs 12 detections; 6 of them were tracked for collecting traffic data and 4 of them were stationary and determined to be false positives. Since the vehicle counts are heavily affected by imagery noise and the noise varies through

the frame sequences, an average smooth density can be achieved by filtering over a predefined period (e.g. 1 minute, 5 minutes, etc.).

4.2.2 Speed and Acceleration

Speed data of each individual vehicle is achieved by locating the vehicle in each frame and calculating the Euclidean distance of its shifting in consecutive frames in the rectified imagery with known resolution and frame rate. Only adequately rectified imagery dataset has unified and accurate resolution; the resolution of un-rectified imagery dataset can only be approximately estimated. Acceleration is measured by differentiating the speed. Figure 10 illustrates the sample speed and acceleration records of a tracked truck in the PSS dataset. The figure illustrates how the data are sensitive to the instability of the frame sequences. One-dimensional filters can be used for interpolation among neighbor frames and helping neutralize high-frequency vibration of the frame sequence. Improving preprocessing to better stabilize the imagery data with more accurate calibration and registration of the frames will smooth the speed over time and increase the accuracy of instantaneous speed with less of a need for filtering. Compared to speed, acceleration is even more sensitive to instability of the frame sequence. Thus, filtering is necessary in the case that the imagery is not perfectly stabilized and it can provide reliable and useful speed and acceleration for a time interval across multiple frames.

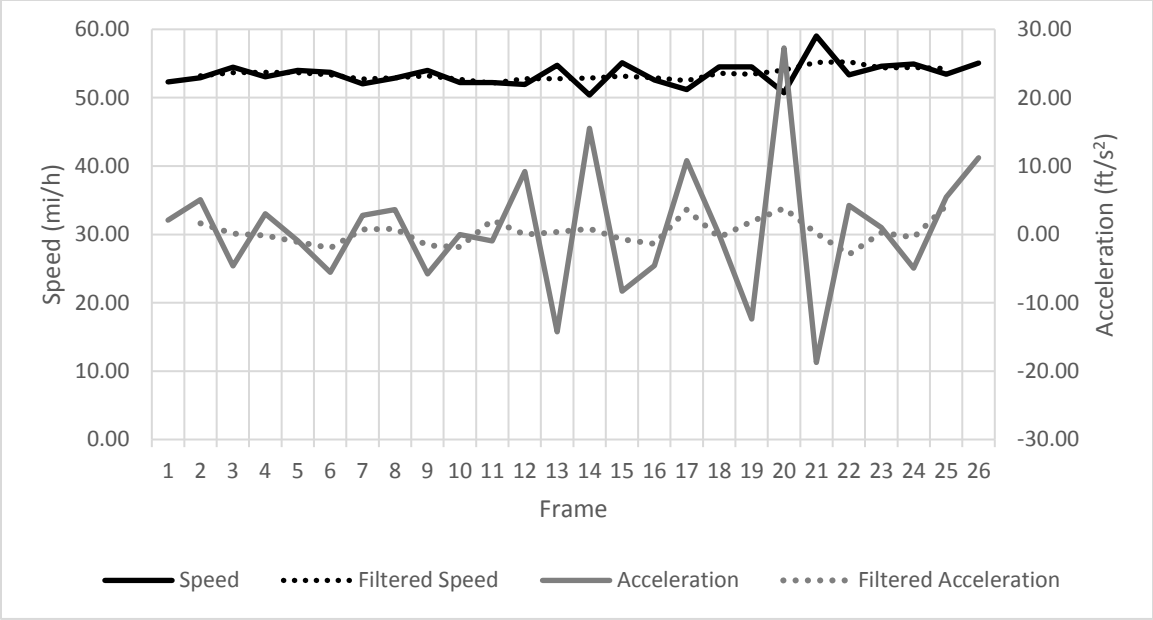


Figure 10 Sample speed and acceleration data of a randomly selected truck

The average speed of traffic flow on a road segment can be either calculated by averaging the speeds of all tracked vehicles in the traffic flow or approximately estimated by averaging the speeds of samples because of CLT. Since the detecting and tracking approaches cannot guarantee 100% accuracy, the former method is still a kind of sampling. For the segment in Figure 9, the average speed would be the average of the speeds of true positive detections in green circles.

4.2.3 Volume

Once average speed and density are determined, volume for a segment can be determined by simply multiplying density by speed. Theoretically, this method prefers space-mean speed, which is determined by travel time (number of frames) for each individual vehicle in the select segment, rather than time-mean speed.

A typical method for determining the traffic volume passing a point is to use a sensor such as an inductive loop detector which counts vehicles that pass over it. Similarly, video detection systems that use cameras mounted on the roadside predefine a virtual detector and count vehicles that pass over it. There are multiple problems to measure traffic volume passing a point with this method if applied to aerial imagery. First, instability of the frame sequence will require continuous recalibration of the frames and repeated relocation of the virtual detectors. Second, because of the low frame rate, vehicles will skip detectors unless its length exceeds the distance vehicles can move in a single frame. However, virtual detectors with a long length can be touched by multiple closely-spaced vehicles simultaneously resulting in undercounting. Furthermore, because of the low resolution, using a single frame to identify vehicles passing over a virtual detector will lead to errors due to closely-spaced vehicles and false positives in many instances.

4.3 Experimental Results and Evaluation

Experiments were implemented to evaluate overall performance of the proposed approach of CNN with SURF. Eight uninterrupted flow segments were selected from PSS dataset (see Figure 11) and processed for density, speed, and volume measurements using only a single pair of consecutive frames. Additionally, two of these segments were selected and processed using 50 consecutive frames. As shown in Figure 11, road segments on OH-4 (red/magenta), I75 (green), and US-35 (blue and cyan) were selected and manually masked.

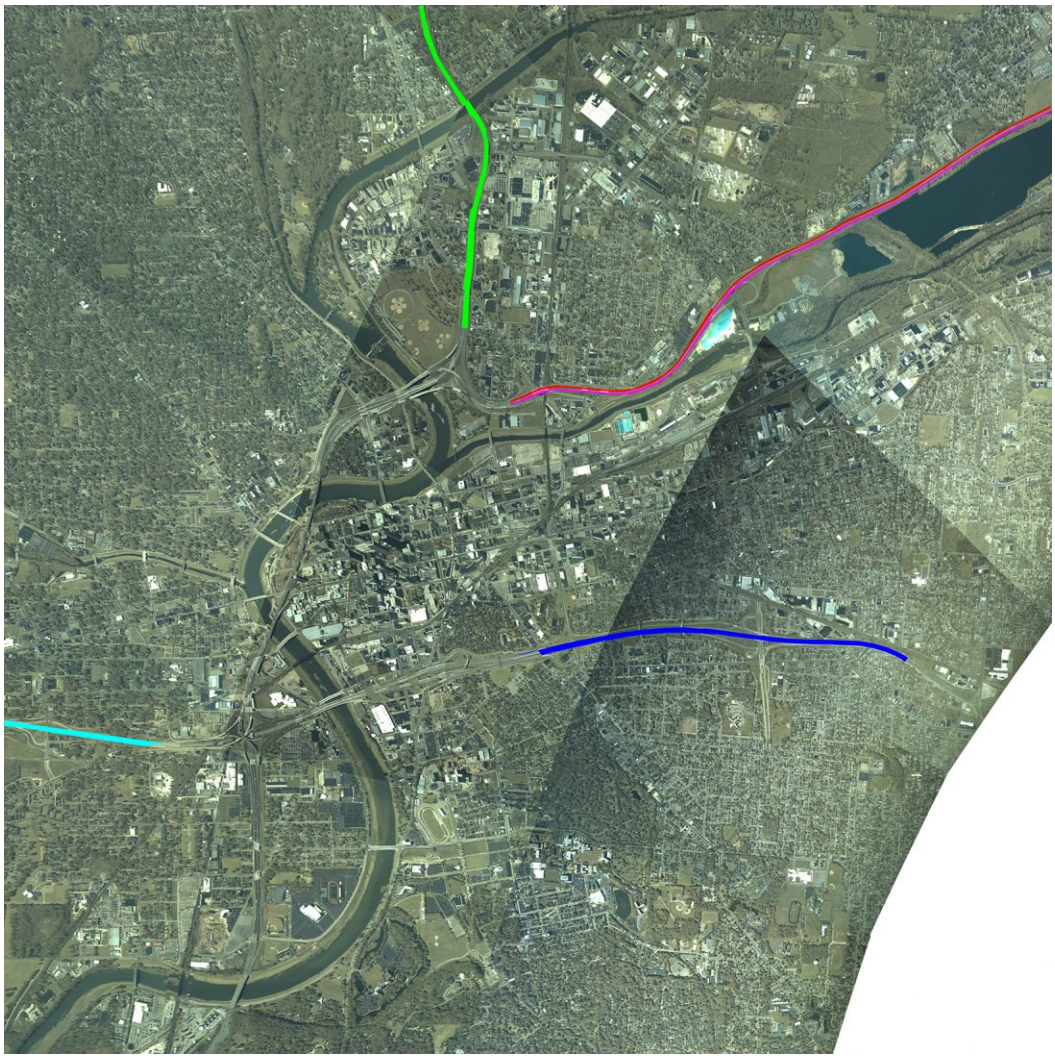


Figure 11 Eight segments selected and masked in the PSS dataset (courtesy of PSS)

Automatically collected measurements and manually measured ground truth are shown in Table 2. The ground truth density was obtained by manually counting vehicles in each frame, and the ground truth speed was obtained by averaging the Euclidean distance of the movement across frames of randomly selected vehicles, and volume was obtained by multiplying density and speed. By comparison of the estimates and ground truth, the

result indicated the density and speed data were accurate, thus leading to accurate estimates of level of service (LOS) and volume data.

Tracking across multiple frames produced reliable data. The performance of the proposed approach was powerful in counting vehicles, leading to accurate average density over time as shown in Figure 12. Similarly, other measurements were also accurate if averaged across multiple frames.

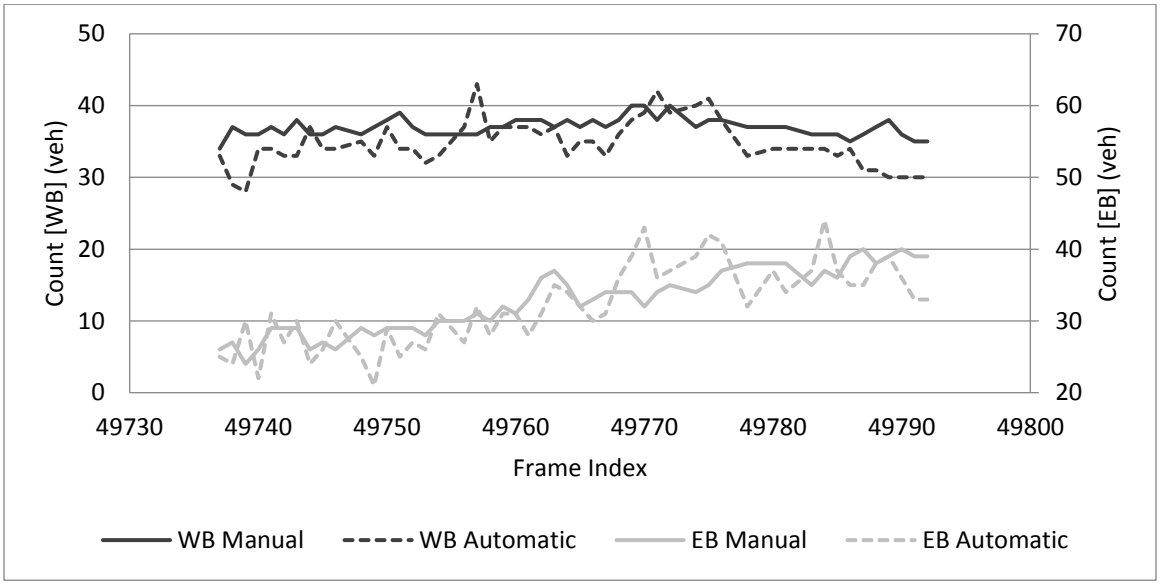


Figure 12 Automatically collected data vs. manually measured ground truth for vehicle counts over 50 frames on OH-4

Table 2 Traffic data measurements and evaluations

Road Segments	No. of Lanes	Length (mi)	Count (veh)		Speed (mph)		Density (veh/mi/ln)		Volume (veh/hr/ln)		LOS		Accuracy		
			A[a]	M[b]	A	M	A	M	A	M	A	M	Speed	Density	Volume
OH-4 (WB)[c]	2	3.05	30	36	66.9	69.8	4.9	5.9	329	412	A	A	95.8%	83.3%	79.8%
OH-4 (EB)[d]	2	3.04	33	36	62.8	66.8	5.4	5.9	341	396	A	A	94.0%	91.7%	86.2%
I75 (SB)	3	1.68	96	104	63.1	65.6	19.0	20.6	1202	1354	C	C	96.2%	92.3%	88.8%
I75 (NB)	3	1.68	60	76	58.8	68.2	11.9	15.1	700	1029	B	B	86.2%	79.0%	68.1%
US-35 (EB) [1]	4	0.83	23	30	57.3	50.0	6.9	9.0	397	452	A	A	85.4%	76.7%	87.9%
US-35 (WB) [1]	4	0.83	11	12	56.0	65.2	3.3	3.6	186	236	A	A	85.9%	91.7%	78.9%
US-35 (WB) [2]	3	0.88	54	54	61.5	65.7	20.5	20.5	1258	1345	C	C	93.6%	100.0%	93.6%
US-35 (EB) [2]	3	0.88	61	67	58.8	62.9	23.1	25.4	1359	1596	C	C	93.5%	91.0%	85.2%
OH-4 (WB)[e]	2	3.05	35	37	65.6	66.5	5.7	6.1	373	404	A	A	98.7%	93.7%	92.6%
OH-4 (EB)[e]	2	3.04	32	33	62.1	65.7	5.3	5.4	327	353	A	A	94.5%	98.2%	92.5%

a Automatically collected data
b Westbound
c Eastbound
d Manually collected ground truth
e Measurements based on 50 consecutive frames

4.3.1 Density

The automatically collected density data was accurate. Ground truth density was calculated by meticulously manual counting vehicles along the segments for each frame and taking the average. With only a single pair of frames, the average accuracy of estimates of density were 88.2% for all eight segments. However, when the density was measured and averaged across 50 consecutive frames, the accuracy was as high as 98.2% for this time interval, as shown in Figure 13. This result indicated that the implementation of proposed approaches was very reliable on collecting density data with a sequence of frames. Density was used to determine LOS which was 100% accurate for all segments evaluated due to the reliable density estimates.

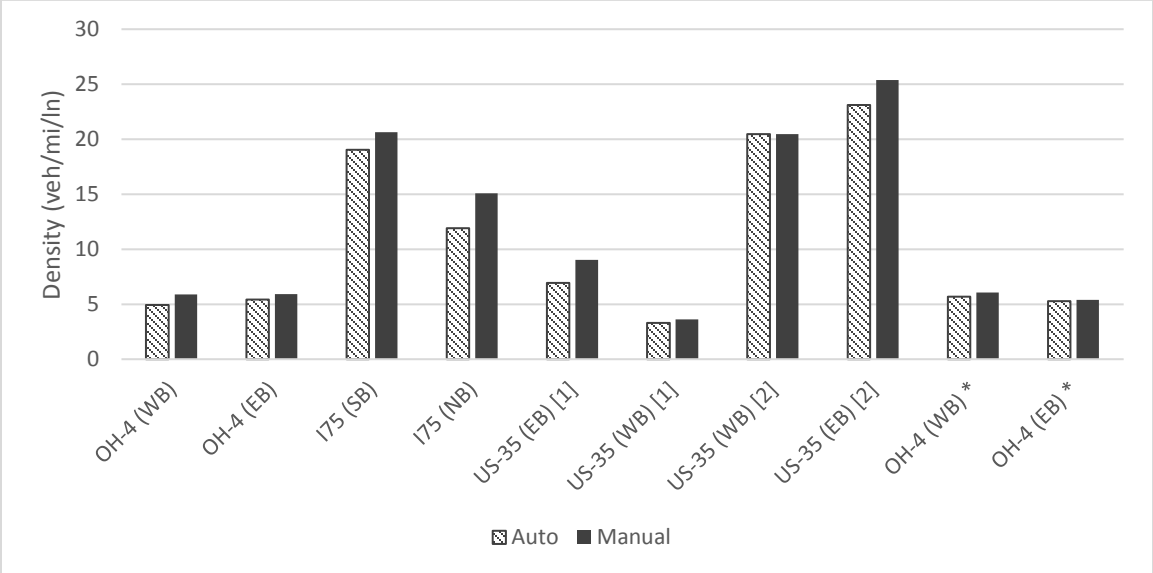


Figure 13 Automatically collected data vs. manually collected ground truth for density

4.3.2 Speed

Speed data was very sensitive to the instability of the aerial imagery. Ground truth speed was calculated by randomly sampling a subset of vehicles for each segment and manually tracking them over the frames and averaging the speeds of individual vehicles. The instantaneous speed based on the correspondence of a single pair of frames cannot always provide a useful result because of instability of the frame sequence (see Figure 9). However, the aforementioned discussion in Chapter 4.2.2 indicated the average speed for a time interval across multiple frames can be very accurate and useful with proper filtering. As shown in Figure 14, the accuracy of average speed across 50 frames was as high as 98.7%.

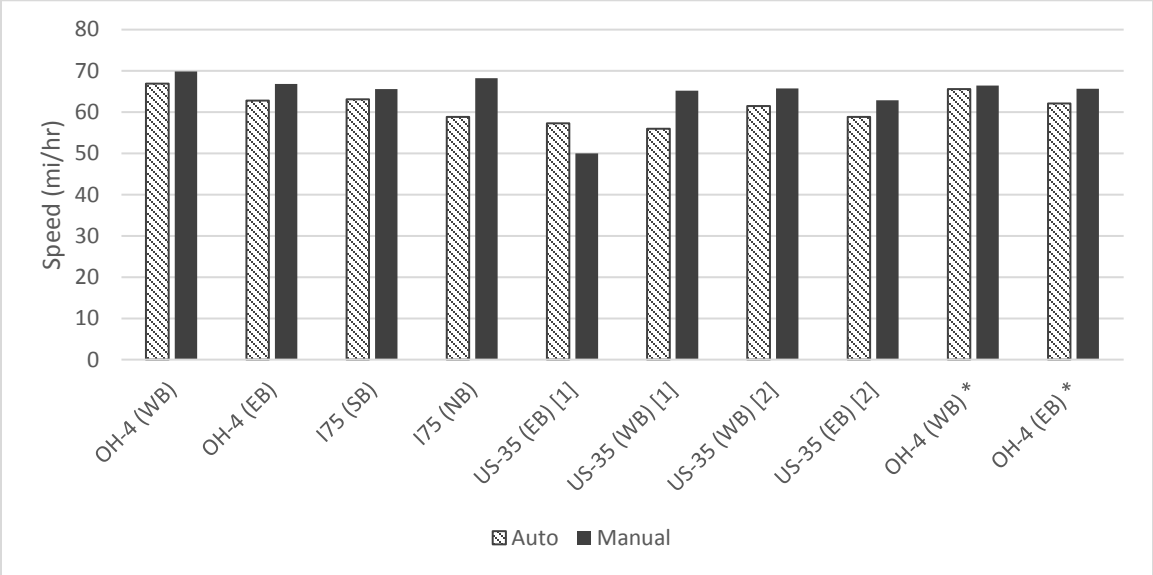


Figure 14 Automatically collected data vs. manually collected ground truth for speed

4.3.3 Volume

The estimates of volume data, calculated by density and speed, was found to be accurate. Even though the instantaneous speed was not reliable, the volume calculated with it still presented reasonable accuracy as shown in Figure 15. The average accuracy is 84.5% for all eight segments with a single pair of frames. The accuracy of volume was greatly improved and was as high as 92.6% with 50 frames.

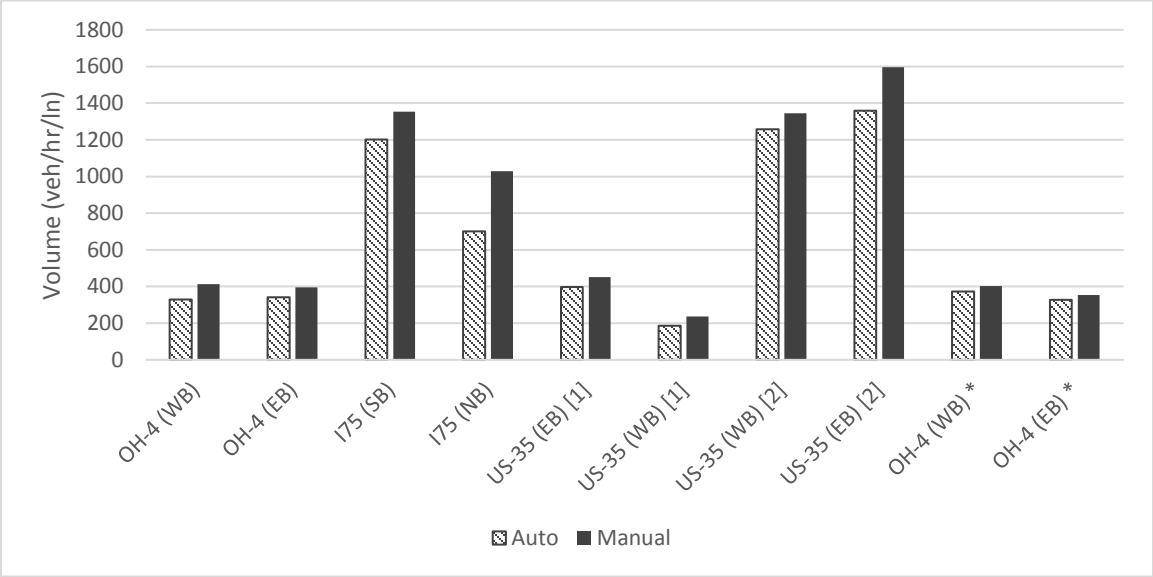


Figure 15 Automatically collected data vs. manually collected ground truth for volume

4.4 Discussion

Many of the challenges identified in Chapter III have been addressed in this chapter. This computer vision-based approach using CNN and SURF is robust enough to deal with the instability, seams, and inconsistent illumination of the images with limited effects on the

accuracy of the collected macroscopic traffic data, including density, speed, and volume. Future work will significantly improve the accuracy with more sophisticated algorithms and by fine-tuning trained ConvNet. The resolution, image size and frame rate are particular challenges addressed by the approach. One remaining challenge is the computation time: It takes 640 seconds on average using a single GPU to process a 25-mi^2 frame which means 25.6 second for a 1-mi^2 region. While this may seem slow, especially for the wide area aerial imagery from WAPS, there are seemingly endless applications for a dataset of this magnitude even if it takes days or weeks to process and create. For ITS applications, such as incident detection, more work is needed to make the computation efficient for real-time processing.

CHAPTER V

MULTIPLE HYPOTHESIS TRACKING APPROACH

5.1 Methodology

In order to improve the performance of vehicle tracking using quality-limited aerial imagery datasets with diverse characteristics, the author proposed and implemented a novel MHT approach to cooperate with the computer vision-based approach presented in Chapter IV. In this chapter, this MHT approach employing an innovative KAM structure was implemented and tested for tracking multiple vehicles, especially for tracking closely-spaced vehicles in saturated traffic flow. The research in this chapter also explored the impacts of frame rate, detection noise, number of lanes, and divided vs. undivided traffic flow on the performance of MHT as well as the effects of appearance information of vehicles and different weight functions.

The MHT approach was proposed and implemented as a module in the whole traffic surveillance system. The MHT module is designed to be a downstream component in the pipeline of vehicle detection and tracking (see Figure 16). This pipeline is very similar to the pipeline established in many existing radar systems. Those MHT modules receive detections from sensor (or detecting) modules as input and provide the tracks with highest probability (or score). Many existing radar systems have already proven the feasibility of MHT for data association in MTT tasks (Arambel, et al., 2004). However, the critical difference is that most radar systems are designed for detecting and tracking fewer targets

with much wider spacing between detections. Tracking vehicles in traffic flow is the long-standing "closely-spaced targets" problem that justifies the necessity of this research to validate the application of MHT in a traffic surveillance system.

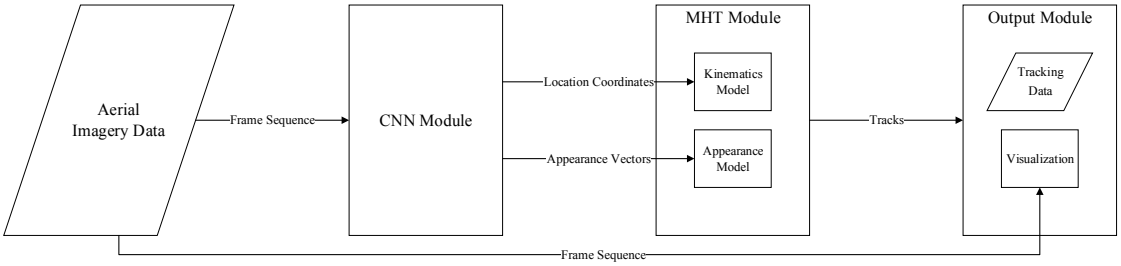


Figure 16 The pipeline of vehicle detection and tracking

5.1.1 Multiple Hypothesis Tracking

Functions of the MHT approach proposed and implemented in this research are to read the output of the trained ConvNet presented in Chapter IV (Zhao, et al., 2016) as input, solve the data association problems with MHT, and provide the most likely results. As shown in Figure 17, the basic pipeline of the MHT approach is to create a hypothesis tree based on the input of vehicle detections, and then evaluate and prune branches to maintain and search for the most likely hypothesis.

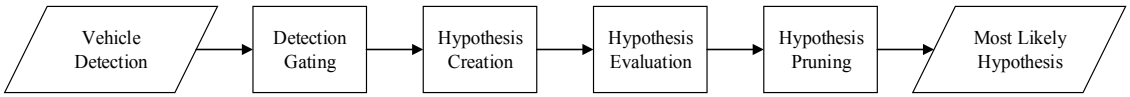


Figure 17 Basic pipeline of MHT for vehicle tracking

The MHT approach maintains and updates tracks by associations of detections in each time step (or frame). A set of consistent associations is a hypothesis. To be a valid MTT system, associations obey the following rules:

1. Each detection can only be associated with one track.
2. Each track can only be associated with one detection at each time step.

An example of typical association for MTT tasks is shown in Figure 17. P1 and P2 are predicted positions of two existing tracks, respectively; while D1, D2, and D3 are the vehicle detections in current frame. Detection gating restricts the acceptable area for associating detections with tracks. In this example, D1 can be associated with P1 but not P2, and D2/D3 can be associated with either P1 or P2. Thus, a possible hypothesis can be one where D1 is associated with P1, D2 is associated with P2, and D3 begins a new track. For the case shown in Figure 18, ten hypotheses are potentially feasible in total. All hypotheses are listed in Table 3 that illustrate all the possible association of tracks and detections.

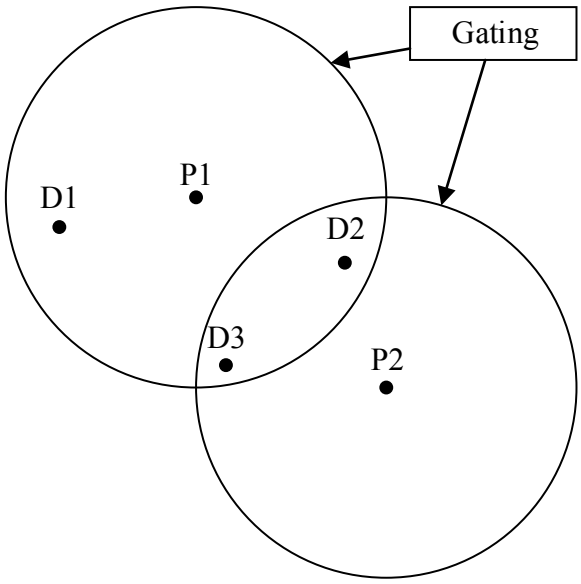


Figure 18 An example of typical data association

Table 3 All potential hypotheses for the conflict situation in the example

Hypothesis	D1	D2	D3
H1	P1	P2	New
H2	P1	New	P2
H3	P1	New	New
H4	New	P1	P2
H5	New	P1	New
H6	New	P2	P1
H7	New	P2	New
H8	New	New	P1
H9	New	New	P2
H10	New	New	New

A hypothesis tree ideally consists of all feasible hypotheses in all frames. A feasible hypothesis is presented by a node in the tree, and the children of this node are the all feasible hypotheses in the next frame deriving from their parent hypothesis. The depth of

the tree indicates the total number of frames and the depth of each node indicates in which frame the corresponding hypothesis is. In a frame, the most likely hypothesis of the tracks is the hypothesis with highest probability (or score) for all associations, and all corresponding hypotheses in previous frames can be traced back by searching the parent nodes. The total number of hypotheses in a hypothesis tree increases exponentially when the number of detection or the number of frames increases. Thus, maintaining and tracing back all feasible hypotheses are not computationally practical. All existing MHT implementations only maintain a certain number of children for each node by pruning low probability (or score) hypotheses and limiting the depth of tracing back. An example of hypothesis tree is shown in Figure 19, the ten nodes at k+1 frame correspond to those ten hypotheses in the example shown in Figure 18.

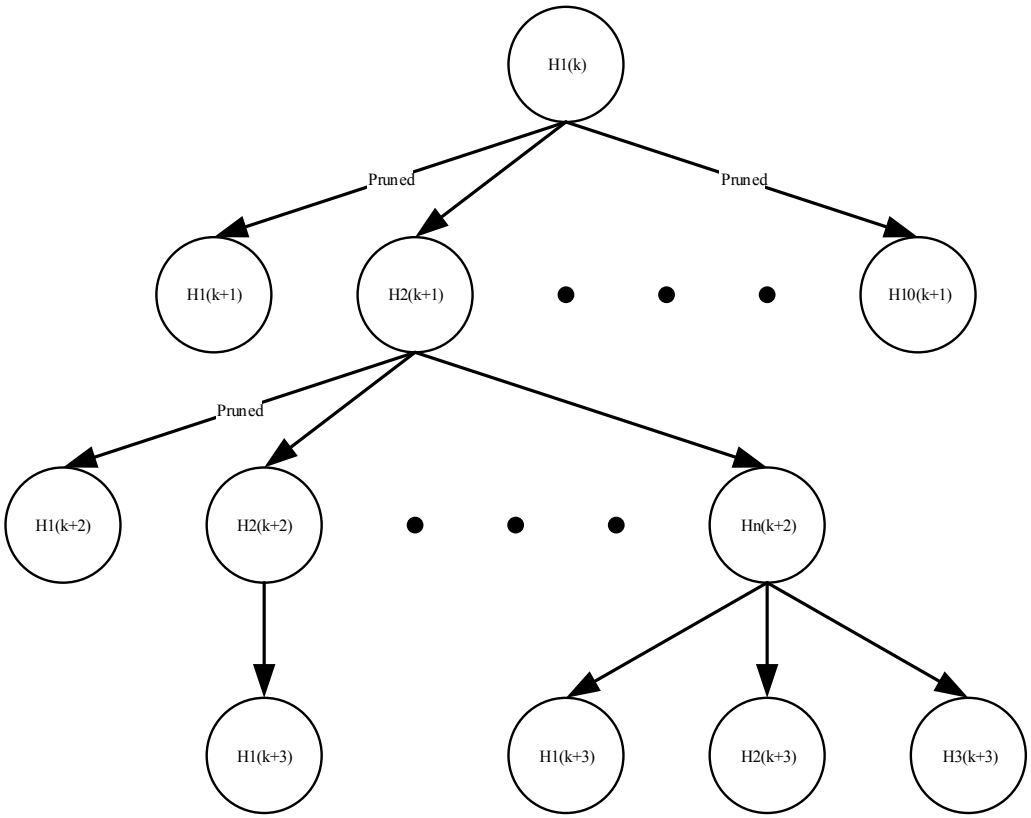


Figure 19 A example of hypothesis tree

The MHT approach was implemented using Multiple Hypothesis Library (MHL) (Antunes, n.d.), which is a public library for basic MHT implementation. To fulfill the requirements in this research, besides extensive coding, many parameters were fine-tuned for satisfactory performance in cooperation with the Kalman filters for kinematics model and appearance model, which will be discussed in Chapter 5.1.2. Those critical parameters included the max number of leaves to maintain, the max depth to trace back, the undetected time limit to label a track undetected, and the max number of children to maintain for each node. The values used in this research were: number of leaves to maintain was set to 50,

max depth to trace back was set to 6, undetected time limit was set to 3, and max number of children for each node was set to 6.

5.1.2 Kinematics and Appearance Models

Many conventional MHT approaches employ a single Kalman filter for maintaining and updating tracks by estimating the states (typically the kinematic measurements like position, velocity, and acceleration) of tracks. A further technique interacting multiple model (IMM), which use multiple Kalman filters, was widely accepted for the performance on tracking maneuvered targets (Blackman, 2004). IMM applies different filter models parallelly and each of them are specifically selected for different types of maneuvers. The combined state estimates and covariance can be computed by either switching among the outputs of different Kalman filters or the weighted composition of them.

KAM proposed and implemented in this research is an innovative structure that takes the advantage of the visual information of detected vehicles in aerial imagery and the image classification capability of CNN. Compared to IMM in which multiple filter models share the same input of kinematics information, KAM contains an additional pipeline, by which the appearance model process the input of the appearance data of detected vehicles, parallel to the pipeline for the kinematics model (see Figure 20). Because the kinematics state of a vehicle is not correlated to its appearance state (except the limited influence from illumination, camera angle, shadow, etc., which is considered by the measurement noise in the model), kinematics vector and appearance vector can be processed by two pipelines separately. An additional reason for separated processing is to speed up the computation

by computing two small covariance matrices parallelly instead of one large covariance matrix. Of course, they can be combined in a single vector (either assuming correlated or not), and can be processed by a single combined pipeline, but the computational complexity will be increased for the combined large covariance matrix.

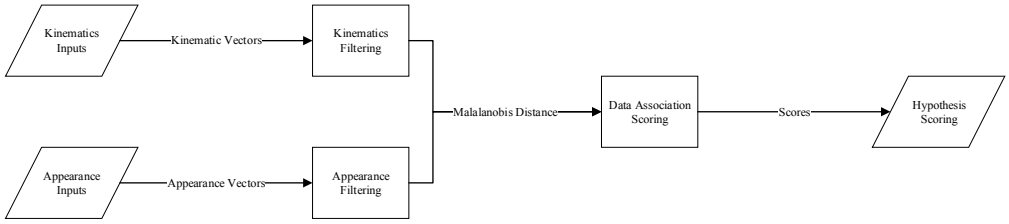


Figure 20 Kinematics and appearance models

5.1.3 Kalman Filters for KAM

Both of kinematics model and appearance model use similar Kalman filters to predict state estimate, covariance estimate, and detections.

The prediction of state estimate,

$$\hat{x}_{(k|k-1)} = F_k \hat{x}_{(k-1|k-1)} + w_{k-1} \quad (1)$$

where $\hat{x}_{(k|k-1)}$ is the state estimate of the ground truth x at time step k given detections up to and including at time step $k-1$; F_k is the state transition model applied to the previous state estimate; and w_{k-1} is process noise.

The prediction of covariance estimate,

$$P_{(k|k-1)} = F_k P_{(k-1|k-1)} F_k^T + Q_k \quad (2)$$

where $P_{(k|k-1)}$ is the error covariance matrix; and Q_k is the covariance matrix of process noise w_k.

The prediction of detection,

$$z_k = H_k x_k + v_k \quad (3)$$

where z_k is the predicted detection of ground truth x_k at time step k; H_k is the detection measurement model mapping state space to detection space; and v_k is detection noise.

The Kalman filter for the kinematics model can use a constant velocity model because the duration of the time step of aerial video is short. Even for 1 Hz imagery data, the variation of velocity in the duration of a time step is very limited. The prediction of state estimate, the prediction of covariance estimate, and the prediction of detection follow Equation (1), (2), and (3), respectively. In this kinematics model, the ground truth kinematics state x_k and its estimate $\hat{x}_k$ are in the form of a vector (length of 4) containing x and y positions, and x and y velocities; the predicted detection z_k is a vector (length of 2) containing x and y positions. The transition model,

$$F_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where Δt is the interval of time step; and process noise

$$w_{k-1} = \begin{bmatrix} \Delta t^2/2 \\ \Delta t^2/2 \\ \Delta t \\ \Delta t \end{bmatrix} a_k \quad (5)$$

where a_k is a random Gaussian acceleration with a standard deviation of 3.28 ft/s^2 and a mean of 0 ft/s^2. The measurement model,

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

; and v_k is a 2×2 identity matrix where elements are scaled by random Gaussian variables with a standard deviation of 1.64 ft and a mean of 0 ft.

The Kalman filter for appearance model can use a consistent appearance model because the variation of appearance is very limited in the duration of a time step. The prediction of state estimate, the prediction of covariance estimate, and the prediction of detection follow also Equation (1), (2), and (3), respectively. In this appearance model, the ground truth appearance state x_k, its estimate $\hat{x}_k$, and predicted detection z_k are in the form of a vector describing an image patch centered at the corresponding vehicle in the aerial imagery. The input appearance data can be the output of a ConvNet similar to the ConvNet designed in Chapter IV. They were designed to be in the exactly same format for the compatibility. Additionally, F_k and H_k are identity matrices; w_{k-1} is aero matrix; and v_k is an identity matrix where elements are random Gaussian variables with a standard deviation of 0.55 and a mean of 0.

Control-input model was not included in KAM because the information is not available from aerial imagery data. The Kalman filters in KAM provides a vehicle's state estimate as a Gaussian distribution; in other words, it not only provides the predicted position, velocity, and appearance of each tracked vehicle, but also the certainty of those

predictions in the form of a covariance matrix. The Apache Commons Mathematics Library was used to implemented Kalman filters in KAM

5.1.4 Mahalanobis distance-based Scoring

KAM uses Mahalanobis distance (Mahalanobis, 1936) to measure the variation between the distribution of a predicted detection and the actual detections from inputs for kinematics model and appearance model, separately. The Mahalanobis distance D follows:

$$D(y,z) = \sqrt{(z-y)^T P^{-1}(z-y)} \quad (7)$$

where y is the actual detection from inputs. Then, KAM processes a gating based on the weighted sum of Mahalanobis distance from both models to remove outliers. The probability (or score) p of an association follows:

$$p = e^{-(w_1 D_1^2 + w_2 D_2^2)} \quad (8)$$

where D_1 and D_2 are the Mahalanobis distances from kinematics model and appearance model, respectively; and w_1 and w_2 are corresponding weights. The probability (or score) of an association indicates the likelihood of the association regarding both kinematics and appearance.

The design of weight functions can be various. In this research, two weight functions were implemented and tested. The first function was a normalized weight function. In this function, w_1 and w_2 were dependent on the appearance weight ratio r_A:

$$w_1 = 1/(1 + r_A) \quad (9)$$

and

$$w_2 = r_A/(1 + r_A) \quad (10)$$

The second function was an unnormalized weight function. In this function, w_1 and w_2 were also dependent on the appearance weight ratio r_A:

$$w_1 = \begin{cases} 1, & if\ r_A < 1 \\ 1/r_A, & if\ r_A \geq 1 \end{cases} \quad (11)$$

and

$$w_2 = \begin{cases} r_A, & if\ r_A < 1 \\ 1, & if\ r_A \geq 1 \end{cases} \quad (12)$$

The probability (or score) of a hypothesis, which indicates the likelihood of this hypothesis, can be achieved by averaging the probabilities (or scores) of all valid associations contained in this hypothesis. It is used to evaluate and prune the nodes in the hypothesis tree in order to maintain and search for the most likely hypothesis.

5.2 Experiments

The experiments designed and implemented in this chapter consisted of two parts. The first experiment was to apply the MHT approach with kinematics model only (appearance weight ratio was set to 0) to numerous datasets of vehicle detection generated with different frame rates (1 Hz, 5 Hz, and 10 Hz), different detection noise (0%, 5%, and 10%), and different scenarios (2-lane and 4 lanes; divided and undivided traffic flow), respectively, for the purpose of evaluating the MHT approach with kinematics model only and revealing the impacts of frame rate, detection noise, number of lanes, and divided vs. undivided

traffic flow on the performance. The second experiment was to apply the MHT approach with KAM to numerous datasets of vehicle detection generated with different frame rates and detection noise using different appearance weight ratios (0, 0.1, 0.5, 1, 2, and 10) and weight functions (normalized vs. unnormalized), for the purpose of evaluating the MHT approach with KAM and revealing the effects of appearance information on the performance.

5.2.1 Synthetic Data

Although the PSS dataset was used in the experiments in Chapter IV (Zhao, et al., 2016) and successfully proved the promising accuracy of macroscopic traffic data collection using aerial video, aerial imagery datasets (see Table 1) are not ideal for the experiments in this chapter. The experiments focused on MHT require saturated traffic flow to study the "closely-spaced targets" problem, as well as controllable frame rate, detection noise, and traffic configurations for the analysis of those concerned factors.

To comprehensively study all concerned factors, experiments were implemented with synthetic vehicle detection data. VISSIM (a microscopic traffic simulation software) was employed to generate the vehicle detection data as it has one of the most accepted driving behavior models. Using synthetic data generated by VISSIM instead of precomputed data generated by the CNN module in the experiments is sensible for several reasons:

1. The MHT module is designed to read vehicle detection data (coordinates of the location of each detected vehicle in each frame) as input and the CNN module is

designed to provide vehicle detection data in exact same format, both of which are compatible with vehicle records (coordinates of the location of each simulated vehicle in each time step) that VISSIM can generate. Detailed information about the format and parsing vehicle records are provided in Appendix D.

2. Two datasets (no matter generated by simulation or by processing aerial imagery) can be equivalent in term of MHT computation due to the scale-agnostic property of MHT, which will be discussed in Chapter 5.2.2.
3. Synthetic data can provide valid traffic information for MHT computation as far as VISSIM can produce valid traffic simulation.
4. Simulation parameters can be manipulated to generate specific synthetic data for various requirements; while precomputed data determined by processing aerial imagery cannot be manipulated.
5. VISSIM can record vehicle ID, which can provide the ground truth information for reliable evaluation; while it is impractical to manually label ground truth in aerial video.

Four 1-mile uninterrupted urban road segments were created. Two have one lane in each direction, and the other two have two lanes in each direction. Both pairs contain a road with a median and a road without a median. The width of the lanes was set to 12 ft and the width of the median was set to 12 ft. The volume of vehicle input was set to 1300 veh/hr/ln. The duration of the simulation was set to 300 seconds.

The frame rates of investigated public and commercial aerial imagery dataset are at least 1 fps (see Table 1), and few high frame rate imagery datasets are available. However,

more high frame rate imagery datasets will be available with the advancement of optical sensor and data storage technologies. Thus, to generate vehicle records simulating the vehicle detection data extracted from aerial imagery with different frame rates, the simulation time step was set to 1, 5, and 10 steps per second to correspond to 1, 5, and 10 frames per second, respectively. The detection performance of the trained ConvNet presented in Chapter IV is: 95.7% accuracy, 94.5% precision, 96.0% recall (sensitivity), and 95.5% specificity (Zhao, et al., 2016). Thus, to generate the detection noise caused by the inaccuracy of the detector, 0%, 5%, and 10% false positive were created by randomly adding detections near existing the ground truth, and the same percentage of false negative were created by randomly removing ground truth from the detection data.

To apply and evaluate the MHT approach with KAM, a vector of float values was appended to each vehicle detection record as its appearance information since a trained ConvNet can output a vector as the descriptor of the input image. The vector length can be flexible based on detailed implementation, and 5 was used in these experiments. Based on the investigation on the imagery datasets listed in Table 1, 10 classifications were defined for the fact that many vehicles have similar appearance, and then corresponding appearance vectors were attached in the synthetic data. A uniformly distributed random float number within the range from -1 to -1 was initialized in each element in the appearance vector to define the appearance of the vehicle. The appearance variation following a normal distribution with a mean of 0 and a standard deviation of 0.1 was added to each element in the appearance vector to simulate the instability of the appearance of the same vehicle in different frames. Additionally, the measurement noise following a normal distribution with

a mean of 0 and a standard deviation of 0.3 was added to each element in the appearance vector to simulate the detection inaccuracy. To explore the effects of appearance information of vehicles on the performance of the MHT approach with KAM, the appearance weight ratios (the weight of the output of appearance model over the weight of the output of kinematics model) were set to 0, 0.1, 0.5, 1, 2, and 10.

5.2.2 Scale-Agnostic Property of MHT

In order to comprehensively research, test, and evaluate the MHT approach, the scale-agnostic property of MHT was originally induced. To study the "closely-spaced targets" problem of tracking vehicles in traffic flow, only one of the following parameters has to be varied: frame rate, vehicle density, or vehicle speed. In terms of MHT computation, varying any one parameter can be equivalent to varying other parameters. The reason is that MHT is scale-agnostic. Mathematically, two datasets are equivalent to MHT computation if they satisfy

$$(D_1 \oslash S_1)f_1 = C = (D_2 \oslash S_2)f_2 \quad (13)$$

for each time step, where $\oslash$ is the element-wise division (Hadamard division); D is the density matrix consists of the distance d_{ij} between each pair of objects i and j for all n objects in a time step,

$$D = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{pmatrix} \quad (14)$$

and S is the speed matrix consists of the speed s_i of each object i in a time step,

$$S = \begin{pmatrix} s_1 & \cdots & s_1 \\ \vdots & \ddots & \vdots \\ s_n & \cdots & s_n \end{pmatrix} (15)$$

Additionally, f is the sample rate and C is a constant matrix. Briefly, two datasets are equivalent regarding MHT computation if they can get the same matrix C. The unit of elements in C is dimensionless.

For example, tracking two ants, 5 inches apart, moving 1 in/s is exactly the same as tracking two cars, 500 feet apart, moving 100 ft/s; and tracking an ant moving 1 mi/hr at 60 frames/hr is equivalent to tracking a car moving 1 mi/min at 60 frames/min. Because of this scale-agnostic property, a frame rate change is equivalent to changing either vehicle speeds or traffic density. To demonstrate, suppose there are two cars, 100 feet apart, moving 100 ft/s, with a sample rate of 1 frame/s. Doubling just the frame rate would result in the cars still going 100 ft/s and separated by 100 feet, but with a sampling rate of 2 frames/s. However, if one looks at the situation in units of "half seconds" (hs, 2 hs = 1 s), then we have the cars going 50ft/hs separated by 100ft at a frame rate of 1 frame/hs. So, in terms of MHT computation, it is equivalent to halving the vehicle speed. In addition to using "half seconds", one could use "half feet" (hft, 2 hft = 1ft). This would mean the cars are going 100 hft/hs, separated by 200 hft, with a sample rate of 1 frame/hs. In terms of MHT computation, doubling frame rate speed is equivalent to halving the vehicle density. To study the “closely-spaced targets” problem of tracking vehicles in traffic flow, only one of the following parameters has to be varied: frame rate, vehicle density, or vehicle speed. In this research, frame rate was chosen as the parameter to vary because it can be easily manipulated.

5.2.3 Metrics and Statistics

For any MTT problem, performance evaluation is critical, but not straight forward (Gorji, et al., 2011). It is not possible to estimate the performance of a tracker based on a single metrics. A tracker can provide inaccurate results while some metrics indicate satisfactory performance (Coraluppi, et al., 2006). Thus, five metrics were specifically selected in this research: percentage of valid associations (NVA%), percentage of false associations (NFA%), percentage of missed detections (NMD%), average number of swaps (ANST), and average track continuity (ATC). Since percent metrics can provide clear comparison for detection data generated with different frame rates, they were used for evaluation but actual value of them, which is used in many research (Gorji, et al., 2011). Three other statistics were also collected for overall information: number of vehicles (NV), number of detections (ND), and number of tracks (NT). Table 4 provides detailed definitions of them.

Table 4 Three statistics and five metrics selected

NV	Total number of ground truth vehicles (or target).
ND	Total number of ground truth detections (one for each vehicle in every frame it was in).
NT	Total number of tracks produced by the MHT. A track is an estimated trajectory of a ground truth vehicle.
NVA%	Number of valid associations over total number of ground truth detections (NVA/ND). An association is valid if the prediction is assigned to one and only one ground truth detection and that assigned ground truth detection is not associated with any other prediction.
NFA%	Number of false associations over total number of ground truth detections (NFA/ND). An association is false if the prediction is not assigned to any ground truth detection.
NMD%	Number of missed detections over total number of ground truth detections (NMD/ND). A missed detection is a ground truth detection that is not associated to any prediction.
ANST	Average number of swaps target per ground truth vehicles (NST/NV). This is the average number of swaps between different tracks on what should be a single ground truth vehicle.

ATC	Average track continuity per ground truth vehicle (TC/NV). Track continuity is a metric for each ground truth vehicle that measures the average proportion of its trajectory is covered by each associated track (Gorji, et al., 2011).

The computation of these metrics and statistics involves extensive coding and is not real-time. The performance evaluation of MHT is dependent on the ground truth information provided by vehicle ID recorded by VISSIM.

5.3 Experimental Results and Analysis

Experiments were implemented to evaluate the performance of implemented MHT approach, and reveal the impacts of frame rate, detection noise, number of lanes, and divided vs. undivided traffic flow as well as the effects of appearance information of vehicles.

For the first experiment, five metrics and three statistics are shown in Table A1 for the experimental results of MHT with kinematics model only (appearance weight ratio was set to 0) using a Mahalanobis distance-based gating of 3×std (three times the standard deviation derived from the error covariance matrix). The metrics and statistics are also shown in Table A2 for the results using a gating of 7×std.

Since the MHT approach presented best overall performance for the 2-lane divided road segment in the first experiment, this scenario was selected for the second experiment. Five metrics are shown in Table A3 for the experimental results of MHT with KAM using a Mahalanobis distance-based gating of 3×std and the unnormalized weight function. The metrics are also shown in Table A4 for the results using a gating of 7×std and the

unnormalized weight function. Additionally, the metrics are shown in Table A5 for the results using a gating of 7×std and the normalized weight function.

5.3.1 MHT with kinematics model only

From the metrics and statistics shown in Table A1 and A2, the following conclusions can be drawn for the performance of the MHT approach with kinematics model only:

1. High frame rate can significantly reduce the negative impact of noise on the performance. Even though the amount of noise is proportional to the number of frames or detections, the performance is significantly improved with high frame rate data for all metrics (see Figure 21). Regarding scale-agnostic property, increasing frame rate is equivalent to enlarging the spacing between targets or reducing the speed of targets which can avoid the "closely-spaced targets" problem.

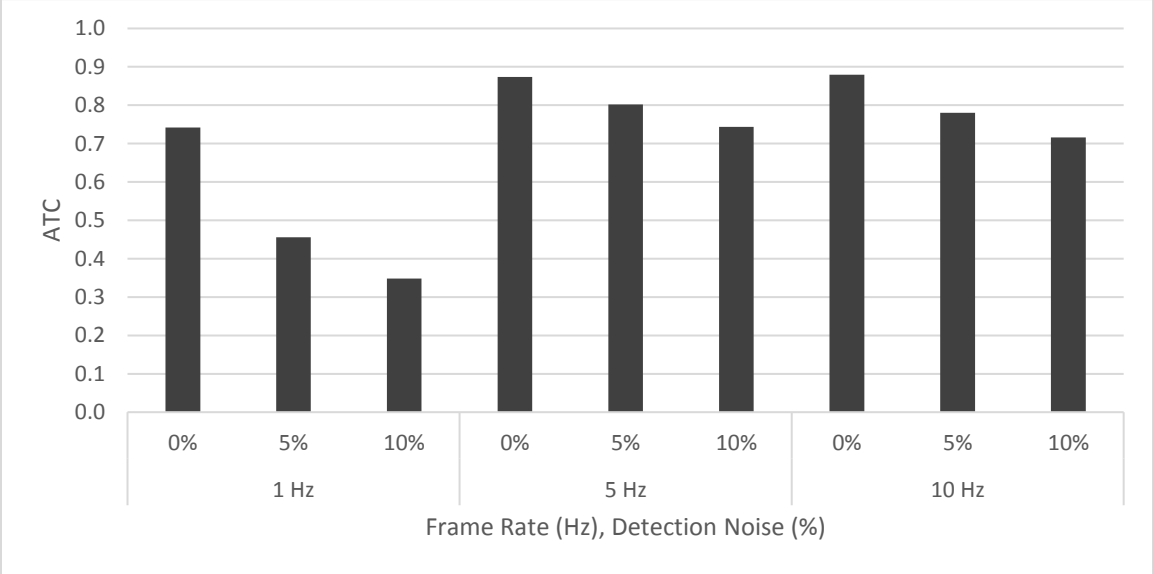


Figure 21 ATC for MHT (2-lane, divided, 7×std gating)

2. High percent noise can lead to significant negative impact for 1 Hz data, while the impact is less significant for 5 Hz and 10 Hz data. The vulnerability of low frame rate data to noise is consistent with previous conclusion that high frame rate data is robust regarding noise (see Figure 21).
3. The performance with divided traffic flow does not present significant differences compared with undivided traffic flow for both 2-lane road and 4-lane road (see Figure 22).

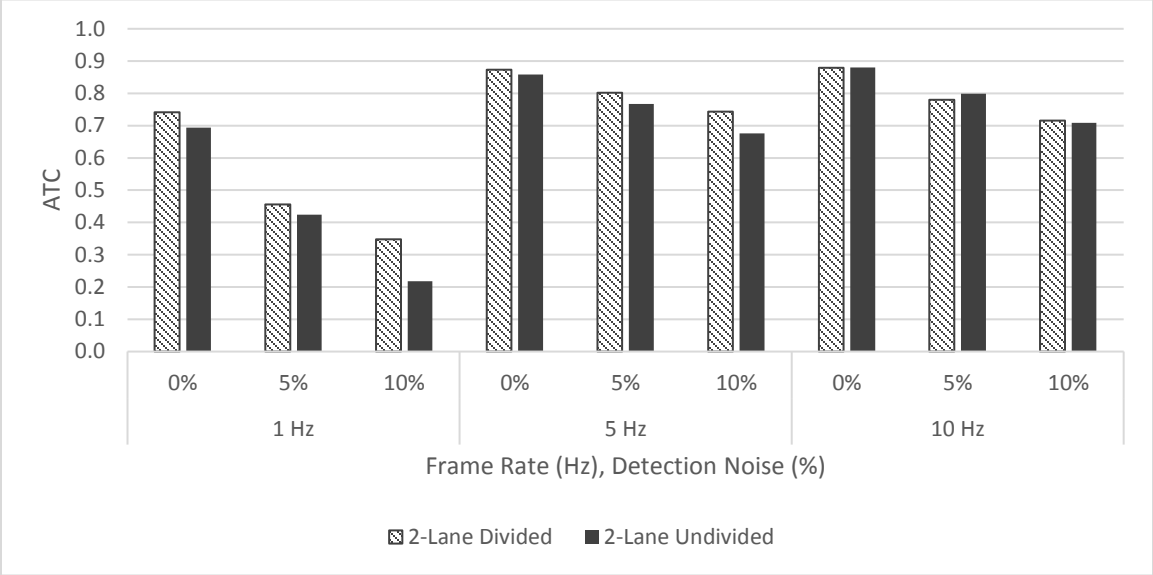


Figure 22 Divided vs. undivided (2-lane, 7×std gating)

4. All metrics indicate that more lanes lead to worse performance at low frame rate. The change is not significant at high frame rate (see Figure 23). The vulnerability of low frame rate data to density is consistent with previous conclusion that increasing frame rate can avoid the “closely-spaced targets” problem.

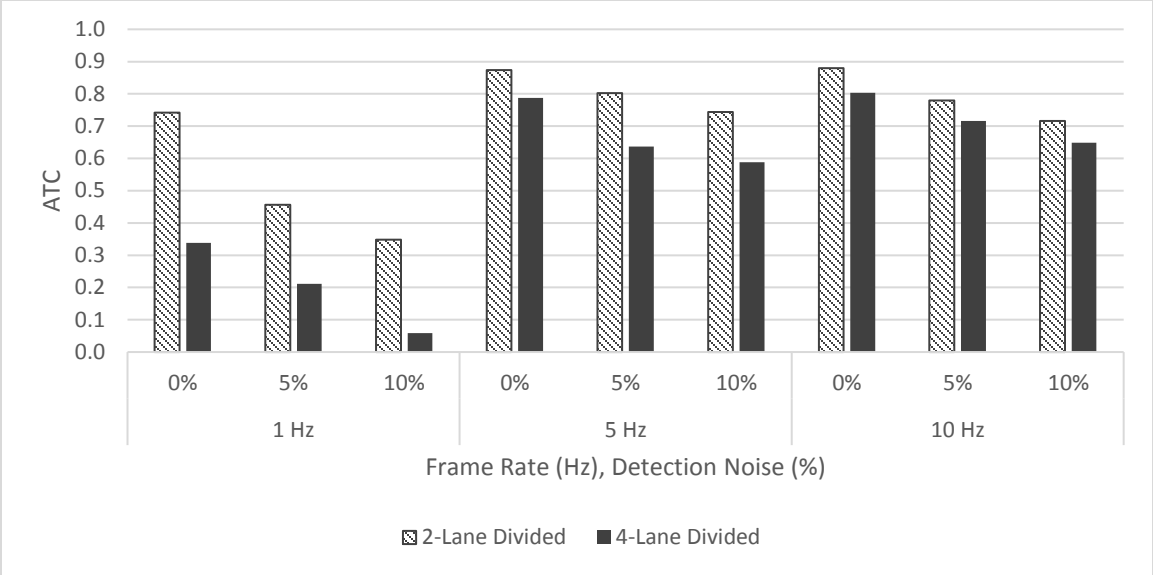


Figure 23 2-lane vs. 4-lane (divided, 7×std gating)

5. Among all data, the detection data generated by undivided 4-lane roads with 1 Hz frame rate and 10% noise provides the worst performance. This is reasonable. Besides the lowest frame rate and the highest percent noise, the tracking process is also influenced by vehicles in the opposite direction and vehicles on adjacent lane.
6. Using high value in Mahalanobis distance-based gating can slightly improve the performance with either 1 Hz data or 10% noise data (see Figure 24). This is reasonable because gating with higher value reserve extra predictions with higher kinematics variation, which can improve the performance with the data of low frame rate and high percent noise.

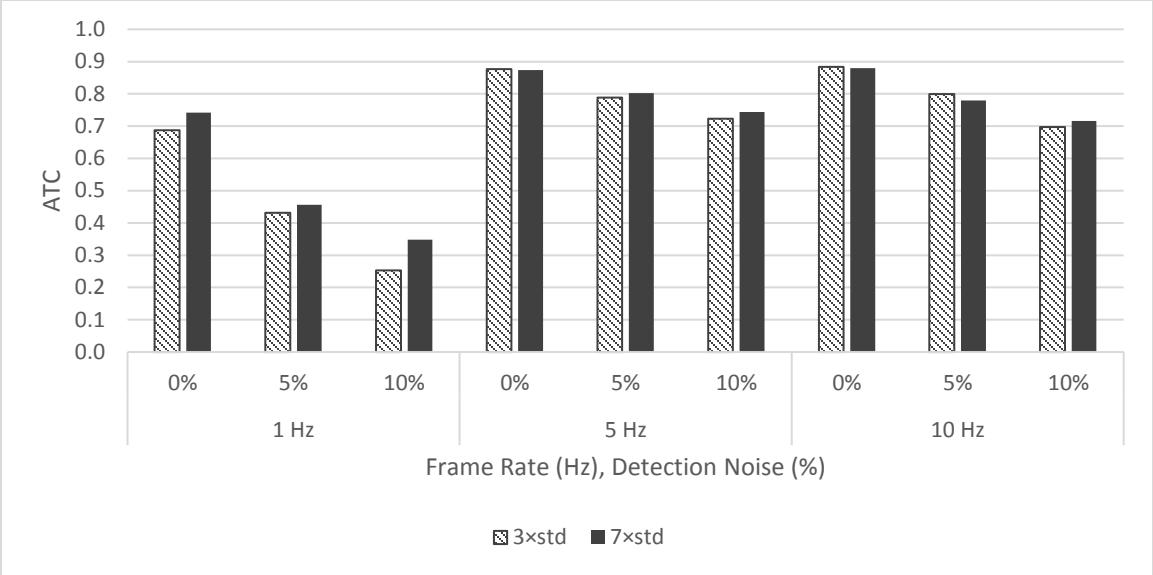


Figure 24 3×std vs. 7×std (2-lane, divided)

5.3.2 MHT with KAM

Besides consistent conclusions about frame rate and detection noise, the following conclusions can be drawn from the metrics shown in Table A3, A4, and A5 for the performance of the MHT approach with KAM:

1. Low appearance weight ratio has limited effects on the performance for all metrics.
2. Using high appearance weight ratio can significantly improve the performance with either low percent noise data or low frame rate data. The performance using high appearance weight ratio with 1 Hz data can be close or even better than the performance using low appearance weights with 5 or 10 Hz data, especially in the condition of low percent noise (see Figure 25).

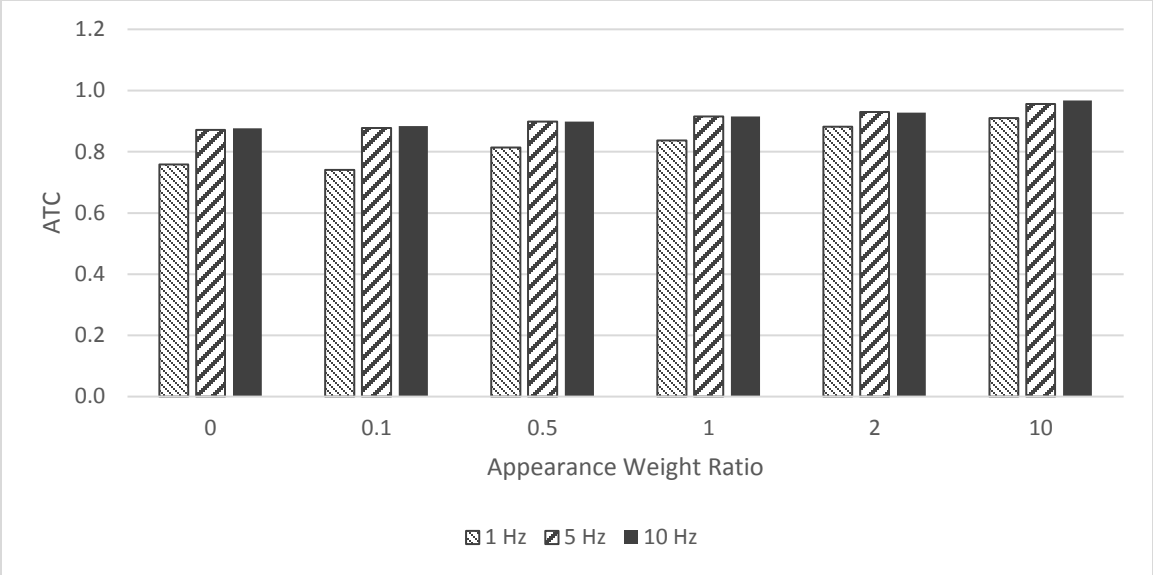


Figure 25 Effects of appearance weight ratio (2-lane, divided, 0% noise, 7×std, normalized)

3. With the data of low frame rate and high percent noise, using high appearance weight ratio can greatly improve the performance for all metrics (see Figure 26). It indicates that the appearance information can be more effective to track vehicles than kinematics information with data of low frame rate and high percent noise.

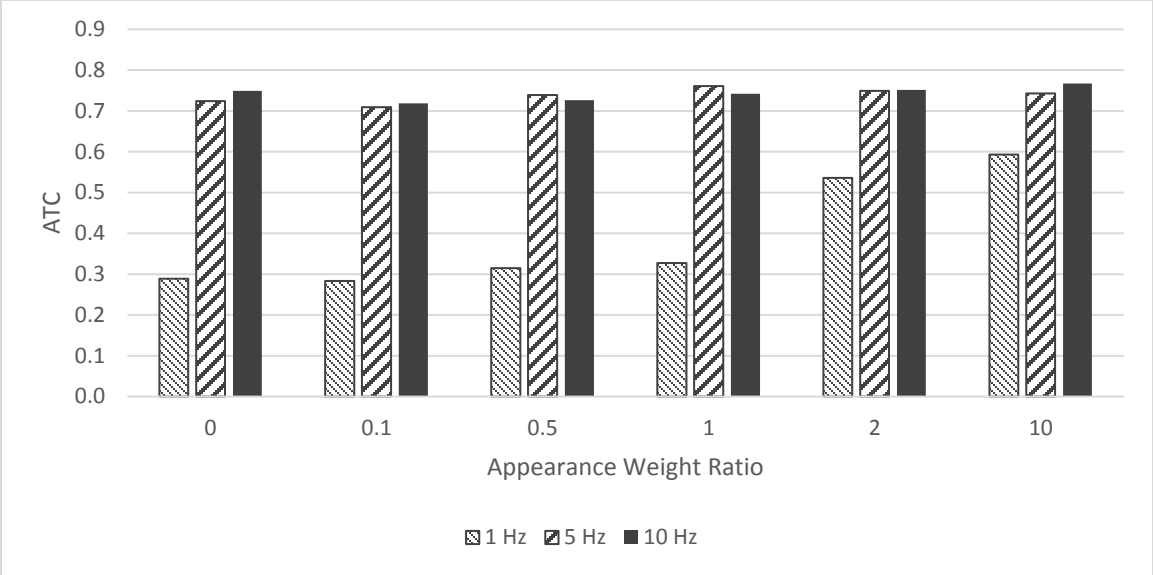


Figure 26 Effects of appearance weight ratio (2-lane, divided, 10% noise, 7×std, normalized)

4. Using a high value in Mahalanobis distance-based gating can improve the performance with the data of 1 Hz and 10% noise. Because the gating with higher combined Mahalanobis distance reserves extra predictions with higher variation of combined kinematics and appearance information, using a high value can improve the performance with the data of low frame rate and high percent noise.
5. The normalized weight function can lead to smooth transitions for all metrics from low appearance weight ratio to high appearance weight ratio, and it outperforms the unnormalized weight function except with the data of low frame rate and high percent noise (see Figure 27). For the unnormalized weight function, local minimum or maximum exists for all metrics. Because the proposed unnormalized weight function scales the weighted sum of Mahalanobis distance, the scale factor

of combined Mahalanobis distance reaches the peak of 2 where the appearance weight ratio is 1. The Mahalanobis distance-based gating using the scaled value reserves extra predications with higher variation. Those extra predications degrade the performance in most conditions but improve the performance with the data of low frame rate and high percent noise, which is consistent with the previous conclusions.

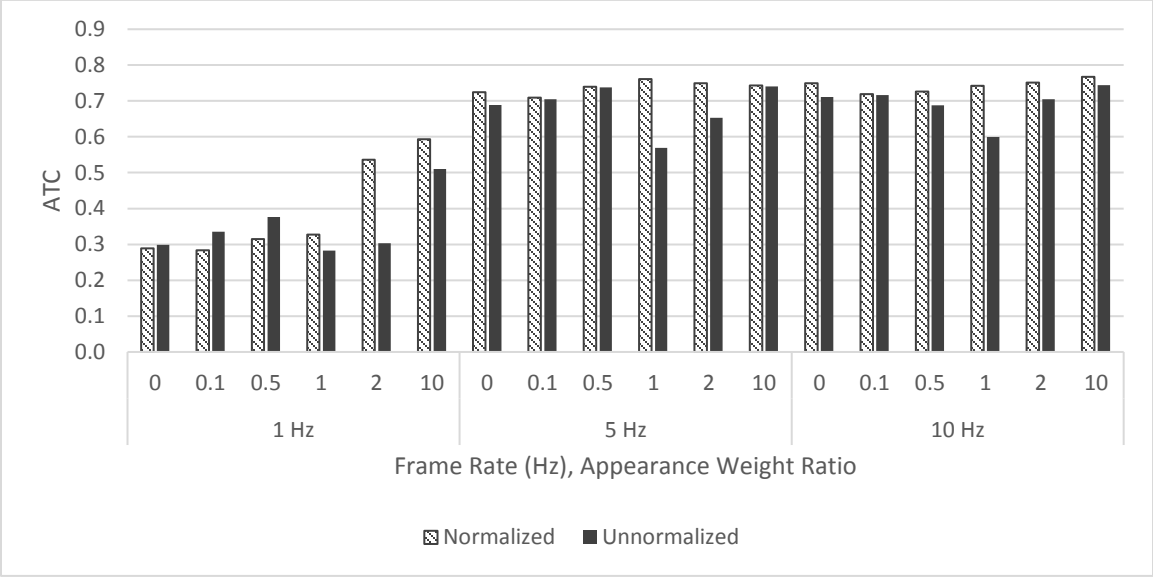


Figure 27 Normalized vs. unnormalized (2-lane, divided, 10% noise, 7×std)

5.4 Discussion

Using the MHT approach presented in this chapter to extend the computer vision-based approach presented in Chapter IV involves interdisciplinary challenges besides extensive coding. The MHT approach with KAM is based on linear dynamical systems, statistics, and probability theory; while the CNN approach with SURF is based on computer vision and deep learning. To establish the cooperation of two modules, their interfaces and the

formats of the vehicle detection data delivered between them were meticulously designed. To comprehensively research, test, and evaluate this MHT approach, the scale-agnostic property of MHT was originally induced. Since lack of ideal aerial imagery data for those experiments, a specialized method was applied to establish the synthetic dataset using VISSIM. Several metrics and statistics were identified and implemented for comprehensive evaluation.

The experimental results corresponded to the author's expectations regarding frame rate, detection noise and appearance weight ratio. However, the insignificant impacts of traffic configurations were unexpected. The KAM structure with the unnormalized weight function performed like the switching between the "credibility" of kinematics information and appearance information, which is unexpected and interesting; however, it was outperformed by the normalized weight function in most instances.

Based on the experimental results presented in this chapter, very promising performance can be achieved and the "closely-spaced targets" problem can be solved using high frame rate aerial video. Unfortunately, most existing aerial imagery datasets have low frame rates, thus the primary challenge for the usage of existing datasets involves how to improve the performance to enable the system to effectively process low frame rate imagery data. In this chapter, the experimental results provided three solutions for this challenge:

1. Improving the computer vision-based approach to achieve low percent detection noise.

2. Applying high appearance weight ratio in KAM.
3. Using large Mahalanobis distance or gating.

CHAPTER VI

CONCLUSION

In recent years, there has been tremendous interest in collecting diverse traffic data in a wide area by traffic surveillance technologies. Using wide area aerial video to achieve traffic surveillance by detecting and tracking vehicles in the network is one of the feasible ways that is facilitated by the advances in many related areas, including high performance optical sensor technology, persistent surveillance with camera arrays, advanced computer vision, reinvented deep learning, high performance computing units, and affordable platforms like UAV.

A great deal of aerial imagery datasets are currently available and more datasets are collected every day for various applications. There is great potential to make full and efficient use of these datasets to automatically extract useful traffic data. In order to achieve automated traffic surveillance for reliable and diverse traffic data collection, this in-depth research was focused on traffic surveillance and data extraction by detecting and tracking vehicles using wide area aerial video.

To achieve the first objective: investigating existing research related to traffic surveillance approaches and data extraction from aerial imagery to help guide this research and determine its contributions, the author reviewed and identified many existing research in several areas closely related to this research including: aerial surveillance systems, computer vision, deep learning, and MHT approaches. It is clear from the literature review

that this research is unique and that there is a need for a system that can automatically extract useful traffic data using WAPS. This system can lead to many potential applications to support and improve transportation planning, operations, research, and professional practice.

To achieve the second objective: investigating challenges with automatically processing wide area aerial video and extracting useful traffic data, the author investigated many exiting commercial and public aerial imagery datasets. Based on the investigation, most available imagery datasets have limited quality and frame rates and each of them have different characteristics and specifications, which make it very challenge to develop a system that can automatically possess those datasets. Numerous challenges were identified and analyzed to guide this research, and used to test implemented approaches in this research.

To achieve the third objective: implementing, testing, and evaluating computer vision-based vehicle detection and tracking approaches to monitor traffic and collect reliable traffic data using aerial imagery, the author implemented and tested several heuristic and machining learning approaches for vehicle detection and tracking using aerial imagery. A feature-based tracker and a CNN-based detector presented most the promising performance in the tests.

To achieve the fourth objective: identifying and refining the most promising approach and determining its capabilities, robustness, and limitations that may guide future research, the author combined a feature-based tracker and a CNN-based detector to achieve

a novel computer vision-based approach that presented very promising performance in detecting and tracking vehicles in wide area aerial video. The evaluation indicated this approach can achieve very accurate measurements for macroscopic traffic data for the traffic on road segments and has the potential to collect accurate microscopic data for individual vehicles. The author also implemented a MHT-based approach with an innovative KAM structure that cooperated with the computer vision-based approach to improve the tracking performance of the traffic surveillance system. The evaluation not only indicated this approach can achieve very promising results but also revealed the influences of some concerned factors and the solutions for some long-standing problems.

By accomplishing all four objectives, the author believes the overall goal of this dissertation was achieved. The author believes that the computer vision-based approach can extract reliable macroscopic traffic data from wide area aerial video and is practice ready. With future improvement by integrating the computer vison-based approach with the MHT-based approach, robust microscopic data extraction will be achieved.

6.1 Contribution

In this dissertation, the author researched and developed a traffic surveillance system that can successfully extract reliable and useful traffic data from wide area aerial video. This system can achieve accurate measurements for macroscopic traffic data and has the potential to collect robust microscopic traffic data.

A novel computer vision-based approach was proposed and implemented to achieve automated traffic surveillance by detecting and tracking vehicles in wide area

aerial video. This approach innovatively combined an emerging deep learning approach with handcrafted techniques. The successful research and development of the CNN approach with SURF achieved accurate measurement for macroscopic traffic data (including density, speed, and volume) and identified the need for more robust tracking to extract reliable microscopic data for individual vehicles.

A MHT-based approach with an innovative KAM structure was proposed and implemented to extend and improve the vehicle tracking system by cooperating with the computer vision-based approach. The innovative KAM structure took the advantage of the visual information contained in aerial imagery and the classification capability of emerging CNN. The scale-agonistic property of MHT was originally induced to comprehensively research, test, and evaluate the MHT approach. The successful research and development of the MHT approach with KAM indicated this approach can achieve very promising performance for tracking each individual vehicle and revealed the impacts of frame rate, detection noise, and traffic configurations on the performance, as well as the effects of appearance information of each vehicle on the performance. This research also provided solutions for the long-standing "closely-spaced targets" problem and solutions to achieve satisfactory performance for processing existing aerial imagery datasets that have limited quality and frame rates.

6.2 Further Research

In future research, approaches implemented in this dissertation will be integrated and improved for better accuracy, robustness, and efficiency. The computer vison-based

approach will be integrated with the MHT-based approach; and the trained ConvNet will be tuned for better performance. New approaches will be researched for potential applications to extend and improve the computer vision-based system. Joint probabilistic data association (JPDA) will be applied to improve the MHT approach; Siamese network and region-based CNN series will be applied to the computer vision-based approach.

Bibliography

Administration, F. H., 2016. *Traffic Monitoring Guide,* Washington, DC: Federal Highway Administration.

Agrawal, A. & Hickman, M., 2004. *Automated extraction of queue lengths from airborne imagery.* Washington, D.C., IEEE, pp. 297-302.

Angel, A., Hickman, M., Mirchandani, P. & Chandnani, D., 2003. Methods of analyzing traffic imagery collected from aerial platforms. *IEEE Transactions on Intelligent Transportation Systems,* 4(2), pp. 99-107.

Antunes, D. M., n.d. *Multiple Hypothesis Library Programming Manual.* [Online] Available at: http://www.multiplehypothesis.com/doc/MHL_Programming_Manual_1.0.pdf [Accessed 1 8 2016].

Arambel, P. O. et al., 2004. *Multiple-hypothesis tracking of multiple ground targets from aerial video with dynamic sensor control.* s.l., International Society for Optics and Photonics, pp. 23-32.

Arel, I., Rose, D. C. & Karnowski, T. P., 2010. Deep machine learning-a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine,* 5(4), pp. 13-18.

Blackman, S. S., 2004. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine,* 19(1), pp. 5-18.

Blackman, S. S., 2004. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine,* 19(1), pp. 5-18.

Blackman, S. S., Dempster, R. J. & a. R. R. W., 2001. *Demonstration of multiple-hypothesis tracking (MHT) practical real-time implementation feasibility.* San Diego, International Society for Optics and Photonics, pp. 470-475.

Blom, H. A. & Bloem., E. A., 2004. *Joint IMM and coupled PDA to track closely spaced targets and to avoid track coalescence.* Stockholm, International Society of Information Fusion, pp. 130-137.

Cao, X., Lan, J., Yan, P. & Li, X., 2012. Vehicle detection and tracking in airborne videos by multi-motion layer analysis. *Machine Vision and Applications,* 23(5), pp. 921-935.

Chen, B.-J. & Medioni, G., 2015. *3-D mediated detection and tracking in wide area aerial surveillance.* Waikoloa, IEEE Computer Society, pp. 396-403.

Cheng, H.-Y., Weng, C.-C. & Chen, Y.-Y., 2012. Vehicle detection in aerial surveillance using dynamic Bayesian networks. *IEEE Transactions on Image Processing,* 21(4), pp. 2152-2159.

Chen, X., Xiang, S., Liu, C.-L. & Pan, C.-H., 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters,* 11(10), pp. 1797-1801.

Coraluppi, S., Grimmett, D. & Theije, a. P. D., 2006. *Benchmark evaluation of multistatic trackers.* Florence, IEEE, pp. 1-7.

Cox, L. J. & Hingorani, S. L., 1996. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(2), pp. 138-150.

Du, X. & Hickman, M., 2012. Estimation of a road mask to improve vehicle detection and tracking in airborne imagery. *Transportation Research Record,* Issue 2291, pp. 93-101.

Gorji, A. A., Tharmarasa, R. & Kirubarajan, T., 2011. *Performance measures for multiple target tracking problems.* Chicago, IEEE, pp. 1-8.

Hickman, M. & Mirchandani, P. B., 2006. *Use of airborne imagery for microscopic traffic analysis.* Chicago, American Society of Civil Engineers.

Highway Capacity Manual 2010. 5th ed. Washington D.C.: Transportation Research Board.

Hinz, S., 2005. *Detection of vehicles and vehicle queues in high resolution aerial images.* Orlando, International Social Science Council, pp. 405-410.

Kalal, Z., Mikolajczyk, K. & Matas, J., 2012. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 34(7), pp. 1409-1422.

Kanhere, N. K., Birchfield, S. T., Sarasua, W. A. & Khoeini, S., 2010. Traffic monitoring of motorcycles during special events using video detection. *Transportation Research Record,* Issue 2160, pp. 69-76.

Kim, Z. & Malik, J., 2003. *Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking.* Nice, IEEE, pp. 524-531.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE,* 86(11), pp. 2278-2324.

Mahalanobis, P. C., 1936. *On the generalised distance in statistics.* Calcutta, s.n., pp. 49-55.

McCormack, E. D. & Trepanier, T., 2008. *The use of small unmanned aircraft by the Washington State Department of Transportation,* Seattle: Washington State Department of Transportation.

Moon, H., Chellappa, R. & Rosenfeld, A., 2002. Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing,* 20(1), pp. 1-13.

Nex, F. & Remondino, F., 2014. UAV for 3D mapping applications: a review. *Applied Geomatics,* 6(1), pp. 1-15.

Nguyen, T. T., Grabner, H., Bischof, H. & Gruber, B., 2007. *On-line boosting for car detection from aerial images.* Hanoi, IEEE, pp. 87-95.

Palaniappan, K. et al., 2010. *Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video.* Edinburgh, IEEE Computer Society, pp. 1-8.

Palaniappan, K., Rao, R. M. & Seetharaman, G., 2011. Wide-area persistent airborne video: Architecture and challenges. In: *Distributed Video Sensor Networks.* London: Springer, pp. 349-371.

Pelapur, R. et al., 2012. *Persistent target tracking using likelihood fusion in wide-area and full motion video sequences.* Singapore, IEEE Computer Society, pp. 2420-2427.

Prokaj, J. & Medioni, G., 2014. *Persistent tracking for wide area aerial surveillance.* Columbus, IEEE Computer Society, pp. 1186-1193.

Puri, A., 2005. *A survey of unmanned aerial vehicles (UAV) for traffic surveillance,* Tampa: Department of Computer Science and Engineering, University of South Florida.

Puri, A., Valavanis, K. . P. & Kontitsis, M., 2007. *Statistical profile generation for traffic monitoring using real-time UAV based video data.* Athens, IEEE, pp. 1-6.

Reid, D. B., 1979. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control,* 24(6), pp. 843-854.

Reinartz, P. et al., 2006. Traffic monitoring with serial images from airborne cameras. *ISPRS Journal of Photogrammetry and Remote Sensing,* 61(3), pp. 149-158.

Saleemi, I. & Shah, M., 2013. Multiframe many–many point correspondence for vehicle tracking in high density wide area aerial videos. *International Journal of Computer Vision,* 104(2), pp. 198-219.

Saunier, N. & Sayed, T., 2007. Automated analysis of road safety with video data. *Transportation Research Record,* Issue 2019, pp. 57-64.

Shi, X. et al., 2013. *Using maximum consistency context for multiple target association in wide area traffic scenes.* Vancouver, IEEE, pp. 2188-2192.

Skszek, S. L., 2001. *"State-of-the-Art" Report on Non-Traditional Traffic Counting Methods,* s.l.: Arizona Department of Transportation.

Tsai, Y. J., Wang, C. R. & Wu, Y., 2011. *A vision-based approach to study driver behavior in work zone areas.* Washington, D.C., Transportation Research Board, pp. 14-16.

Türmer, S., Leitloff, J., Peter Reinartz & Stilla, U., 2010. *Automatic vehicle detection in aerial image sequences of urban areas using 3D HoG features.* Saint-Mande, International Society for Photogrammetry and Remote Sensing, pp. 50-54.

Xiao, J., Cheng, H., Sawhney, H. & Han, F., 2010. *Vehicle detection and tracking in wide field-of-view aerial video.* San Francisco, IEEE Computer Society, pp. 679-684.

Zhao, T. & Nevatia, R., 2003. Car detection in low resolution aerial images. *Image and Vision Computing,* 21(8), pp. 693-703.

Zhao, X., Dawson, D., Sarasua, W. A. & Birchfield, S. T., 2016. *An automated traffic surveillance system with aerial camera arrays: Data collection with vehicle tracking.* Washington, D.C., Transportation Research Board.

Zhao, X., Dawson, D., Sarasua, W. A. & Birchfield, S. T., 2016. Automated Traffic Surveillance System with Aerial Camera Arrays Imagery: Macroscopic Data Collection with Vehicle Tracking. *Journal of Computing in Civil Engineering.*

APPENDIX A MHT EXPERIMENTAL RESULTS

Table A1 MHT with kinematics model only (3×std gating)

Frame Rate	Noise	2-Lane Divided								2-Lane Undivided							
		NV	ND	NT	NVA%	NMD%	NFA%	ANST	ATC	NV	ND	NT	NVA%	NMD%	NFA%	ANST	ATC
1 Hz	0%	188	17479	313	99.977	0.023	0.446	1.340	0.688	188	17478	301	99.983	0.017	0.401	1.324	0.717
	5%	188	16606	763	99.801	0.199	11.430	4.548	0.432	188	16605	797	99.874	0.126	11.382	5.495	0.409
	10%	188	15732	379	50.121	49.879	8.708	3.027	0.253	188	15731	392	49.018	50.982	9.268	4.032	0.210
5 Hz	0%	189	87092	245	99.993	0.007	0.001	0.296	0.876	189	87092	247	99.991	0.009	0.001	0.323	0.861
	5%	189	82738	401	99.885	0.115	5.527	0.741	0.789	189	82738	397	99.900	0.100	5.532	0.688	0.782
	10%	189	78383	686	99.745	0.255	11.871	1.095	0.723	189	78383	669	99.740	0.260	11.811	1.074	0.720
10 Hz	0%	189	174173	239	99.996	0.004	0.001	0.265	0.883	189	174173	244	99.995	0.005	0.000	0.302	0.871
	5%	189	165465	325	99.951	0.049	5.246	0.550	0.799	189	165465	329	99.926	0.074	5.204	0.635	0.789
	10%	189	156756	542	99.849	0.151	10.991	1.127	0.697	189	156756	510	99.887	0.113	11.066	1.021	0.725
		4-Lane Divided								4-Lane Undivided							
1 Hz	0%	394	35248	1236	99.997	0.003	2.726	6.500	0.316	394	35249	1258	99.989	0.011	3.132	7.272	0.288
	5%	394	33486	2722	99.961	0.039	16.034	12.886	0.175	394	33487	2662	99.970	0.030	16.514	13.165	0.187
	10%	394	31724	234	9.296	90.704	2.433	1.315	0.067	394	31725	392	12.643	87.357	3.984	3.157	0.072
5 Hz	0%	396	175793	553	99.976	0.024	0.015	0.503	0.791	396	175796	556	99.975	0.025	0.019	0.482	0.799
	5%	396	167004	1350	99.916	0.084	6.254	1.775	0.625	396	167007	1389	99.912	0.088	6.351	1.995	0.609
	10%	396	158214	1986	99.694	0.306	12.862	2.111	0.562	396	158217	2003	99.674	0.326	12.958	2.025	0.586
10 Hz	0%	396	351650	552	99.995	0.005	0.002	0.449	0.805	396	351658	557	99.992	0.008	0.002	0.449	0.802
	5%	396	334068	853	99.929	0.071	5.358	0.864	0.714	396	334076	841	99.930	0.070	5.373	0.846	0.712
	10%	396	316485	1435	99.838	0.162	11.404	1.437	0.627	396	316493	1480	99.823	0.177	11.414	1.487	0.630

Table A2 MHT with kinematics model only (7×std gating)

Frame Rate	Noise	2-Lane Divided								2-Lane Undivided							
		NV	ND	NT	NVA%	NMD%	NFA%	ANST	ATC	NV	ND	NT	NVA%	NMD%	NFA%	ANST	ATC
1 Hz	0%	188	17479	291	99.983	0.017	0.223	1.027	0.741	188	17478	306	99.971	0.029	0.452	1.441	0.693
	5%	188	16606	751	99.880	0.120	11.056	4.420	0.456	188	16605	758	99.849	0.151	11.316	4.819	0.425
	10%	188	15732	382	64.779	35.221	9.770	2.463	0.348	188	15731	352	49.476	50.524	8.741	3.505	0.218
5 Hz	0%	189	87092	246	99.989	0.011	0.001	0.302	0.873	189	87092	246	99.993	0.007	0.000	0.333	0.858
	5%	189	82738	385	99.857	0.143	5.389	0.656	0.802	188	82738	397	99.915	0.085	5.496	0.755	0.767
	10%	189	78383	627	99.770	0.230	11.723	1.021	0.743	189	78383	690	99.774	0.226	11.824	1.270	0.676
10 Hz	0%	189	174173	242	99.996	0.004	0.002	0.280	0.880	189	174173	240	99.998	0.002	0.001	0.275	0.880
	5%	189	165465	343	99.942	0.058	5.232	0.635	0.780	189	165465	339	99.952	0.048	5.262	0.571	0.798
	10%	189	156756	538	99.848	0.152	10.953	1.095	0.716	189	156756	526	99.872	0.128	11.004	1.079	0.708
		4-Lane Divided								4-Lane Undivided							
1 Hz	0%	394	35248	1162	100.000	0.000	2.678	6.038	0.339	394	35249	949	99.994	0.006	1.614	3.995	0.396
	5%	394	33486	2586	99.967	0.033	15.828	12.069	0.211	394	33487	2581	99.937	0.063	16.851	12.728	0.187
	10%	394	31724	233	8.602	91.398	2.418	1.244	0.059	394	31725	241	11.099	88.901	2.623	1.312	0.067
5 Hz	0%	396	175793	564	99.983	0.017	0.013	0.518	0.788	396	175796	558	99.986	0.014	0.009	0.495	0.797
	5%	396	167004	1267	99.932	0.068	6.142	1.601	0.637	396	167007	1308	99.906	0.094	6.148	1.652	0.628
	10%	396	158214	1970	99.724	0.276	12.769	1.990	0.588	396	158217	1993	99.716	0.284	12.836	1.944	0.566
10 Hz	0%	396	351650	557	99.996	0.004	0.001	0.449	0.803	396	351658	557	99.998	0.002	0.001	0.447	0.807
	5%	396	334068	843	99.947	0.053	5.351	0.768	0.716	396	334076	876	99.931	0.069	5.328	0.854	0.707
	10%	396	316485	1408	99.851	0.149	11.344	1.288	0.648	396	316493	1409	99.842	0.158	11.333	1.386	0.636

Table A3 MHT with KAM (2-lane, divided, 3×std gating, unnormalized weights)

Frame Rate	Weight Ratio	0% Noise					5% Noise					10% Noise				
		NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC
1 Hz	0	99.977	0.023	0.423	1.404	0.687	99.861	0.139	11.634	5.229	0.448	48.144	51.856	8.696	2.995	0.258
	0.1	99.994	0.006	0.446	1.612	0.685	99.855	0.145	10.683	4.197	0.473	44.819	55.181	8.009	2.867	0.203
	0.5	99.994	0.006	0.303	1.553	0.683	99.880	0.120	11.038	5.160	0.395	48.354	51.646	7.456	2.069	0.282
	1	99.983	0.017	0.646	2.245	0.604	99.874	0.126	12.568	7.745	0.291	40.383	59.617	6.681	2.133	0.233
	2	99.977	0.023	0.309	0.979	0.773	99.831	0.169	12.255	6.138	0.360	55.670	44.330	8.664	2.032	0.307
	10	99.949	0.051	0.109	0.383	0.894	99.819	0.181	11.695	4.457	0.429	82.011	17.989	11.473	2.309	0.464
5 Hz	0	99.995	0.005	0.000	0.302	0.874	99.906	0.094	5.532	0.693	0.793	99.791	0.209	11.924	0.979	0.735
	0.1	99.992	0.008	0.001	0.307	0.869	99.882	0.118	5.475	0.720	0.780	99.779	0.221	11.847	1.090	0.716
	0.5	99.984	0.016	0.002	0.312	0.873	99.851	0.149	5.623	1.349	0.703	99.698	0.302	11.801	1.360	0.656
	1	99.867	0.133	0.008	0.370	0.851	99.703	0.297	6.063	2.963	0.525	99.380	0.620	12.033	2.381	0.581
	2	99.944	0.056	0.008	0.233	0.897	99.769	0.231	5.975	2.201	0.601	99.594	0.406	12.096	1.725	0.641
	10	99.962	0.038	0.006	0.095	0.962	99.836	0.164	5.857	1.312	0.704	99.666	0.334	12.082	1.132	0.717
10 Hz	0	99.994	0.006	0.001	0.286	0.876	99.952	0.048	5.240	0.593	0.787	99.849	0.151	11.011	1.079	0.713
	0.1	99.994	0.006	0.001	0.286	0.877	99.937	0.063	5.213	0.566	0.798	99.827	0.173	10.990	1.085	0.718
	0.5	99.982	0.018	0.002	0.296	0.871	99.918	0.082	5.189	0.640	0.783	99.825	0.175	10.868	1.159	0.716
	1	99.907	0.093	0.002	0.323	0.861	99.781	0.219	5.256	0.878	0.725	99.494	0.506	11.062	1.788	0.614
	2	99.959	0.041	0.003	0.190	0.912	99.830	0.170	5.326	0.598	0.789	99.605	0.395	11.005	1.434	0.660
	10	99.975	0.025	0.003	0.090	0.964	99.857	0.143	5.359	0.608	0.814	99.682	0.318	11.228	1.180	0.730

Table A4 MHT with KAM (2-lane, divided, 7×std gating, unnormalized weights)

Frame Rate	Weight Ratio	0% Noise					5% Noise					10% Noise				
		NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC
1 Hz	0	99.960	0.040	0.269	0.809	0.781	99.843	0.157	11.321	4.968	0.392	59.757	40.243	9.630	2.628	0.298
	0.1	99.977	0.023	0.326	1.191	0.736	99.874	0.126	11.014	4.314	0.446	61.404	38.596	8.689	2.213	0.336
	0.5	99.983	0.017	0.195	1.197	0.722	99.904	0.096	11.568	5.027	0.419	68.453	31.547	9.751	2.532	0.376
	1	99.983	0.017	0.749	2.473	0.574	99.843	0.157	12.128	6.793	0.333	49.956	50.044	8.041	2.106	0.283
	2	99.983	0.017	0.383	1.319	0.707	99.904	0.096	11.833	5.346	0.412	54.977	45.023	8.677	2.229	0.303
	10	99.966	0.034	0.063	0.303	0.911	99.861	0.139	11.803	4.266	0.425	85.857	14.143	11.836	1.707	0.510
5 Hz	0	99.991	0.009	0.000	0.296	0.875	99.879	0.121	5.493	0.683	0.781	99.789	0.211	11.643	1.111	0.689
	0.1	99.983	0.017	0.000	0.296	0.876	99.884	0.116	5.438	0.571	0.799	99.745	0.255	11.559	1.111	0.705
	0.5	99.985	0.015	0.001	0.317	0.870	99.885	0.115	5.565	0.984	0.736	99.791	0.209	11.705	0.979	0.738
	1	99.866	0.134	0.016	0.397	0.848	99.762	0.238	6.009	2.635	0.570	99.492	0.508	11.946	2.254	0.569
	2	99.924	0.076	0.008	0.238	0.896	99.820	0.180	5.841	1.841	0.631	99.584	0.416	11.874	1.513	0.653
	10	99.960	0.040	0.003	0.095	0.961	99.851	0.149	5.845	1.280	0.692	99.673	0.327	11.954	1.164	0.740
10 Hz	0	99.997	0.003	0.001	0.275	0.880	99.949	0.051	5.241	0.630	0.790	99.874	0.126	11.066	1.079	0.711
	0.1	99.999	0.001	0.000	0.280	0.877	99.958	0.042	5.234	0.582	0.791	99.848	0.152	10.940	1.074	0.716
	0.5	99.984	0.016	0.001	0.296	0.872	99.921	0.079	5.230	0.656	0.786	99.830	0.170	10.944	1.138	0.688
	1	99.906	0.094	0.002	0.323	0.861	99.784	0.216	5.293	0.942	0.723	99.589	0.411	10.983	1.746	0.600
	2	99.964	0.036	0.002	0.185	0.915	99.854	0.146	5.308	0.841	0.763	99.682	0.318	11.013	1.212	0.705
	10	99.972	0.028	0.005	0.101	0.959	99.845	0.155	5.382	0.508	0.837	99.686	0.314	11.265	0.963	0.744

Table A5 MHT with KAM (2-lane, divided, 7×std gating, normalized weights)

Frame Rate	Weight Ratio	0% Noise					5% Noise					10% Noise				
		NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC	NVA%	NMD%	NFA%	ANST	ATC
1 Hz	0	99.989	0.011	0.172	0.867	0.759	99.813	0.187	11.327	4.718	0.410	55.734	44.266	8.804	2.809	0.289
	0.1	99.977	0.023	0.217	1.021	0.740	99.861	0.139	10.629	4.282	0.467	63.012	36.988	9.547	3.250	0.283
	0.5	100.000	0.000	0.080	0.686	0.813	99.898	0.102	10.870	3.931	0.498	62.096	37.904	9.090	2.862	0.315
	1	99.983	0.017	0.051	0.521	0.837	99.886	0.114	11.291	4.261	0.425	60.221	39.779	9.408	2.521	0.328
	2	99.971	0.029	0.046	0.335	0.881	99.886	0.114	11.393	4.255	0.455	83.835	16.165	11.238	1.590	0.536
	10	99.983	0.017	0.063	0.287	0.910	99.849	0.151	11.321	4.016	0.449	92.817	7.183	13.279	1.654	0.593
5 Hz	0	99.994	0.006	0.001	0.307	0.871	99.919	0.081	5.457	0.608	0.789	99.767	0.233	11.739	1.116	0.724
	0.1	99.992	0.008	0.000	0.286	0.878	99.901	0.099	5.442	0.714	0.771	99.775	0.225	11.533	1.090	0.709
	0.5	99.986	0.014	0.002	0.233	0.898	99.905	0.095	5.515	0.651	0.797	99.842	0.158	11.710	0.894	0.739
	1	99.984	0.016	0.000	0.196	0.915	99.894	0.106	5.498	0.741	0.786	99.777	0.223	11.545	0.884	0.761
	2	99.977	0.023	0.001	0.159	0.930	99.894	0.106	5.610	0.646	0.803	99.833	0.167	11.795	0.873	0.749
	10	99.966	0.034	0.005	0.106	0.956	99.883	0.117	5.814	0.899	0.778	99.700	0.300	11.946	1.005	0.743
10 Hz	0	99.999	0.001	0.000	0.286	0.877	99.944	0.056	5.242	0.630	0.783	99.837	0.163	11.027	0.989	0.749
	0.1	99.998	0.002	0.001	0.265	0.884	99.954	0.046	5.249	0.540	0.810	99.870	0.130	10.987	0.974	0.719
	0.5	99.997	0.003	0.001	0.228	0.898	99.948	0.052	5.227	0.503	0.821	99.858	0.142	10.997	0.989	0.726
	1	99.997	0.003	0.000	0.185	0.916	99.936	0.064	5.230	0.455	0.835	99.828	0.172	10.968	0.910	0.742
	2	99.992	0.008	0.001	0.153	0.928	99.927	0.073	5.283	0.402	0.850	99.865	0.135	11.090	0.852	0.751
	10	99.982	0.018	0.001	0.074	0.967	99.891	0.109	5.344	0.434	0.854	99.722	0.278	11.205	0.968	0.768

APPENDIX B CONFIGURATIONS OF FEATURE-BASED VEHICLE TRACKING FRAMEWORK

Appendix B provides the information of available methods for feature detector, feature descriptor, feature matcher, and feature matcher filter that can be applied in the feature-based vehicle tracking framework. The following codes show the configuration used in the experiments.

```
  /*
  "FAST"       - FastFeatureDetector
  "STAR"       - StarFeatureDetector
  "SIFT"       - SIFT (nonfree module)
  "SURF"       - SURF (nonfree module)
  "ORB"        - ORB
  "BRISK"      - BRISK
  "MSER"       - MSER
  "GFTT"       - GoodFeaturesToTrackDetector
  "HARRIS"     - GoodFeaturesToTrackDetector with Harris detector
enabled
  "Dense"      - DenseFeatureDetector
  "SimpleBlob" - SimpleBlobDetector
  */
const string detector_type = "SURF";

  /*
  "SIFT"  - SIFT
  "SURF"  - SURF
  "BRIEF" - BriefDescriptorExtractor
  "BRISK" - BRISK
  "ORB"   - ORB
  "FREAK" - FREAK
  */
const string descriptor_type = "SURF";

  /*
  "BruteForce" (it uses L2)
  "BruteForce-L1"
  "BruteForce-Hamming"
  "BruteForce-Hamming(2)"
  "FlannBased"
  */
const string matcher_type = "FlannBased";

  /*
```

```
  0 - NoneFilter
  1 - CrossCheckFilter
  */
const int matcher_filter_type = 1;
```

APPENDIX C DESIGN OF THE CONVNET FOR VEHICLE DETECTION

Appendix C provides the information of the ConvNet designed for vehicle detection in the PSS dataset using Caffe. Three design files (*.prototxt*) of the network are shown in this appendix. Weights (*.caffemodel*) of the network are not included in this appendix.

vehicleDetectorSolver.prototxt is the configuration file used to determine how the network is trained.

```
test_iter: 200
test_interval: 1000
base_lr: 0.005
display: 20
max_iter: 4000
lr_policy: "step"
gamma: 0.1
momentum: 0.9
weight_decay: 0.0005
stepsize: 1000
snapshot: 1000
snapshot_prefix: "vehicle_detector_train"
solver_mode: GPU
net: "detectorNetTraining.prototxt"
```

detectorNetTraining.prototxt is the model definition file used to determine the architecture of the network for training.

```
name: "VehicleDetectorNetTraining"
layer {
  name: "data"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    scale: 0.00390625
```

```
    mirror: false
    crop_size: 0
    mean_value: 139.569
    mean_value: 132.106
    mean_value: 130.663
  }
  data_param {
    source:
"/local_scratch/pbs.7461837.pbs02/vehicleTracking/build/src/caffe/train
.leveldb"
    batch_size: 256
  }
}
layer {
  name: "data"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    scale: 0.00390625
    mirror: false
    crop_size: 0
    mean_value: 139.569
    mean_value: 132.106
    mean_value: 130.663
  }
  data_param {
    source:
"/local_scratch/pbs.7461837.pbs02/vehicleTracking/build/src/caffe/test.
leveldb"
    batch_size: 256
  }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
  convolution_param {
    num_output: 25
    pad: 3
    kernel_size: 7
    stride: 1
```

```
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 1
    pad: 1
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
  convolution_param {
    num_output: 50
    pad: 2
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 4
```

```
    stride: 2
  }
}
layer {
  name: "fc1"
  type: "InnerProduct"
  bottom: "pool2"
  top: "fc1"
  inner_product_param {
    num_output: 100
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "fc1"
  top: "fc1"
}
layer {
  name: "fc2"
  type: "InnerProduct"
  bottom: "fc1"
  top: "fc2"
  inner_product_param {
    num_output: 2
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu2"
  type: "ReLU"
  bottom: "fc2"
  top: "fc2"
}
layer {
  name: "accuracy"
  type: "Accuracy"
  bottom: "fc2"
  bottom: "label"
  top: "accuracy"
  include {
    phase: TEST
  }
```

```
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "fc2"
  bottom: "label"
  top: "loss"
}
```

detectorNetDeploy.prototxt is the deployment file used to determine the architecture of trained network for deployment.

```
name: "VehicleDetectorNetDeploy"
layer {
  name: "memData"
  type: "MemoryData"
  top: "data"
  top: "label"
  transform_param {
    scale: 0.00390625
    mirror: false
    crop_size: 0
    mean_value: 139.569
    mean_value: 132.106
    mean_value: 130.663
  }
  memory_data_param {
    batch_size: 1
    channels: 3
    height: 100
    width: 100
  }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
  convolution_param {
    num_output: 25
    pad: 3
    kernel_size: 7
    stride: 1
    weight_filler {
```

```
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 1
    pad: 1
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
  convolution_param {
    num_output: 50
    pad: 2
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 4
    stride: 2
```

```
  }
}
layer {
  name: "fc1"
  type: "InnerProduct"
  bottom: "pool2"
  top: "fc1"
  inner_product_param {
    num_output: 100
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "fc1"
  top: "fc1"
}
layer {
  name: "fc2"
  type: "InnerProduct"
  bottom: "fc1"
  top: "fc2"
  inner_product_param {
    num_output: 2
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu2"
  type: "ReLU"
  bottom: "fc2"
  top: "fc2"
}
```

APPENDIX D SYNTHETIC VEHICLE DETECTION DATA

Appendix D provides the detailed information about synthetic vehicle detection data.

The CNN module outputs kinematics vehicle detection data, which consists of two variables: frame index and location of the detected vehicle. The kinematics model in MHT module only requires these two variables to track vehicles. The kinematics vector consists of 4 elements:

- x and y positions, which are provided by the kinematics vehicle detection data;
- x and y velocities, which can be easily computed by differencing x and y positions.

The appearance vector, which consists of a flexible number of float values, is the description of an image patch. The image patch of a detected vehicle can be created with the location of the detected vehicle and the aerial imagery. A trained ConvNet can process the image patch and encode it into the form of appearance vector as the description of the image patch.

VISSIM can record a variety of vehicle information generated during simulation. To establish reliable synthetic data for those specific requirements of the experiments, several types of simulation data were selected for recording vehicle information. An example of output vehicle record is shown:

Vehicle Record

File: c:\users\xiz\desktop\mht\1 lane divided\1 lane divided.inp
Comment:
Date: Monday, August 01, 2016 11:01:24 AM
VISSIM: 5.40-08 [38878]

t : Simulation Time [s]
tTot : Total Time in Network [s]
VehNr : Number of the Vehicle
Type : Number of the Vehicle Type
WorldX : World coordinate x (vehicle front end at the end of the simulation step)
WorldY : World coordinate y (vehicle front end at the end of the simulation step)
WorldZ : World coordinate z (vehicle front end at the end of the simulation step)
RWorldX : World coordinate x (vehicle rear end at the end of the time step)
RWorldY : World coordinate y (vehicle rear end at the end of the time step)
RWorldZ : World coordinate z (vehicle rear end at the end of the time step)

t	tTot	VehNr	Type	WorldX	WorldY	WorldZ	RWorldX	RWorldY	RWorldZ
4.4	0	1	100	1706.0531	107.7911	0.0000	1710.4531	107.7915	0.0000
4.6	0	1	100	1703.1693	107.7909	0.0000	1707.5693	107.7913	0.0000
4.8	0	1	100	1700.2725	107.7907	0.0000	1704.6725	107.7910	0.0000
5.0	0	1	100	1697.3626	107.7905	0.0000	1701.7626	107.7908	0.0000
5.2	0	1	100	1694.4396	107.7903	0.0000	1698.8396	107.7906	0.0000
5.4	0	2	100	100.1276	99.9451	0.0000	95.3676	99.9447	0.0000
5.4	0	1	100	1691.5036	107.7900	0.0000	1695.9036	107.7904	0.0000
5.6	0	2	100	102.9705	99.9453	0.0000	98.2105	99.9449	0.0000
5.6	0	1	100	1688.5545	107.7898	0.0000	1692.9545	107.7901	0.0000
5.8	0	2	100	105.8259	99.9455	0.0000	101.0659	99.9451	0.0000
5.8	0	1	100	1685.5976	107.7896	0.0000	1689.9976	107.7899	0.0000
6.0	0	2	100	108.6940	99.9458	0.0000	103.9340	99.9454	0.0000
6.0	0	1	100	1682.6420	107.7894	0.0000	1687.0420	107.7897	0.0000
6.2	0	2	100	111.5746	99.9460	0.0000	106.8146	99.9456	0.0000
6.2	0	1	100	1679.6957	107.7891	0.0000	1684.0957	107.7895	0.0000
6.4	0	2	100	114.4678	99.9462	0.0000	109.7078	99.9459	0.0000
6.4	0	1	100	1676.7625	107.7889	0.0000	1681.1625	107.7892	0.0000
6.6	0	2	100	117.3735	99.9465	0.0000	112.6135	99.9461	0.0000
6.6	0	1	100	1673.8423	107.7887	0.0000	1678.2423	107.7890	0.0000
6.8	0	3	100	99.6792	99.9450	0.0000	94.9192	99.9446	0.0000
6.8	0	2	100	120.2918	99.9467	0.0000	115.5318	99.9463	0.0000
6.8	0	1	100	1670.9351	107.7885	0.0000	1675.3351	107.7888	0.0000
7.0	0	3	100	102.5578	99.9453	0.0000	97.7978	99.9449	0.0000
7.0	0	2	100	123.2227	99.9470	0.0000	118.4627	99.9466	0.0000
7.0	0	1	100	1668.0411	107.7882	0.0000	1672.4411	107.7886	0.0000
7.2	0	3	100	105.4335	99.9455	0.0000	100.6735	99.9451	0.0000
7.2	0	2	100	126.1661	99.9472	0.0000	121.4061	99.9468	0.0000
7.2	0	1	100	1665.1601	107.7880	0.0000	1669.5601	107.7884	0.0000
7.4	0	3	100	108.3064	99.9457	0.0000	103.5464	99.9453	0.0000
7.4	0	2	100	129.1222	99.9474	0.0000	124.3622	99.9471	0.0000
7.4	0	1	100	1662.2921	107.7878	0.0000	1666.6921	107.7881	0.0000
7.6	0	3	100	111.1766	99.9460	0.0000	106.4166	99.9456	0.0000
7.6	0	2	100	132.0907	99.9477	0.0000	127.3308	99.9473	0.0000
7.6	0	1	100	1659.4372	107.7876	0.0000	1663.8372	107.7879	0.0000
7.8	0	3	100	114.0441	99.9462	0.0000	109.2841	99.9458	0.0000
7.8	0	2	100	135.0719	99.9479	0.0000	130.3119	99.9475	0.0000
7.8	0	1	100	1656.5954	107.7874	0.0000	1660.9954	107.7877	0.0000
8.0	0	3	100	116.9091	99.9464	0.0000	112.1491	99.9461	0.0000
8.0	0	2	100	138.0606	99.9482	0.0000	133.3006	99.9478	0.0000
8.0	0	1	100	1653.7666	107.7872	0.0000	1658.1666	107.7875	0.0000
8.2	0	3	100	119.7717	99.9467	0.0000	115.0117	99.9463	0.0000
8.2	0	2	100	141.0478	99.9484	0.0000	136.2878	99.9480	0.0000
8.2	0	1	100	1650.9509	107.7869	0.0000	1655.3509	107.7873	0.0000
8.4	0	3	100	122.6293	99.9469	0.0000	117.8693	99.9465	0.0000
8.4	0	2	100	144.0260	99.9487	0.0000	139.2660	99.9483	0.0000
8.4	0	1	100	1648.1483	107.7867	0.0000	1652.5483	107.7871	0.0000
8.6	0	3	100	125.4757	99.9471	0.0000	120.7157	99.9468	0.0000
8.6	0	2	100	146.9916	99.9489	0.0000	142.2316	99.9485	0.0000
8.6	0	1	100	1645.3587	107.7865	0.0000	1649.7587	107.7869	0.0000
8.8	0	4	100	101.0277	99.9451	0.0000	96.4177	99.9448	0.0000
8.8	0	3	100	128.3076	99.9474	0.0000	123.5476	99.9470	0.0000
8.8	0	2	100	149.9446	99.9492	0.0000	145.1846	99.9488	0.0000
8.8	0	1	100	1642.5770	107.7863	0.0000	1646.9770	107.7866	0.0000
9.0	0	4	100	103.9084	99.9454	0.0000	99.2984	99.9450	0.0000
9.0	0	3	100	131.1250	99.9476	0.0000	126.3650	99.9472	0.0000
9.0	0	2	100	152.8851	99.9494	0.0000	148.1251	99.9490	0.0000
9.0	0	1	100	1639.7938	107.7861	0.0000	1644.1938	107.7864	0.0000
9.2	0	4	100	106.7890	99.9456	0.0000	102.1790	99.9452	0.0000
9.2	0	3	100	133.9282	99.9478	0.0000	129.1682	99.9474	0.0000
9.2	0	2	100	155.8130	99.9496	0.0000	151.0530	99.9492	0.0000
9.2	0	1	100	1637.0015	107.7859	0.0000	1641.4015	107.7862	0.0000
9.4	0	4	100	109.6692	99.9458	0.0000	105.0592	99.9455	0.0000

The information contained in vehicle record can provide the frame index and location for each individual simulated vehicle:

- the frame index can be created by the simulation time;
- the location of a simulated vehicle can be calculated by the location of vehicle front and rear.

Besides, vehicle record also contains number of vehicle, which can provide the ground truth information for reliable evaluation. Total time in network and vehicle type were not used in the experiments in this research but they can be potentially useful for testing and evaluation.

To generate appearance information in synthetic vehicle detection data, a vector of float values was appended to each vehicle detection record as its appearance information. The vector length can be flexible based on detailed implementation, and 5 was used in these experiments. 10 classifications were defined for the fact that many vehicles have similar appearance, and then corresponding appearance vectors were attached in the synthetic data. A uniformly distributed random float number within the range from -1 to -1 was initialized in each element in the appearance vector to define the appearance of the vehicle. The appearance variation following a normal distribution with a mean of 0 and a standard deviation of 0.1 was added to each element in the appearance vector to simulate the instability of the appearance of the same vehicle in different frames. Additionally, the measurement noise following a normal distribution with a mean of 0 and a standard

deviation of 0.3 was added to each element in the appearance vector to simulate the detection inaccuracy. The implementation is shown:

```
# create 10 random 5 element vectors to represent typical car types:
appearanceVecLen = 5
numberCarModels = 10

appearanceVecLow = -1.0
appearanceVecHigh = 1.0

carModels = []
for i in range(numberCarModels):
    carModels.append(np.random.uniform(appearanceVecLow,
appearanceVecHigh, appearanceVecLen))

# generate samples
carVariationStdDev = 0.1
measurmentStdDev = 0.3
vehicles = {}

# generates a sample given a car index:
def getAppModel(vId):
    if not vId in vehicles:
        modelId = np.random.randint(numberCarModels)
        vehicles[vId] = carModels[modelId] + np.random.normal(0,
carVariationStdDev, appearanceVecLen)

    return   vehicles[vId]   +   np.random.normal(0,   measurmentStdDev,

appearanceVecLen)
```

To generate the detection noise caused by the inaccuracy of the detector, false positive were created by randomly adding detections near existing the ground truth from synthetic vehicle detection data, and the same percentage of false negative were created by randomly removing ground truth from the detection data. The implantation is shown:

```
public void addFalsePositives(Map<Double, List<Detection>> allFrames,
int numberToAdd) {
  List<List<Detection>> allFramesList = new
ArrayList<List<Detection>>(allFrames.values());
  for (int i = 0; i < numberToAdd; i++) {
    List<Detection> randFrame = allFramesList.get(
        ThreadLocalRandom.current().nextInt(allFrames.size()));
    Detection randDetection = randFrame.get(
        ThreadLocalRandom.current().nextInt(randFrame.size()));
```

```
    double randDirection = ThreadLocalRandom.current()
        .nextDouble(2.0*Math.PI);
    double randDistance = ThreadLocalRandom.current()
        .nextDouble(4.0, 24.0)*.3048;
    Detection fp = new Detection();
    fp.x = randDetection.x +
        Math.cos(randDirection)*randDistance;
    fp.y = randDetection.y +
        Math.sin(randDirection)*randDistance;
    fp.vehicleNumber = -1;
    fp.vehicleType = -1;
    fp.t = randDetection.t;
    if (this.appearanceVectorLength > 0) {
      Detection randDetection2 = randFrame.get(
          ThreadLocalRandom.current().nextInt(randFrame.size()));
      fp.appearance = randDetection2.appearance.clone();
    }
    allFrames.get(fp.t).add(fp);
  }
}

public void addFalseNegatives(Map<Double, List<Detection>> allFrames,
int numberToRemove) {
  List<List<Detection>> allFramesList = new
ArrayList<List<Detection>>(allFrames.values());
  while (numberToRemove > 0) {
    List<Detection> randFrame = allFramesList.get(
        ThreadLocalRandom.current().nextInt(allFrames.size()));
    if (randFrame.size() <= 0)
      continue;
    int randDetectionIdx =
        ThreadLocalRandom.current().nextInt(randFrame.size());
    Detection randDetection = randFrame.get(randDetectionIdx);
    if (randDetection.vehicleNumber >= 0) {
      randFrame.remove(randDetectionIdx);
      numberToRemove--;
    }
  }
}
```