Computer Algorithms for Traffic Signal Recognition

By
Dongxi Zheng

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Civil and Environmental Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2016

Date of final oral examination: 05/09/2016
The dissertation is approved by the following members of the Final Oral Committee:
David A. Noyce (advisor), Professor, Civil and Environmental Engineering
Madhav V. Chitturi (co-advisor), Research Associate, Civil and Environmental Engineering
Bin Ran, Professor, Civil and Environmental Engineering
Soyoung Ahn, Professor, Civil and Environmental Engineering
Charles R. Dyer, Professor, Computer Sciences
John D. Lee, Professor, Industrial and Systems Engineering

ABSTRACT

A set of algorithms are proposed for traffic signal recognition (TSR) on challenging videos. During the method development, minimal to no assumptions were made about the uniformity of cameras, the accessibility of advanced controls (e.g., shutter speed), the availability of camera-dependent sample data, the environmental lighting conditions, or the distance to the traffic lights. Such openness of input requires the algorithms to be relatively generic and adaptable to various devices and scenarios.

The proposed methodology consists of two major subsets: 1) image based traffic light detection and classification and 2) spatiotemporal information based coordination. At the core of the methodology is a candidate traffic light detection method based on the concept of conspicuity, which involves lightness, color saturation, and contrast. Detected candidates are then classified based on robust relative color similarity. When processing a video, spatiotemporal information (i.e., GPS based camera position and frame timestamp) is used to effectively narrow down the temporal search range and coordinate TSR across frames.

Naturalistic driving videos were tested against these algorithms to analyze the performance and reveal challenges. The proposed detection method outperformed two other generic detection algorithms in nearly all lighting-distance scenarios, although the absolute recall rates (around 50%) were low due to the compromised data quality. Classification achieved nearly 95% accuracy even with strong color variation in the data. The spatiotemporal coordination effectively reduced the data and helped to reach ideal temporal accuracy of TSR through persistent tracking. Challenge wise, sunny daytime was found undesirable due to strong ambient light and a single set of

parameters in the detection model was not optimal for all lighting conditions. Nevertheless, intuitive rules were found for tuning the model towards different lighting conditions.

In summary, this study contributes to the state of knowledge in TSR by proposing a set of novel algorithms and analyzing their performance on unprecedented naturalistic driving data. These algorithms are expected to be more suitable than existing methods for processing videos acquired by a diverse camera set under various lighting conditions.

ACKNOWLEDGEMENT

The author owes thanks to many. First and for most, I am in endless debt to my advisor Professor David A. Noyce, without whose recognition, support, and patient, I would not have started my journey in a graduate school, kept learning for nearly eight years, and reached this point of accomplishment. It is also my exceptional fortune to work with Dr. Madhav V. Chitturi, a wise, honest, and critical colleague and friend who motivated me and whipped my procrastination along the way.

I feel grateful and lucky to have Professors Bin Ran, Soyoung Ahn, John Lee, and Charles Dyer as my dissertation committee members. Professor Ran is my role model in the field of intelligent transportation systems and urban planning. His wisdom of transportation innovation has been a major source of inspiration to me. Professor Ahn is always the bright person that I enjoy talking to. Her comprehensive thinking helped me to see a better merit in my research. I am thrilled to have the chance meeting and working with Professor Lee, whose academic integrity not only kept me on the right track of research and project commitments, but also set a new level of research standards that I never thought I would have touched. Professor Dyer, the nice and smiling "Chuck", cannot be thanked more for his guidance on computer vision and always being a patient listener to my immature thoughts on research. Also, the books he lent me were so referable that I almost decided to keep for myself silently after graduation.

I must appreciate the fraternity provided by other TOPS lab members and alumni. Dr. Xiao Qin mentored me like a big brother when I started my graduate school. Dr. Zhixia Li has been my magical source box and took me to my first TRB journal publication. Dr. Steven Parker was always as approachable as the WisTransPortal database he maintained. Andrea Bill responded to my needs

faster than she talked. Kelvin Santiago has been my best virtual competitor and mutual-teasing friend. Li-Hong (David) Chiu deserves my gratitude for his extensive labor work in helping me collect ground truth data. Dr. Ghazan Khan, Dr. Jing Jin, Dr. Yang Cheng, Dr. Danjue Chen, among others, have brought positive impacts to my research and life in many aspects

Special thanks should go to the crew of the backing project of this research. Dr. Brandon Smith was among my most exemplar scholars in computer vision. I learned countless things from Brandon in addition to referring to his work performance to pace mine, except his was for the facial landmark localization in the project. Professors Yu Hen Hu and Robert Radwin gave me useful tips on algorithm design and advices on the applications of my work. They were wise professors to have around. Last but not least, Xuan Wang and Oguz Akkas were motivated peers to work with. I hope I learned things as fast as they could.

All the roads I have stumbled through and will walk on are with and for my beloved family. My late father, Rongju Zheng, tinkered most of my personality. My mother, Huiying Huang, taught me gratefulness. My father and mother in law, Xingyuan Lei and Fajin Jiang, helped me out during my most stretching time. My god parents financially supported me through my earlier ages of education before grad school. My wife, Yuxia Lei, weaves a place I call home with all her love, in which my little Daniel and Sophia constantly remind me how brave and hardworking I should be as I expect them to be when they grow up. I hope you find it worth the long wait.

TABLE OF CONTENTS

LIST OF SELECTED NOTATIONS

w_L	-	Weight of average lightness
w_{LC}	-	Weight of lightness contrast
w_S	-	Weight of average saturation
RAD	-	A set of searched radii in pixels
h_{peak}	-	Threshold of normalized conspicuity value
N_{top}	-	Maximum number of candidates with the top conspicuity values to be detected
N_{pi}	-	Maximum number of candidates allowed in each localization iteration
asp_{min}	-	Minimum aspect ratio
pd_{min}	-	Minimum pixel density

LIST OF ACRONYMS

ADAS – Advanced Driver Assistance System
AVS – Autonomous Vehicle System
DAS – Data Acquisition System
DOF – Dense Optical Flow
DRS – Dynamic Roadway State
FHWA – Federal Highway Administration
FOT – Field Operational Test
FOV – Field-Of-View
FPR – False Positive Rate (= 1 – Precision)
GCS – Geographic Coordinate System
GPS – Global Positioning System
HDR – High Dynamic Range
HK – Helmholtz-Kohlrausch
HOG – Histogram of Gradients
HMM – Hidden Markov Model
IMU – Inertial Measurement Unit
MUTCD – Manual of Uniform Traffic Control Devices
ND – Naturalistic Driving
NDS – Naturalistic Driving Study
OSM – OpenStreetMap
PCS – Projected Coordinate System
RDCW - Road Departure Crash Warning
RID – Roadway Information Dataset
RLR – Red Light Running
ROI – Region of Interests
RSA – Road Safety Audit
SHRP 2 – Second Strategic Highway Research Program
STAC – Safety Training and Analysis Center
SUR – Seemingly Unrelated Regression
SVM – Support Vector Machine

TCD – Traffic Control Device

TSD – Traffic Signal Detection

TSC – Traffic Signal Classification

TSR – Traffic Signal Recognition

TTEC – Time to Edge Crossing

VGA – Video Graphic Array

V2V – Vehicle to Vehicle Communication

VTTI – Virginia Tech Transportation Institute

WGC84 – World Geodetic System 1984

XML – EXtensible Markup Language

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

Vision based traffic signal recognition (TSR) systems are crucial components of intelligent transportation systems (ITS) and advanced driving assistance systems (ADAS). Roadside video cameras have been used to identify the traffic signal state along with the vehicle trajectories to capture red-light-runners *(1)*. Onboard video cameras with traffic signal recognition back-ends found even wider applications such as infrastructure inventory, signal state and approaching speed advisory, and autonomous driving *(2–5)*. Because video cameras possess sensing advantages such as high data frequency, rich colors, nonintrusive (or passive) interface, long and reliable distance range, and inexpensive infrastructure investment, they remain an economic and reliable choice over other sensing technologies such as radar, LiDAR, and telecommunication for TSR.

However, image formation process, environmental condition, and camera pose and orientation can jointly introduce challenging target appearance. Severe color variation, such as distorted color, underexposure, and overexposure, is one of such challenges. Since most existing systems relied on color segmentation for initial signal detection, the ability to accommodate color variation and effectively separate traffic signal pixels from the background became a fundamental requirement in their system designs. The general solution is to calibrate (or trained) a color classifier. Previous studies based their calibration process either on training images or on local traffic signal design standards. The standards based methods were inherently camera independent, but their effectiveness has not been validated over a wide variety of camera settings and their transferability to other geographical regions could be limited. The training images based methods

relied on using the same camera setting in both calibration and testing. Porting any of these methods to a different camera setup is therefore non-trivial, because new training images need to be resampled. In addition, some of these systems used special camera exposure controls to alleviate color variation, making adoption on other less controllable devices impossible.

While the data acquisition systems can always be calibrated to achieve optimal data quality for TSR, there are cases when the data are generic and with far less perfect quality. One of such situations is extracting traffic signal states from general purpose driving videos. The extracted information can be critical in assessing traffic violations and driver behavior, such as red-light-running. *A robust TSR system that accommodates generic data source will lead to efficient and innovative workflows in law enforcement agencies, auto insurance companies, and the general traffic safety research community*. Such research need is recently boosted with the availability of massive video data collected by naturalistic driving studies (NDS).

NDS are gaining popularity for traffic safety investigations. As people have long realized, human errors are a key contributor to traffic crashes. Both highway design and traffic controls need to take into account human's physical and mental capabilities *(6, 7)*. So far, researchers' primary source of evidence of human errors are historical crash reports. These empirical observations have revealed valuable insights into problems such as aggressive driving, impaired driving, drowsy driving, distracted driving, and confusion, among others *(8)*. However, crash reports can only recover loosely connected pieces of information, sometimes biased, and are not capable of providing a continuous spatial-temporal account for analyzing deeper aspects of driver behaviors. In addition, crashes are not the only consequences caused by human errors. Near misses and traffic violations are also hazardous events attributing to driver performance but normally underrepresented. Therefore, a more comprehensive way of collecting continuous driving data that

cover more events of interests has been pressingly needed. NDS meets such research need by providing honest on-board video recordings of the driver and the roadway environment, in addition to the vehicle's position and kinematics *(9, 10)*. A typical NDS data acquisition system (DAS) was illustrated by Antin et al. and copied in Figure 1-1 *(11)*. Data collection runs continuously and unobtrusively over a relatively long study period (e.g., one year) whenever the instrumented vehicle is driven *(12, 13)*. Pioneering researches using NDS data have reported novel findings about lane departures on rural two-lane curves, offset left-turn lanes, rear-end crashes on congested freeways, and driver inattention and crash risk *(14–16)*. More researches are expected to be supported by NDS data for decades to come.

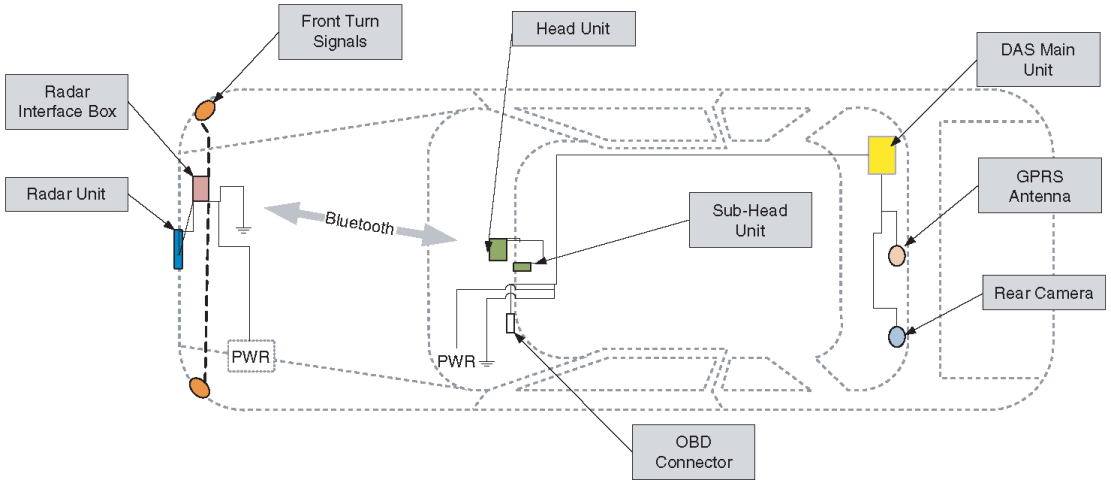


Figure 1-1 Illustration of the DAS of 100-Car and SHRP 2 NDS *(11)*.

Unfortunately, the unparalleled temporal and spatial coverage of NDS data come at the expense of aggressive data compression for reduced but still enormously large data size. Two renowned NDS studies on passenger cars, for example, are the 100-Car study and the Second Strategic Highway Research Program (SHRP 2) NDS *(17)*. The 100-Car study collected nearly 43,000 hours of driving data over an 18 month period with 241 primary and secondary participant drivers. A total of 82 crashes, 761 near misses, and 8,295 other types of interesting events were

identified *(12)*. Extreme behaviors such as severe fatigue, impairment, judgment error, risk taking, willingness to engage in secondary tasks, aggressive driving, and traffic violations were reported *(18)*. SHRP 2 NDS used an updated DAS on over 3,300 vehicles participated at six study sites in different states, providing a wider range of geography, weather, state laws, and road features. Over a period of nearly three years, more than two million hours of driving data were collected, which captured 1,465 crashes and 2,710 near misses among other events *(13, 19)*. The total video data size is over two petabytes. Before being transferred to a data center, all video data and other sensor data of individual vehicles were stored on their on-board storage units (the "DAS main unit" in Figure 1-2). Such system design imposed a requirement for aggressive data compression, particularly on the video data. A demonstration given by Antin et al. *(11)* is copied in Figure 1-2 to show four views of video being composited into one image frame during data collection. The pixel resolution of the front-view video is 350-by-480, which is not generous for an 83-degree field of view (FOV) being covered.

Figure 1-2 Snapshot of compressed SHRP 2 videos *(11)*.

1.2 Problem Statement

Both the compromised quality and the massive size of naturalistic driving videos are unaddressed challenges to all existing TSR systems. Imprecise camera imaging process, varying lighting conditions, and changing distance can lead to severely inconsistent appearance of traffic lights. No prior studies have looked into such extreme data setting. In addition, scaling existing systems to million hours of videos is not only a problem of improving per-frame recognition speed, but also a challenge to the spatiotemporal framework that coordinates the TSR. In order to address the above challenges, several research questions need to be answered:

- What features of a traffic light are most invariant to various lighting conditions and pixel resolutions and how can these features be modeled and used in detection?
- How can the robustness be improved on traffic signal color classification when the training color samples do not closely match the testing data?
- What spatiotemporal information is useful in extracting relevant frame ranges for TSR? How can such spatiotemporal information be used to coordinate TSR so more reliable results can be achieved?

1.3 Research Objectives

The proposed study should achieve the following objectives:

- Provide an up-to-date literature review on vision based TSR systems with comprehensive understanding about their application scenarios, detection and classification methods, and spatiotemporal coordination approaches. Identify limitations of the existing methods for improvement.

- Explain in a scientific way how traffic signals can be distinguished in a road scene and model this process mathematically. Implement this process in a computer program as a generic traffic signal detector.
- Develop a classifier that gives soft classification to detected traffic signal candidates while maintaining the ability of reflecting the true signal color with high confidence measures.
- Establish a spatiotemporal framework that effectively identifies relevant frames for TSR processing and coordinates TSR with temporal tracking in a way that increases the stability of recognition results.
- Collect detailed ground truth data from naturalistic driving videos with various lighting conditions and test the above three TSR related components on these videos with analyses of the performance.
- Recommend applications of the developed methodology.

1.4 Research Scope

The current research is under the following scope restriction:

- In terms of video data, this study only considers videos taken by an on-board camera that faces the direction of travel. The field of view captures the front roadway and should be able to include the overhead signal faces when the vehicle is in the middle of the intersection.
- This study focuses on offline data reduction for driving context and driver behavior analysis rather than real time vehicle navigation. However, it might be possible to port the proposed methodologies to real time applications with certain modification and computational speed improvement.

- Although pedestrian signals are also important for behavioral analysis, the only command for drivers are from vehicular traffic signals. So, traffic signal state only refers to vehicular traffic signals at highway intersections. Ramp metering is also excluded from this study.
- Arrow shape traffic signals are currently not separately considered in this study. Due to low pixel resolution and light diffusion, the arrow shape signals are not expected to be clearly outlined. Distinction between arrow and circle has also been found unnoticeable beyond 50 m *(20)*.

1.5 Contributions

This research will contribute to the state of knowledge by:

- Developing a collection of generic traffic signal recognition algorithms that can be applied on a wide range of video data without device dependent calibration. Yet, the algorithms allow intuitive control to accommodate various physical scenarios.
- Providing an unprecedented insights into the challenges of traffic signal recognition using videos with compromised quality and difficult lighting conditions.

1.6 Thesis Organization

The rest of this thesis is organized into five chapters. Chapter 2 gives a comprehensive review on TSR related research. Topics include TSR applications, detection methods, classification methods, and spatiotemporal coordination. Chapter 3 explained the proposed methodologies. First, the conspicuity based detection model is explained with comprehensive formulation. Then, a histogram similarity based signal classifier is proposed. Both the detector and the classifier form the core of TSR and are embedded in a spatiotemporal framework. Key stages

of the framework are described, including map projection, vicinity calculation, movement classification, short range initialization, and long range tracked recognition. Chapter 4 describes the data collection effort. Chapter 5 tested the detector and the classifier both separately on individual frames and under the spatiotemporal framework. Performance results are given and analyzed. Chapter 6 concludes this study with discussions.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In general, TSR is composed of two major stages: detection and classification. During the detection stage, traffic signal candidates (individual lenses or whole signal faces) are extracted from the image as separate regions. In the classification stage, the detected candidates are tested against other criteria to determine their signal state (i.e., red, green, or yellow) and/or shape (i.e., arrow or circle). Although TSR can be performed on individual images separately, a more common circumstance is working with videos where the temporal information can facilitate successive detections and prune candidates. Physical information such as camera calibration, position, and orientation and traffic signal maps are also helpful for eliminating irrelevant frames or narrowing the ROIs in the image.

Depending on the actual application, TSR can be implemented or extended differently according to the underlying data acquisition systems (i.e., data variety, format, and quality), expected environmental conditions, workflows, and other problem settings. A thorough review of the existing systems is provided in this chapter as a knowledge foundation. Past studies are first summarized by application domain. Then, techniques used in detection, classification, and spatiotemporal analysis are reviewed in separate sections.

2.2 Applications

Automatic TSR has been employed in both real-time navigation systems and offline data analysis systems. In real-time systems, TSR serves as an additional eye of the system user as they approach and negotiate a signalized intersection. In offline systems, TSR provides additional horsepower for information extraction.

2.2.1 Real-Time Applications

Intelligent driving (both ADAS and AVS) and pedestrian assistance are two major real-time TSR applications. For intelligent driving, recognized traffic signals are used to reinforce driver perception, provide navigation and speed advisory, or directly control the car *(3–5, 20–41)*. For pedestrian assistance, the system is typically embedded in a mobile device to aid visual deficient to cross streets *(42–48)*. Some examples of real-time applications are given below.

Intelligent Driving

ADAS prototypes have been proposed to assist vision deficient drivers or provide driving advisory. Almagambetov et al. used a windshield mounted camera to assess traffic signal colors for drivers with color-vision deficiency *(21)*. In order to minimize color transmission errors, a camera that encoded colors natively in the Y'UV color space was used. Color ranges based on traffic signal design standards were used as a reference for color classification. Koukoumidis et al. proposed a coordinated ecofriendly driving advisory system for approaching signalized intersections *(3)*. Their system featured a windshield mounted smartphone running an application that exchange the recognized traffic signal state with other smartphones in the application network to predict future signal state. Based on the prediction, a safe and emission minimized speed control advisory was given. Camera exposure time was fixed on the smartphone to facilitate detection and

GPS sensor was used to trigger the application when the vehicle was in a 50 meter range of a signalized intersection.

Several groups aimed at fully autonomous driving. An initial attempt of TSR for autonomous vehicles was proposed and implemented by Lindner et al. in the IVI-Edmap research project *(41, 49)*. They employed both color-based and gray value based detectors for TSR, supported by a HDR camera and a VGA camera, respectively. In addition, differential GPS, enhanced digital maps, and structure-from-motion (SFM) techniques were mentioned as important aids to the TSR process. Shen et al. implemented their TSR system on an OSU-ACT autonomous vehicle platform using a commodity camera *(37)*. Similar studies were also conducted by Xu et al. and Guo et al. to accommodate complex urban scenarios in China *(20, 33)*. Fairfield and Urmson proposed their TSR system as a core component behind Google's autonomous driving car project *(4)*. Their system used a prior 3D map of traffic signal heads constructed during a trial run. Detection essentially became projecting the 3D target position onto the captured image according to the camera position, orientation, and intrinsic parameters. Images acquired using a fixed-exposure camera was primarily used to assess the signal color within the projected traffic signal head regions. Levinson et al. refined this approach but providing a probabilistic method to accommodate the 3D to 2D projection error due to data acquisition accuracy *(5)*.

Pedestrian Crossing

Mobile devices were exploited to assist vision deficient pedestrians to cross streets controlled by pedestrian signals. Shioyama et al. were among the earliest attempts *(48)*. Their system recognized pedestrian signals on the far end of the crosswalk and estimated the length of the crosswalk based on the pavement marking pattern and camera calibration parameters. However, they did not implemented the system on an actual mobile device. Eddowes and Krahe looked into

daytime and nighttime pedestrian signal recognition and implemented the system on a portable digital assistant (PDA) *(47)*. However, their method did not work well in daytime due to the failure of RGB based simple color segmentation. More reliable and real-time systems have been proposed by Roter et al. and Ivanchenko et al. using Nokia N95 mobile phones, but only daytime scenarios were considered *(44, 45)*. Mascetti et al. used more recent Android device to deploy their system which required advanced controls over the camera's ISO, aperture, and shutter speed according to the environmental light level *(42)*. Angin et al. considered using server side computation to improve the image processing speed of their cascade classifier based recognition system, which relied on the continuous availability of internet access for data transmission *(46)*.

2.2.2 Offline Analyses

Traffic surveillance, inventory, and safety investigation could also be facilitated by a TSR system *(1, 2, 4, 5, 50, 51)*. A few key studies in the literature are briefly listed below.

Surveillance

Surveillance, specifically red-light-running detection, has been the major offline application of TSR. Videos collected for this purpose were primarily recorded by a stationary camera mounted on the upstream roadside aiming at the traffic signals and the approaching traffic. Yung and Lai proposed a system that integrated the detection of traffic signal state and the estimation of vehicle movements at the stop bar to identify red-light-runners during daytime *(1)*. Chung et al. proposed to incorporate fuzzy logic in the detection stage and used average background extraction to constantly monitor and adapt to illumination changes *(50)*.

Inventory

On-board video or sparse photo sequence based highway inventory is a common practice in the United States. Photo logs or video logs have been used by states to geocode highway infrastructure such as traffic signs and guardrails and monitor pavement conditions. Tu and Li were among the first to propose using TSR for mapping traffic signals *(2)*. They used color and gradient histograms to detect traffic signal heads of four major perception angles. However, their system was focused on estimating the spatial relationship between the camera and the detected traffic signal heads and was not capable of classifying the traffic signal state. Fairfield and Urmson and Levinson et al. also constructed a 3D map of traffic signal heads using video data so such prior map could be used to facilitate real-time TSR in autonomous vehicles *(4, 5)*. However, constructing the 3D map involved a considerable amount of human efforts.

Incident Investigation

Responsibilities in traffic incidents need to be verified by solid evidence. Agencies such as insurance companies even offer rewarding driving trackers to their customers in order to collect actual driving data. In addition, with affordable dashboard cameras that are sometimes integrated with GPS navigation systems, drivers can also proactively monitor their daily commutes in preparation for incidents. Front-view videos collected under such circumstances become strong evidence during incident investigation. At signalized intersections, the traffic signal states can be the key of judging whether the driver was violating the traffic law. Yelal et al. proposed a recognition and tracking system to log traffic signal state from on-board videos for after crash investigation *(51)*. Unfortunately, their system was very preliminary and no performance measure was reported.

2.2.3 Summary

Examples of TSR applications in different domains are given and a more exhaustive list is summarized in Table 2-1. In addition to this list, there are other more general TSR studies focusing on experimenting specific methods on certain TSR stages rather than developing a complete system. They will be covered in later sections of this chapter where related. Among all these existing studies, TSR for incident investigation or more generally, evaluation of driving videos, is very limited. There are two potential reasons.

The first is that the analysis periods are short and could gain little benefit from a TSR program compared to manual reviewing. For example, in incident investigation, the video only needs to be analyzed within a short timeframe of the incident and human reviewers are typically sufficient and reliable. However, this might no longer be the case with the increasing availability of lengthy driving videos and the need to identify events without reported timestamps (e.g., near-misses). Manual review could be infeasible and need to be assisted, if not fully replaced, by an automatic procedure.

Another more fundamental reason of limited research into applying TSR on general purpose driving videos is the generic feature of the data and the recognition difficulty it raises. In other applications, like ADAS, the system typically had control over the camera in order to obtain data tailored to the need of TSR, such as calibrating the exposure time according to the traffic signal's light emission pattern so the target appeared consistently in various illumination conditions. Even when controls over the camera were not available, a considerable amount of sample data could be collected in advance using the same camera so calibration against the device could be done, such as finding the camera-dependent color ranges of various traffic signals (see Section 2.3.1). However, for a more generic data source, where neither the control over the camera

is possible nor the camera-specific sample data are available or sufficient, the above systems may fail to set up properly. Additionally, with videos collected using different cameras, the appearance of traffic signals can be extremely inconsistent, especially the color. Unpredicted illumination conditions and visibility of the environment can add to the complexity. As a result, traditional traffic signal detection features such as color ranges used in most existing studies can be too variant to calibrate or use. Low resolutions should also be expected from a generic data source. General purpose driving monitoring data are primarily aimed to provide human recognizable visual evidence rather than high definition images. Therefore, videos can come at relatively low pixel resolutions and even with motion blurs, which would render detection inaccurate. Previous TSR methods using edge-based shape detection may fail in such situation (see Section 2.3.2). Therefore, more generic detection features are needed. In the rest of this chapter, legacy methods for traffic signal detection, classification, and spatiotemporal analyses are reviewed.

Table 2-1 Overview of Major Existing Systems

Researchers	Camera	Other Data	Environmental Condition	Max. Sensing Range	Signal Colors and/or Shapes
Intelligent Driving					
Almagambetov et al. *(21)*	Y'UV camera Res. = 640x480	/	Day/night/dawn/dusk Sunny/cloudy	122 m	Green/red/yellow Circular/arrow
Guo et al. *(20)*	Res. = 1000x1000	Position Heading (optional)	Morning/afternoon/night	120 m	Green/red Circular/arrow
Diaz-Cabrera et al. *(52)*	CMOS camera Res. = 752x480 f = 8 mm Fixed shutter speed	GPS position (optional)	Day/night Sunny/snowy	115 m	Green/red/yellow Circular
Jang et al. *(22)*	High speed camera Max. FR = 100 FPS Res. = 640x480 Alternating exposure times	/	/	50 m	Green/red Circular/arrow
John et al. *(23)*	[unspecified]	GPS position Traffic signal locations and headings	Afternoon/dusk	100 m	Green/red Circular
Wang et al. *(25)*	Res. = 1292x964 f = 8.5 mm	/	Morning/afternoon Sunny/cloudy	90 m	Green/red Circular/arrow
Kim et al. *(26)*	Res. = 620x480	/	Night	/	Green/red Circular
Koukoumidis et al. *(3)*	Smartphone camera Fixed exposure time	Smartphone GPS reading	/	50 m	Green/red/yellow Circular/arrow

Table 2-1 Overview of Major Existing Systems (continued)

Researchers	Camera	Other Data	Environmental Condition	Max. Sensing Range	Signal Colors and/or Shapes
Intelligent Driving					
Cai et al. *(53)*	Res. = 1392x1040 f = 25 mm FOV = 20.4 deg. Fixed gain Fixe shutter speed	/	Sunny/cloudy Direct sunlight/backlighting	/	Green/red/yellow Arrow
Siogkas et al. *(28)*	Res. = 640x480 f = 12 mm	/	Day/night	/	Green/red Circular
Fairfield and Urmson *(4)*	Res. = 2040x1080 FOV = 30 deg. Fixed gain Fixed shutter speed	3D prior signal map constructed during a mapping trial	Morning/afternoon/night	200 m	Green/red/yellow Circular/arrow
Levinson et al. *(5)*	Res. = 1280x1024 Fixed gain Fixed shutter speed	3D prior signal map constructed during a mapping trial	Noon/sunset/night	140 m	Green/red/yellow Circular
Kim et al. *(29)*	HDR CMOS camera Res. = 620x480	/	Day	/	Green/red Circular
Gong et al. *(30)*	Res. = 780x580 f = 15 mm	/	Day	/	Green/red/yellow Circular
Yu et al. *(31)*	Res. = 680x480	/	Day/dusk/nigh	/	Green/red/yellow Circular/arrow
Nienhuser et al. *(32)*	Res. = 512x384	/	Day	/	Green/red/yellow Circular
Xu et al. *(33)*	Res. = 640x480	/	Day	/	Green/red/yellow Circular

Table 2-1 Overview of Major Existing Systems (continued)

Researchers	Camera	Other Data	Environmental Condition	Max. Sensing Range	Signal Colors and/or Shapes
Intelligent Driving					
Charette and Nashashibi *(34)*	[unspecified]	/	Day	/	[unspecified]
Park and Jeong *(36)*	CCD camera Res. = 320x240	/	Day Cloudy	/	Green/red Circular
Shen et al. *(37)*	Res. = 640x480	GPS reading IMU reading	Day	70 m	Green/red/yellow Circular
Joo et al. *(38)*	Res. = 640x480	/	Morning/noon afternoon/dusk	140 m	Green/red Circular
Kim et al. *(39)*	[unspecified]	/	Day/night Cloudy	100 m	Green/red/yellow Circular
Hwang et al. *(40)*	Res. = 720x480	GPS reading	Day	130 m	Green/red-yellow Circular
Lindner et al. *(41)*	HDR and VGA cameras f = 16 mm (color) f = 12 mm (gray)	Differential GPS position and heading (1 m and 1 deg. accuracy)	/	/	Green/red/yellow Circular/arrow
Mobile Pedestrian Guide					
Mascetti et al. *(42)*	Android mobile camera Res. = 2448x3264 Fixed ISO Fixed aperture Fixed shutter speed	Accelerometer and Gyroscope reading	/	/	Green/red/yellow
Ying et al. *(43)*	[unspecified]	/	Day/dusk	/	Green/red/yellow Round

Table 2-1 Overview of Major Existing Systems (continued)

Researchers	Camera	Other Data	Environmental Condition	Max. Sensing Range	Signal Colors and/or Shapes
Mobile Pedestrian Guide					
Roters et al. *(44)*	Nokia N95 Autofocus Camera Res. = 320x240	/	Day	/	Green/red Pedestrian
Ivanchenko et al. *(45)*	Nokia N95 Res. = 640x480	/	Day	/	White Pedestrian
Shioyama et al. *(48)*	Res. = 640x480 f = 5.9 mm	/	Day	/	Green/red Pedestrian
Surveillance					
Chung et al. *(50)*	Stationary camera Res. = 320x240	/	Day/night	/	Green/red/yellow Circular
Yung and Lai *(1)*	Stationary camera Res. = 640x480	/	Day	/	Green/red/yellow Circular
Inventory					
Tu and Li *(2)*	Res. = 720x400	/	Day	/	Only detect whole signal heads
Crash Investigation					
Yelal et al. *(51)*	Res. = 720x480	/	Day	/	/

2.3 Detection

Detection is the most fundamental and essential step of all TSR systems. Although certain prior knowledge, such as a 3D map of traffic signals and camera calibration data, have been used to directly locate traffic signals in the image without image processing *(4)*, such advantage is not commonly available in most cases. This section is focused on summarizing image based TSD.

2.3.1 Color Segmentation

Color segmentation was the most commonly used method in extracting image regions that are likely to contain target traffic signals and was often combined with other detection methods to locate traffic signal candidates. The general idea of color segmentation is to check for each pixel in the input image whether its color value falls into an empirical region in a chosen color space. A pixel can be labeled with a particular signal color during this process or it can be assigned a fuzzy membership score for each possible signal color. Figure 2-1 demonstrates this concept with an example image being filtered by three color histograms. Each histogram shows the 2D distribution of saturation and hue values of all sample pixels from a particular traffic signal color. For a traffic signal color, each bin's value of its 2D histogram is back projected onto the input image pixels whose saturation and hue values fall into that bin, forming a color membership score map where brighter regions are more likely to contain target traffic signals.

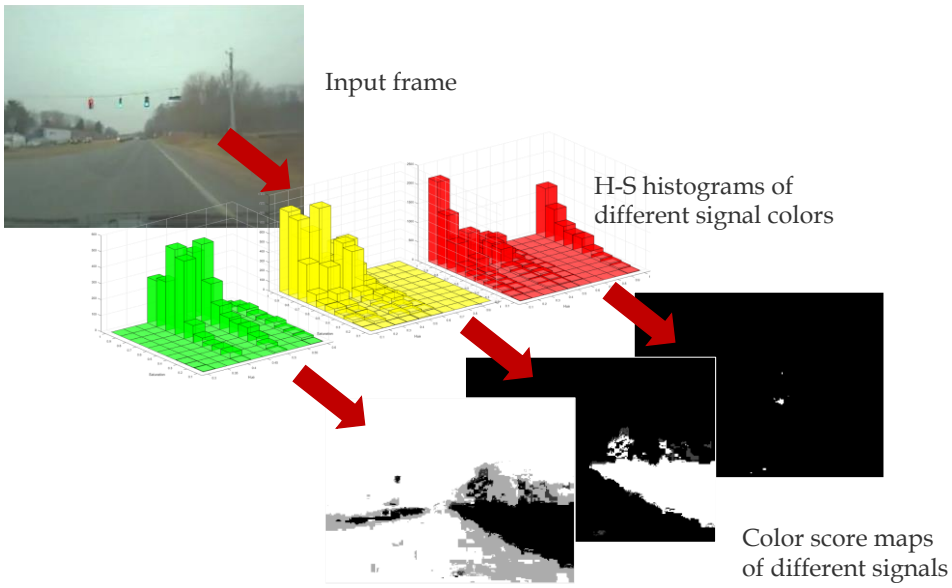


Figure 2-1 Demonstration of color segmentation based TSD.

In the literature, a number of variants of the color segmentation approach have been proposed. The major difference between these variants is the color coordinates being used.

RGB and Related Transformations

RGB values are the native color coordinates in most image data and were used by several past studies. Some studies calibrated the camera according to the traffic light emission pattern so direct RGB values were sufficient to distinguish traffic signals from the background and each other *(3, 4, 27, 54)*. However, camera control is not always possible and uncontrolled exposure can cause significant variation of the target's RGB values, resulting in less stable color ranges for segmentation. In order to overcome this issue, some researchers looked into transformations of the RGB coordinates. Joo et al. applied rotated principle component analysis on RGB and the gray scale values and used the first and second principle component images for segmentation *(38)*. However, their method lacked physical meaning and did not report good robustness. Roters, et al. also applied principle component analysis on sample RGB point cloud *(44)*. They found three

principle dimensions and trained thresholds to distinguish three signal colors and dark background color under the principle dimensions. However, their detection recall rates were low. In general, RGB based color segmentation is subject to color variation problems introduced by environmental illumination changes and needs camera exposure controls to mitigate the challenge.

HSV and Cylindrical Color Coordinates

The HSV color coordinates rearrange the RGB values in a cylindrical manner so chromatic information, i.e., saturation and hue, can be separated from the luminance, providing a more reliable way to compare colors without considering the lighting change. As a result, HSV has attracted some TSR researchers. Gong et al. trained statistical curves on all HSV coordinates based on samples with various lighting, background, and brightness conditions *(30)*. Wang, et al. applied principle component analysis on the 3D sample pixel cloud in the HSV space to find 2D principle components to distinguish green signals and red signals *(25)*. Under the principle component coordinates, the shortest distance from a testing pixel to sample pixels of a signal color was calculated and compared to a maximum threshold to decide whether the pixel belonged to the signal color. Jang et al. used a high speed camera with per-frame alternating exposure to capture both normal exposure and low exposure images, practically allowing any instance of a scene to have a two-level dynamic range *(22)*. Then static thresholds on all three HSV coordinates were trained using the low exposure images. Mascetti et al. used a fixed-exposure camera and trained empirical ranges on HSV values, among which only the hue channel showed different ranges for different signal colors and with the red signal range and yellow signal range overlapping *(42)*. To distinguish yellow and red, they compared the pixel counts of yellow and red in each candidate region. The signal color with the most pixel count won. Guo et al. trained the hue range assuming it followed a Gaussian distribution and combined the trained hue range with fixed minimum

thresholds of saturation and value to extract candidate pixels *(20)*. Similar to HSV are other cylindrical color coordinate systems such as HSL, IHSL, and HSI, they have also been used by some researchers in traffic signal detection *(37, 40, 50, 55–57)*. Although cylindrical color coordinates provide good separation between luminance and chromatic information, they are essentially linear transformations of the RGB color coordinates and do not provide additional power in separating colors with subtle difference but sensitive to human vision.

CIE Based Color Spaces

Color spaces defined based on the International Commission on Illumination (CIE) measurements are aimed to provide device independent color matching that approximates human color perception and were used in some of the previous TSR studies. These color spaces also separate luminance from chromatic information, except that the chromatic plane is not a radial model of the hue and the saturation as in HSV, but are measurements of the relationship to certain color primitives. For example, in the CIELab space, the "L" represents the lightness value (i.e., luminance) and the "a" and "b" coordinates correspond to the relative positions of a color between two pairs of opponent color primitives, respectively. Siogkas et al. multiplied "L" to the summation of "a" and "b" to form a feature map *(28)*. Fast radial symmetry transform was used to locate peaks corresponding to green and red traffic signals using this map. Similarly, John et al. used a multiplication of the gray scale value, the "a" value, and the saturation value of HSV as a feature map *(23)*. Since the "a" value contained a sign, positive pixels on this map became indications of red traffic signals and the negative pixels implied the possible regions of green traffic signals. Tu and Li constructed three-dimensional color histograms using the CIE 1976 (L*,u*,v*) color space for both template signal head images and test image patches *(2)*. The similarity between the color histograms was used in conjunction with the similarity of edge gradient histograms to generate a

similarity map. However, their templates were signal heads with yellow backplates perceived from front, sides, and back with no active signal lights, so the color feature was related primarily to the signal head rather than individual lenses, and could not be used to classify traffic signal state. Other CIE XYZ color space variants, such as the YCrCb color space and the Farnsworth's perceptually uniform color system (UCS), have also been employed *(29, 48)*. However, transformation from RGB color values to the above color spaces requires a reference white color, because RGB is device dependent. In addition, transformation to these color spaces may still retain RGB color encoding errors. A recent study by Almagambetov et al. reported state-of-the-art detection performance by using natively captured Y'UV color space values and corresponding U-V plane regions defined according to the Institute of Transportation Engineers (TIE) and the British Standards Institute (BSI) standards *(21)*. Unfortunately, they relied on the minimized color coding errors of an Y'UV camera, which is not commonly available in most TSR systems.

In general, color segmentation is an efficient way to identify candidate signal regions, but its performance can be sensitive to the empirical parameters, whose reliability depends on both the quantity and quality of sample data. As a result, other researchers tried to explore camera independent features for traffic signal detection.

2.3.2 Texture and Shape Detection

Texture and shape are geometric features that are independent of camera's color formation process and provide a more generic way for detection. In the literature, edge based shape detection, template matching, and structure element based morphological operations are typical approaches towards texture or shape recognition.

Edge Based Detection

Traffic signal lenses are often assumed to preserve their round shapes in the image with their edge pixels closely following the circumference of a circle. Under this assumption, edge pixels are first extracted based on pixel gradients. Various edge detection methods, such as the Canny edge detector, the Sobel edge detector, and the Laplace edge detector, have been used by previous researchers *(3, 41, 56, 58)*. In order to find circles that are well represented by the edges, the Hough transform was often used *(3, 41, 58)*. Chiang et al. proposed an ellipse detection method based on genetic algorithms to generalize circle detection in the presence of perspective distortion *(56)*. Unfortunately, the above studies in fact performed edge detections on the color segmented image rather than the original image, which inherently includes the problems faced by color segmentation. In a slightly different way, Gomez et al. applied border following algorithms on edge pixels of the original image to find rectangular regions corresponding to traffic signal faces *(59)*. Issues were found with false detections on other rectangular areas such as the spacing between roadside trees.

Edge based shape detection can be the most color independent approach, given the edges are extracted from the original image rather than a color segmented map. However, without the gauge of color segments, the edges in the image can be extremely noisy and can trigger a considerable amount of false circle detections, simply because the edge pixels of different objects happen to align on most part of a circle's circumference. Also, for true circular objects, when the pixel resolution is not sufficient, the circular geometric pattern of the edge can be corrupted, resulting in false negatives. In summary, edge based shape detection needs clean edge data with decent resolution to work properly, while the shape detection algorithms are also computationally intensive.

Template Matching

The texture of the traffic signal face, e.g., a dark horizontal or vertical rectangular region with a bright circular area in one of several key positions, is distinct to human drivers and useful for detection. Linder et al. employed the AdaBoost framework to train a cascade classifier for different states of a three section vertically arranged signal face and used it to detect traffic signal heads in gray scale images *(41)*. However, since the texture is the reflection of the traffic signal section arrangement, one classifier should be trained for each arrangement type to cover all possible cases. In addition, such machine learning framework was found to introduce little detection improvement compared to image processing while increasing the computational load *(34)*. The texture based approach has two other major theoretical issues. The first is the nighttime condition when the rectangular backplate region merged into the dark background. The second is partial occlusion to the signal face while the signal light itself is visible. In this case, the texture is incomplete. Partial occlusion can occur by blockage from other objects or because part of the signal face is out of the view (e.g., exceeds the top edge of the image as the vehicle drives under and traffic signal).

Spot-Light Detection

So far, the spot-light detection based approach is arguably the state-of-the-art generic detection algorithm for traffic signals. Charette and Nashashibi was among the first to propose using spot-light detection in natural driving environment for traffic signal recognition *(34, 35)*. Spot lights were detected using a morphological white top-hat algorithm on gray scale images to identify signal lenses. A fixed size (11-by-11 pixels) structure element was used in the top-hat operation. This approach, assisted with their adaptive template classification framework, was reported to match a machine learning based alternative. However, their fixed structure element

size prevented them from accommodating various sizes of target and the gray scale image does not provide color saturation information which could be very helpful in distinguishing signal lights from other white lights, as will be incorporated in the proposed methodology.

2.3.3 Summary

Various detection methods were surveyed in the literature, covering both 1) color segmentation based methods that depend on camera-dependent empirical parameters and 2) texture or shape detection based methods that are more camera independent. The color segmentation methods heavily rely on the quality and quantity of sample data to properly train their empirical parameters. When a different camera is used, these methods need to be recalibrated. In addition, variation in lighting conditions can affect the reliability of the empirical parameters, which would require exposure control on the camera to compensate. Texture and shape based methods are more independent of cameras. Edge based shape detection methods solely rely on the geometric information of edges, but are prone to edge noise and compromised target resolution. Texture based template matching incorporates shape information in a more robust way, but is vulnerable to nighttime conditions and partial occlusion. Spot-light detection is arguably the state-of-the-art generic algorithm for locating traffic signals. However, existing implementations used fixed-scale morphological top-hat operations on gray scale images to identify spot-lights, which could not accommodate various target sizes and could theoretically lead to confusion between traffic signal lights and other non-colorful light sources.

2.4 Classification

Detection produces a list of candidate regions that will be classified to a particular traffic signal color or even a none-traffic-signal. Note, in some studies, classification also included the decision of the signal's shape (i.e., arrow or circle), which is out of the scope of this study and will not be further discussed below. Classification is only performed within each candidate region, at most with a buffered margin included to bring a relevant neighborhood context. In other words, classification is a local operation to label a patch of the input image. Color and position are the two major clues for classification.

2.4.1 Color Based

When properly calibrated, empirical color ranges not only provide an efficient filter for traffic signal detection, but also reliably classify the traffic signal color on the fly. Classification was inherently done during detection in most of the studies that performed color segmentation *(1, 3, 25, 27, 28, 31, 36, 37, 39, 40, 44, 48, 50, 52, 55, 58, 60).* Some studies followed texture or shape based detection methods with color based classification *(43, 59).*

Simple thresholds are sometimes not sufficient to distinguish between similar traffic signal colors and additional decision rules are needed. For example, red and yellow traffic signals have been noticed easily confused in certain color coordinates *(42).* Gomez et al. compared the numbers of pixels belonging to the overlapping yellow and red signal color ranges to decide which signal color was more likely to be true *(59).* Almagambetov et al. used sequential rules that separated green from other signal colors by the value coordinate of HSV and then separated red from yellow by the hue coordinate of HSV *(21).*

While color based classification is the most effective approach, it may be unreliable when the empirical parameters are not properly established, as in the detection stage. With that concerns, some researchers explored the relationship between signal colors and their relative positions in the signal face to determine the color, which applied also on gray scale images.

2.4.2 Position Based

When a detected candidate region represents a signal lens, a common approach of position based classification is to assume the candidate at different positions of a particular face arrangement and check for the best hypothesis. Under a position hypothesis, the expected whole signal face region is cropped out and tested regarding its texture. Some researchers trained a support vector machine to classify the texture according to the image patch's histogram of gradients (HOG) *(22, 32)*. Cascade classifiers trained using AdaBoost on Haar features have also been used to classify the texture *(30, 34)*. Lindner et al. employed a feed-forward neural network to test the position hypotheses *(41)*. When the whole signal face instead of the active lens is detected, the position check becomes more straightforward *(43)*.

Although position based classification is robust against color distortion or variation, it needs to consider various signal face arrangements to be comprehensive. Even when all possible arrangements are considered, confusions can still occur when the mapping relationship between position and color is not one-to-one. For example, in Figure 2-2, two 4-section vertical arranges that are commonly used in the United States present different colors at the second and the third positions. In addition, nighttime, low resolution, partial occlusion, and perspective deformation could affect the perception of the entire signal face and hence the accuracy of position estimate.

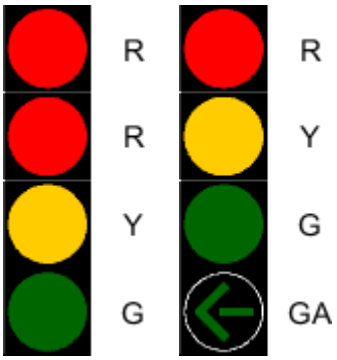


Figure 2-2 Two 4-section vertical arrangements with different color positions.

2.4.3 Aggregated Features

Due to the respective limitations of color and position based classifications, some researchers resort to holistic aggregation of both. Charette and Nashashibi proposed an adaptive template that modeled the entire 2D traffic signal structure as hierarchical components, each of which contributed to the final classification score *(34)*. This template allowed programmers to specify the contributions of color, shape, and position of each component using their weights. However, establishing such a template can be complicated and one template can only cover a particular signal face arrangement. Convolutional neural network (CNN) in fact provides a way of classifying different signal face arrangements in a uniform framework. However, it has only been used for traffic signal classification by John et al. on color features *(23)*. There are other reasons why CNN based traffic signal classification is rarely visited. First, traffic signals are feature poor. Color, shape, and contrast almost exhaust the feature list of a traffic signal head and none of these features are extremely unique to traffic signals. The power of CNN is its capability of automatically identifying distinct and consistent features of a target object out of a large training sample set. With limited and inconsistent features and the easy resemblance by other road objects, traffic signals may benefit little classification advantage from CNN compared to a knowledge based approach. In addition to the possible marginal gain, the requirement of large and representative

training data introduces considerably higher cost of using CNN. Further, a CNN classifier is a black-box model whose explanatory ability is weak. It is hard to explain what contribute to the recognition process based on weights given to nodes in each neural network layer. Last but not least, the training process of CNN is slow. This makes model recalibration time consuming.

2.4.4 Summary

Classification of traffic signal colors can be based on color information, relative lens position, or both. Color based classification is efficient but sensitive to the color variation and empirical parameter quality. Position based classification is robust against color variation but requires decent perception of the entire traffic signal face. When both are used, they compensate each other's limitations but at the same time raise the question of which contributes more to the final classification score. Solutions have been proposed by hierarchically organizing traffic signal components into an adaptive template and manually assigning weights on each component for their color, shape, and position. However, this approach is inflexible to adapt to various signal face arrangements. CNN in another alternative for classification using aggregated features, but the trained model is black-boxed and its transferability between datasets is hard to justify. In addition, CNN normally requires large training data to reach a stable model.

2.5 Spatiotemporal Coordination

Spatiotemporal information can facilitate traffic signal recognition in many ways for video analysis. First of all, with a lengthy video in which signalized intersections were passed occasionally, a majority of the frames do not even capture any traffic signal and most of these frames can be effectively excluded with spatial information such as GPS data and a map of signalized intersections. Second, the positions of a traffic signal in successive frames are not independent from each other and are generally close to each other. Accordingly, if the traffic signal is detected in one frame, its position in the next frame can be roughly bounded, which will help to narrow down the detection region. Last but not least, the change of traffic signal color follows a fixed sequence and can be used to prune the classification results across multiple frames. Some existing methodologies related to spatiotemporal coordination are reviewed.

2.5.1 Activation Range

Distance to a signalized intersection is a common trigger for traffic signal recognition systems. Shen et al. used GPS and IMU to estimate rough distance to a signalized intersection and initiated detection at about 70 m upstream of the intersection *(37)*. Levinson et al. and Fairfield and Urmson used similar equipment to initiate detection up to 140 m and 200 m upstream, respectively to account for normal driver stopping sight distance *(4, 5)*. A kd-tree search algorithm was employed by Fairfield and Urmson to quickly find the closest intersection (not necessarily a signalized intersection) to the current position of the vehicle and camera before a prior 3D map of individual traffic signals were built *(4)*. Koukoumidis et al. used the smartphone GPS to initiate their system within 50 m of a signalized intersection *(3)*. To the best of the author's knowledge, the spatial information has not been used in offline traffic signal recognition involving lengthy videos. More efficient search algorithms can be applied with the simultaneous availability of all

frames' positions (in contrast to sequential availability in real-time applications) and detection can start from a downstream location backward with close frames serving as a more reliable starting point.

2.5.2 Candidate Association

Associating or tracking candidates was found to be effective in filling a small amount of detection gaps and suppressing a considerable number of false detections *(41)*. Association can be done purely on the 2D image plane or can be facilitated by 3D data when available. In a 3D space, when the position and orientation of the camera is available, target signal positions can be more reliably predicted on the next frame *(4, 5, 32, 52)*. However, 3D information is not always available. In the absence of 3D data, tracking can only be done based on 2D image data. Fortunately, a good range of tracking methods are available. Roters et al. applied Kanade–Lucas–Tomasi (KLT) feature tracking for pedestrian signal detection *(44)*. However, for traffic signals, whose shapes are much simpler than a pedestrian signal, KLT features are limited. Other researchers applied the continuously adaptive mean shift (CAMShift) algorithm to track candidates based on their color histogram *(5, 30)*. This approach works under the assumption that the color distribution of the tracked candidate changes slowly. It will fail when the signal color suddenly changes from one color to another. Some researchers looked into using the Kalman filter to stabilize tracking even in face of occlusion, but the linear motion model they used did not correctly reflect how traffic signals moved in the image as the vehicle approached *(21)*. Sudden signal color change (and hence position change) will also be missed by a Kalman filter. In contrast to the above point or region trackers, dense optical flow, which describes the pixel-wise motion between two images, has never been used in the literature for traffic signal tracking. The advantage of dense optical flow is that it can be done on gray scale images with robustness to color variation and when occlusion or sudden

disappearance happens to a region, the movement of its neighborhood can still be reliably estimated and used as a reference to justify the occluded region's movement. Also, multiple regions are tracked simultaneously in one flow calculation.

2.5.3 Pruning

The Hidden Markov Model (HMM) has been used by researchers to assist current classification by considering classification history *(32)*. The actual signal color was considered a hidden state in the HMM process and the initial classification was considered a measurement of that state. The hidden state was estimated based on the measurement and the consideration of the classification history and the fixed signal color rotation order. However, the study assumed a reliable tracking of the candidate. When the tracking is uncertain, the HMM smoothing will adversely introduce addition complexity to the temporal analysis.

2.5.4 Summary

Spatiotemporal information has been used in different ways to facilitate traffic signal recognition. Distances between frames and traffic signals are effective filters for candidate frames. However, the past studies only used distances as an activation range in online applications. For offline data extraction, the distance information can be retrieved in a faster way and used to coordinate TSR in a manner that maximizes the recognition possibility without following the sequential time order. No past study has looked into that opportunity. When tracking is considered to facilitate TSR, existing studies used methods that rely on either rich geometric features or stable color features to associate detected candidates across frames. These methods are either not sufficient to traffic signals or may fail in case of signal color change between frames. A more robust tracking approach should be explored, especially in case of multiple traffic signals appearing at the same time. Last but not least, traffic signal sequence could be pruned according

to the color change order and the confidence of candidate classification. Past studies reported positive effects of certain smoothing methods, but assuming that the candidates were already associated in a track. No study investigated how the color change order and classification confidence could be cleanly integrated into the association process to improve the accuracy of tracking.

2.6 Summary

This chapter surveys past studies on vision based TSR from different perspectives and identifies several research gaps. From the application point of view, research using TSR for massive driving data reduction has never been reported. Possible reasons include the absence of massive driving videos and hence the need for automatic data reduction and the difficulty introduced by uncontrolled data quality and scene complexity. As naturalistic driving studies are gaining popularity and producing millions of hours of video data, the first reason no longer holds. Consequentially, it introduces the need and opportunity to explore methods that accommodate the data quality and scene complexity. With such application setting, many preconditions required by past TSR systems are invalidated, such as the accessibility to camera controls, the availability of sizable device dependent training data, comprehensive prior knowledge of the scenes, high resolution videos, etc. As a result, the following research gaps need to be filled:

- In terms of traffic signal detection, many previous methods would fail without proper training data or detailed texture information in the image. Spot-light detection on gray scale images is more generic, but its lack of consideration of the color saturation effect could theoretically lead to confusion between traffic signal lights and other non-colorful light sources. A generic detection method that makes better utilization of information rather than grayscale values is needed.

- In terms of classification, the color based approach makes the best tradeoff between computational cost, classification accuracy, and robustness. However, with empirical range based hard classification, the result can be sensitive to the representativeness of the training data. A color classifier that makes decision based on the relative similarities of a candidate to three possible signal colors is theoretically more robust and needs to be investigated.
- As an offline application, spatiotemporal information can be retrieved and used in innovative ways to facilitate TSR. Past studies sequentially processed frames in an online workflow after a distance range was reached. In an offline setting, the distance information can be calculated faster without the constraint of sequential order, which is worth investigation. Also, with the flexibility of processing frames in any order, it is more intuitive to start TSR on frames within a more reliable distance range and use the stabilized results to facilitate TSR in distant frames.
- Past studies using temporal tracking and pruning to facilitate TSR reported positive outcomes. However, their tracking methods may fail in the presence of low resolution data or instant signal color change. Existing tracking methods are also confusable in the presence of multiple resembling traffic signals. Dense optical flow provides a more robust tracking feature in the above situations and is worth exploiting for tracking traffic signal candidates. In relation, a clean way of integrating the constraint of signal color change order and classification confidence into the tracking process is needed.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Overview

A conspicuity based generic traffic signal detector is proposed together with a color histogram based signal color classifier, composing a TSR module for processing individual images or video frames. The conspicuity based detector is the core of the methodology. Conspicuity is defined and modeled according to a scientific hypothesis about why humans can easily perceive traffic lights from images without reliance on precise color information. It assesses each pixel's likelihood of being the center of a traffic signal lens. Conspicuity is modeled as a weighted geometric mean of multiple convolutional features and gives an invariant ranking to all pixels with respect to any value scale change in these features. A candidate localization algorithm is developed to extract traffic signal candidates with proper positions and sizes using the conspicuity map. The color classifier trained a 2D histogram of the "a" and the "b" channels of the CIELab color space for each traffic signal color. For a detected candidate, the same type of histogram is calculated and compared to the trained histograms. A similarity score is defined as the normalized complement of the hyper angle between the candidate's histogram and a trained histogram, giving a measure of the classification confidence.

In the situation of processing lengthy driving videos, a spatiotemporal framework is proposed (Figure 3-1). The framework uses the GPS position data of the frames in the video and a map of signalized intersections. In practice, the frame positions are normally interpolated using relatively sparse GPS readings. OpenStreetMap™ is selected as an on-demand mapping of signalized intersections. The framework first checks the vicinity of each frame to a signalized

intersection and recognizes clips of frames corresponding to actual traversals through a signalized intersection. In addition, the vehicle movement in each clip (i.e., left-turn, thru, or right-turn) can be classified based on the deflection of the frame trajectory. A key frame, called anchor frame, is defined as the closest frame to a signalized intersection and, with reference to this frame, a two-stage temporal TSR coordination is performed. At the short range initialization stage, TSR is separately conducted on frames within a short trajectory distance from the anchor frame. Also, candidates in the current frame are associated with temporal tracks in the history. Stable tracks are selected and pruned. At the long range tracked recognition stage, frames beyond the short range are processed, but detection is restricted in an estimated region of interests (ROI) predicted by each of the pruned tracks. Both stages rely on a dense optical flow based method to predict the position and size of each candidate in an adjacent frame. The employed dense optical flow engine is robust against the change of signal color and is able to accommodate the situation when the candidate crosses the image boundary and disappears from the view.

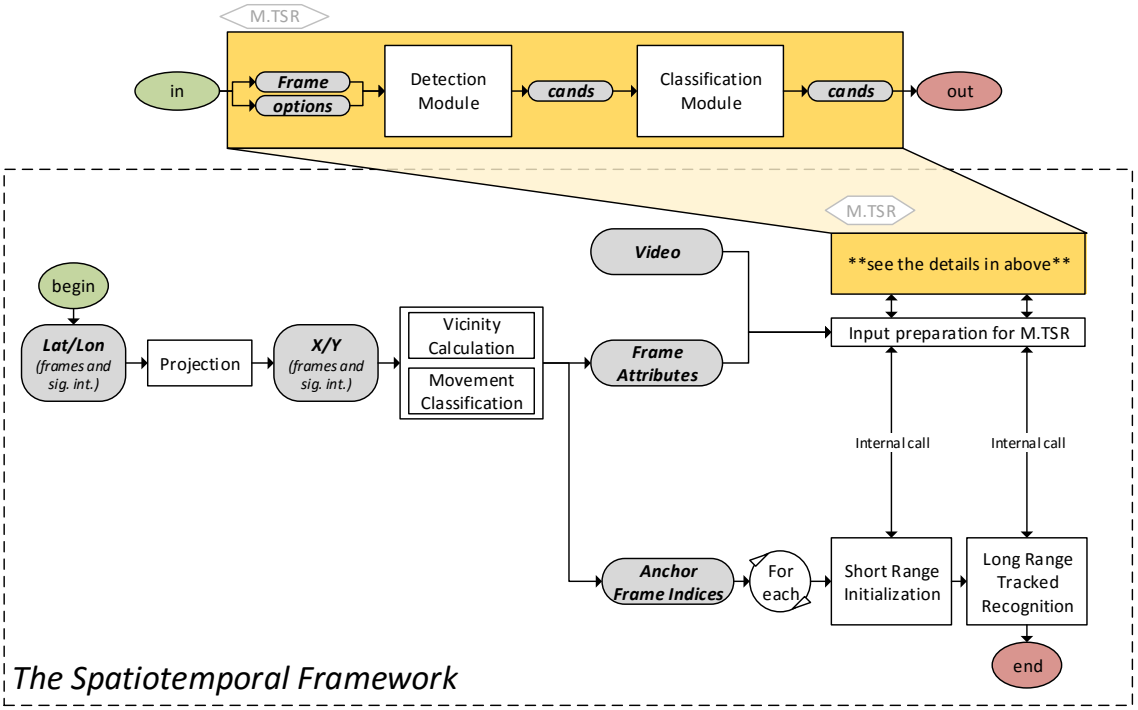


Figure 3-1 Methodology overview.

3.2 Candidate Detection

In this section, the concept of conspicuity is defined for each pixel of the input image (or a region of interest in it) in order to reveal the areas where active traffic signal lenses can be. A set of localization procedures are then used on this conspicuity map to extract traffic signal candidates. Several filtering criteria are applied on each candidate to exclude false positives. Resulted candidates are further classified for their signal color with color information (Section 3.3).

3.2.1 Conspicuity Map

Human eyes can easily identify active traffic lights from videos even when signal colors are distorted. Our brains' ability in structuring the roadway scene and restricting target search in relevant regions is certainly a major contributor. For example, in Figure 3-2, the image on the right is a shuffled version of the image on the left. Traffic signals can be easily identified in the original image. However, it may take more time and focus to find the same lights in the shuffled image, because the structure of the scene is destroyed and the targets can be anywhere in the image. Even with the state-of-the-art semantic segmentation methods (e.g., *(61)*), the efficiency and accuracy of analyzing the scene structure by human are not yet fully transferable to computers, not to mention the added expensive computational overheads.

Figure 3-2 Illustration of traffic signal identification with and without scene structure.

Nevertheless, traffic lights are distinct road elements that catch human's attention even in adverse conditions when the structure of the scene is ambiguous (e.g., in raining or snowy days), leading to the proposal of the concept of conspicuity in this study. Consider an example traffic light illustrated in Figure 3-3 with limited resolution and noisy color appearance. If we define a disk area A1 centered at *p(i, j)* with a radius of *r* pixels roughly over the lens region and a border area A2 between A1 and A1's bounding box, the most attractive feature to human eyes is arguably the high brightness of A1 as well as its contrast to that of A2. Such attractive property is considered the *conspicuity*.

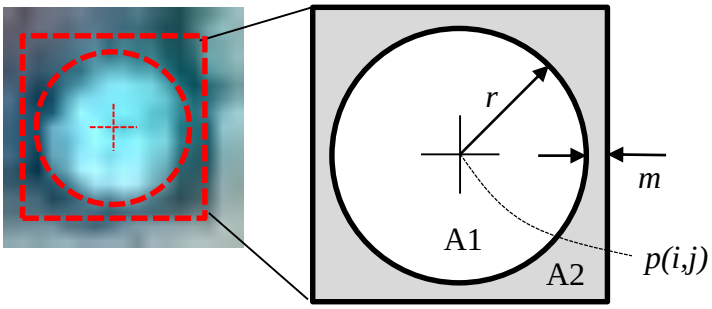


Figure 3-3 Illustration of the basic conspicuity concept.

According to color science research, human's perception of brightness is not only related to how much light the observed surface emits and/or reflects (luminance) but is also related to the purity of the surface color in contrast to white (saturation) *(62)*. This is called the Helmholtz-Kohlrausch (HK) effect. According to this effect, even if a light meter measures the same amount of luminance from both a color surface and a white surface, the color surface will still appear brighter to normal human vision than the white surface does. Similarly, a surface with more color saturation appears brighter under the same lighting condition. An example of this effect is illustrated in Figure 3-4. In the left patch, the red tile may look much brighter than its neighbor tile of another color. However, when both tiles are converted to grayscale on the right to roughly

present their luminance, the brighter looking red tile becomes darker than the other tile. Therefore, when modeling the brightness of digital image pixels, both color luminance and saturation should be considered to approximate human vision.

Figure 3-4 Illustration of the Helmholtz-Kohlrausch effect.

A model of conspicuity is proposed in Equations 3.1 - 3.2 to simulate how traffic lights in an image are perceived by a human reviewer. Equation 3.1 defines the conspicuity value of a particular pixel in a given image, assuming that it is the center of a potential signal lens with a radius of *r* pixels. Basically, this equation calculates the weighted geometric mean of three features. The first and the second features are based on pixel lightness and account for the effect of luminance. The third feature, which is in itself a maximum of two sub features, accounts for the effect of saturation. Details of each feature will be explained in the following subsections. While the brightness is jointly measured by the first and the third features, the brightness contrast is only measured in the second feature. The reason for not including saturation in the contrast measure is that the border area A2 is not necessarily low in saturation in situations like nighttime (light diffusion and “halo” effect) or yellow signal boxes. Equation 3.2 aggregates conspicuity values across a range of radii to account for target size variation.

$$C_{i,j\,|\,r} = \sqrt[w_L+w_{LC}+w_S]{LD_{i,j\,|\,r}^{w_L} \times LC_{i,j\,|\,r}^{w_{LC}} \times \max\left(SD_{i,j\,|\,r}, SA_{i,j\,|\,r}\right)^{w_S}} \quad (3.1)$$

$$C_{i,j} = \max\left(C_{i,j\,|\,r} \mid r \in RAD\right) \quad (3.2)$$

where,

$C_{i,j\,|\,r}$ = The conspicuity of pixel *p(i, j)* corresponding to a potential lens radius of *r*,
$LD_{i,j\,|\,r}$ = the average lightness in A1,
$LC_{i,j\,|\,r}$ = the contrast of average lightness between A1 and A2,
$SD_{i,j\,|\,r}$ = the average color saturation in A1,
$SA_{i,j\,|\,r}$ = the average color saturation in an annulus area centered at *p(i, j)* with an inner radius of *r* pixels and an outer radius of *2r* pixels. This accounts for the "halo" effect in nighttime and will be explained later,
w_L = a positive weight of conspicuity contribution from $LD_{i,j\,|\,r}$
w_{LC} = a positive weight of conspicuity contribution from $LC_{i,j\,|\,r}$
w_S = a positive weight of conspicuity contribution from $SD_{i,j\,|\,r}$ and $SA_{i,j\,|\,r}$
$C_{i,j}$ = the maximum conspicuity of *p(i, j)* among a set of potential lens radii,
RAD = a set of potential lens radii.

An argument should be made about favoring a weighted geometric mean over a weighted arithmetic mean in the model of Equation 3.1. The first and most intuitive motivation is that the geometric mean reflects an "and" relationship between the averaged variables more accurately than the arithmetic mean does. For example, when one of the three features in Equation 3.1 is zero, the resulting conspicuity is suppressed to zero, which is desired. If an arithmetic mean was used, the resulting conspicuity would have been less suppressed. A second and probably more important reason for using the geometric mean is that the ratio between conspicuities of any two pixels is invariant to any value scale change of the underlying variables. A proof is given in Appendix A. Such ratio invariant is important because it means, regardless of what value scale each variable chooses, the relative conspicuity between pixels will remain the same. Conspicuity peaks are always peaks in spite of any changes to their absolute values. Another advantage of using geometric mean is that the contribution to conspicuity of each feature is insensitive to the absolute value of its weight (or exponent), but to how its weight compares to that of the other features. If

all weights are the same, all features contribute the same; a feature with a larger weight always contributes more than one with a smaller weight does. This provides an intuitive control over the contributions of each feature. In the contrary, in an arithmetic mean model, two features with the same weights but different value scales contribute differently. In spite of the above advantages of using a geometric mean model, regression (or weight training) is less convenient for geometric mean due to its nonlinear form. Luckily, this can be easily overcome by applying a logarithm operation on both sides of Equation 3.1 and transform it into a linear model.

Before each feature is further explained, sample conspicuity maps are demonstrated in Figure 3-5 for different signal states. Note the coexistence of multiple traffic lights with varying distances. The centers of active signal lights are always among the bright spots in the conspicuity map, if not the brightest. Since a conspicuity map is a soft voting of candidate centers of traffic signals, the conspicuity value decreases as the distance from the actual center increases. In addition, other objects in the scene that resemble traffic signal sections in terms of brightness and contrast are also given high conspicuity values, such as the holes between tree leaves and vehicle headlights, but normally not as high as those around the actual center pixels of traffic lights. In Section 3.2.2, denoising and localization procedures will be introduced.

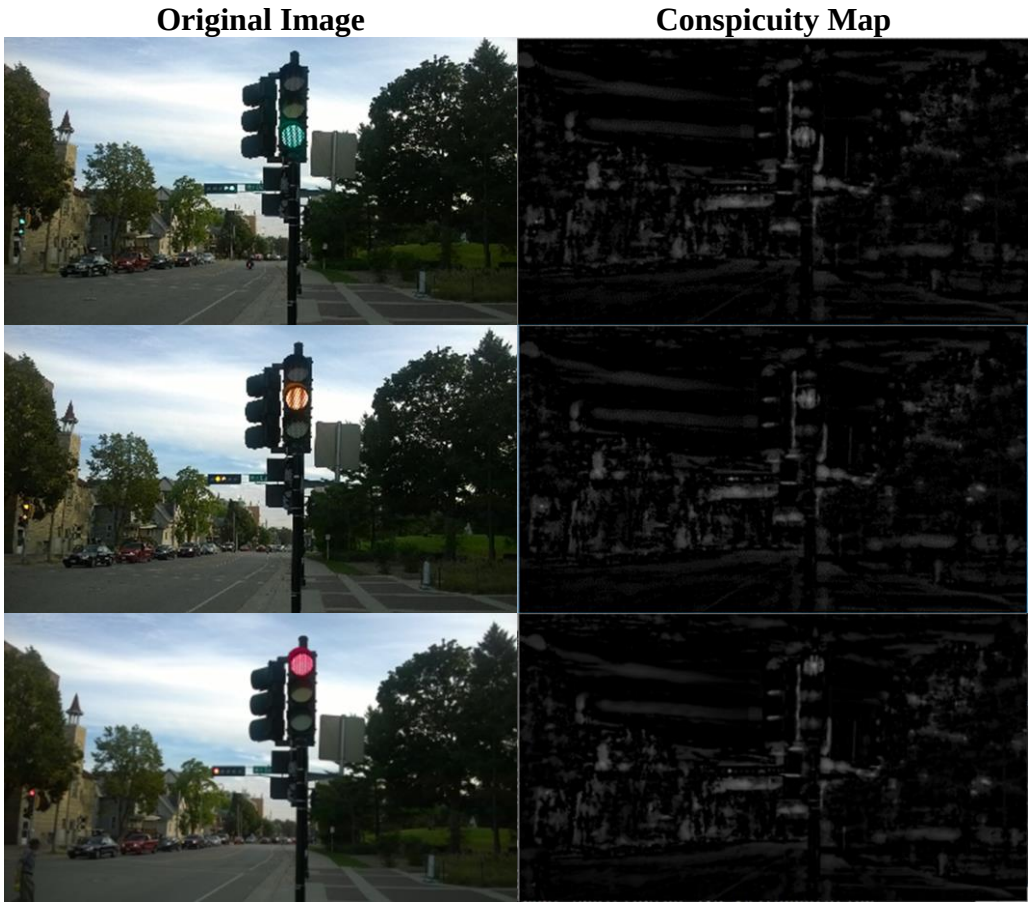


Figure 3-5 Demonstration of the conspicuity maps on sample images.

Average Lightness

The average lightness, $LD_{i,j\,|\,r}$, is the arithmetic mean of lightness of all pixels within a distance of r from the subject pixel $p(i, j)$ (see Equations 3.3 – 3.4 and Figure 3-7). Many color models offer a channel that measures lightness, such as CIELab, HSL, and HSV. The L channel of CIELab is chosen for measuring the lightness and scaled to [0, 1]. The reason of choosing CIELab is because the lightness is independent from saturation under this color model and prevents the resulting $LD_{i,j\,|\,r}$ from being affected by the saturation attribute. Such independence is illustrated in Figure 3-6 with comparison to HSL and HSV. Colors covered by a RGB model are converted to HSL, HSV, and CIELab in this figure. As can be seen, HSL and HSV embed bilinear

and linear relationships between lightness and saturation, respectively. In contrast, every lightness level in CIELab (imagine a plane perpendicular to the lightness axis) can cover the whole range of saturation for a set of colors.

$$LD_{i,j\,|\,r} = \frac{\sum \sigma_{i',j'\,|\,r} \times L_{i',j'}}{\sum \sigma_{i',j'\,|\,r}} \qquad (3.3)$$

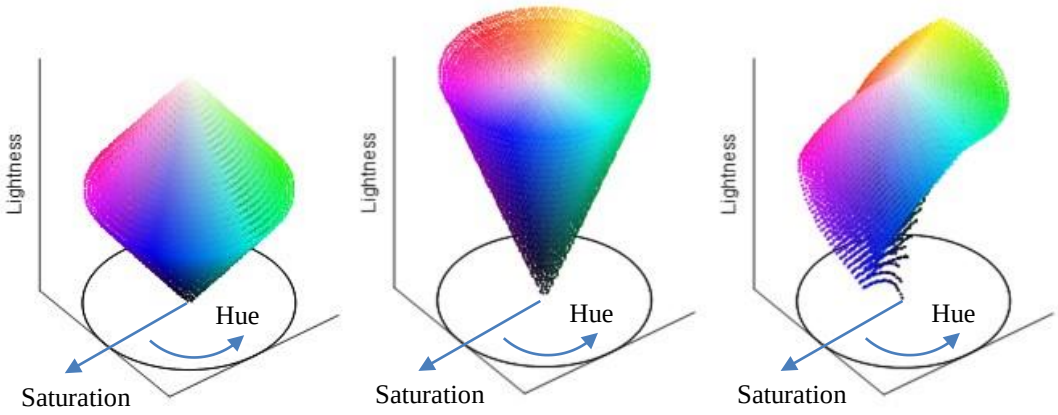
$$\sigma_{i',j'\,|\,r} = \begin{cases} 1, & if\ |p(i,j), p(i',j')| \le r \\ 0, & otherwise \end{cases} \qquad (3.4)$$

where,

$LD_{i,j\,|\,r}$ = the average lightness in A1,

$L_{i',j'}$ = the lightness value of pixel *p(i', j')*,

$\sigma_{i',j'\,|\,r}$ = the flag indicating whether pixel *p(i', j')* is in A1.



HSL
(Lightness is the L channel)

HSV
(Lightness is the V channel.)

CIELab
(Use HSL for saturation and hue. Similar plot when HSV is used)

Figure 3-6 Comparison of HSL, HSV, and CIELab.

Lightness Contrast

The lightness contrast, $LC_{i,j\,|\,r}$, is the difference of average lightness between A1 in A2 in Figure 3-3. However, inspired by the directional template used by Lindner et al. *(41)*, the actual calculation is slightly more sophisticate to enforce a good contrast in all four quadrants, as formulated Equations 3.5 – 3.9 and illustrated in Figure 3-7. In Figure 3-7, each quadrant area is bounded by a red bounding box and the four quadrants overlap on the row or the column of the subject pixel *p(i, j)*. Any negative difference in a quadrant is rounded up to zero.

$$LC_{i,j\,|\,r} = \max\left(LC_{i,j\,|\,r\,|\,k} \;\middle|\; k \in \{1,2,3,4\}\right) \quad (3.5)$$

$$LC_{i,j\,|\,r\,|\,k} = \max\left(0, LD_{i,j\,|\,r\,|k} - LB_{i,j\,|\,r\,|k}\right) \quad (3.6)$$

$$LD_{i,j\,|\,r\,|k} = \frac{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} + \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right) \times L_{i',j'}}{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} + \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right)} \quad (3.7)$$

$$LB_{i,j\,|\,r\,|k} = \frac{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} - \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right) \times L_{i',j'}}{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} - \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right)} \quad (3.8)$$

$$\sigma_{i',j'\,|\,r\,|\,k} = \begin{cases} 1, & if\ |p(i,j), p(i',j')| \le r\ and\ p(i',j')\ is\ in\ quadrant\ k \\ 0, & if\ p(i',j') is\ out\ of\ quadrant\ k \\ -1, & if\ |p(i,j), p(i',j')| > r\ and\ p(i',j')\ is\ in\ quadrant\ k \end{cases} \quad (3.9)$$

where,

$LC_{i,j\,|\,r}$ = the lightness contrast of *p(i, j)* assuming a potential target lens of radius *r*,
$L_{i',j'}$ = the lightness value of *p(i, j)*,
$LC_{i,j\,|\,r\,|\,k}$ = the lightness contrast of *p(i, j)* in quadrant *k*,
$LD_{i,j\,|\,r\,|k}$ = the average lightness in quadrant *k* of A1,
$LB_{i,j\,|\,r\,|k}$ = the average lightness in quadrant *k* of A2,
$\sigma_{i',j'\,|\,r\,|\,k}$ = the flag indicating whether where *p(i',j')* resides.

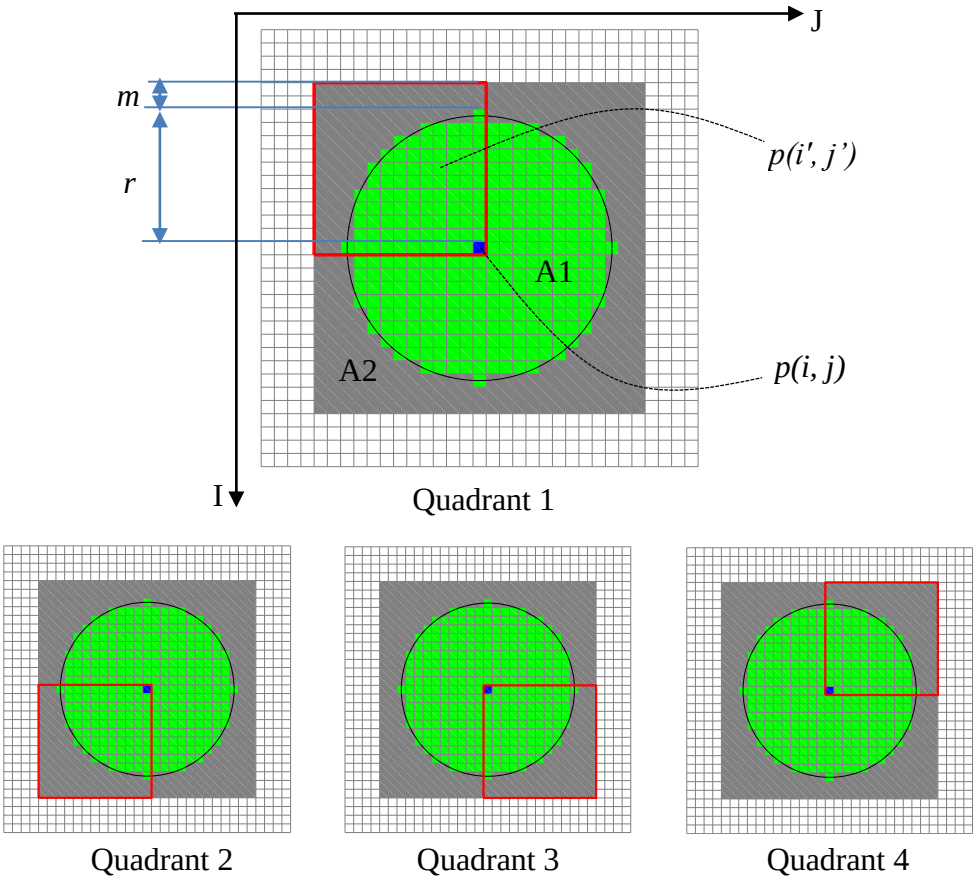


Figure 3-7 Illustration of quadrants.

Average Saturation

The last feature in Equation 3.1 is the average saturation, which is the maximum of the average saturations in A1 and in an annulus area, A3, as illustrated in Figure 3-8. The annulus area has an inner radius of *r* and an outer radius of *2r*. The reason of including the saturation in A3 is to adapt to nighttime scenarios when the color information is lost due to the pixels in A1 being overexposed to nearly white and the diffusion of light forms a halo with the desired color information around the lens. Using the maximum in A1 and A3 automates the adaptation to both scenarios. Mathematical formulation is given in Equations 3.10 – 3.12.

$$SD_{i,j\,|\,r} = \frac{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} + \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right) \times S_{i',j'}}{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} + \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right)} \qquad (3.10)$$

$$SA_{i,j\,|\,r} = \frac{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} - \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right) \times S_{i',j'}}{\sum\left(\sigma_{i',j'\,|\,r\,|\,k} - \left|\sigma_{i',j'\,|\,r\,|\,k}\right|\right)} \qquad (3.11)$$

$$\sigma_{i',j'\,|\,r\,|\,k} = \begin{cases} 1, & if\,|p(i,j), p(i',j')| \le r \\ 0, & if\,|p(i,j), p(i',j')| > 2r \\ -1, & if\ |p(i,j), p(i',j')| > r\ and\ |p(i,j), p(i',j')| \le 2r \end{cases} \qquad (3.12)$$

where,

$SD_{i,j\,|\,r}$ = the average color saturation in A1,

$SA_{i,j\,|\,r}$ = the average color saturation in A3,

$\sigma_{i',j'\,|\,r\,|\,k}$ = the flag indicating where *p(i',j')* resides.

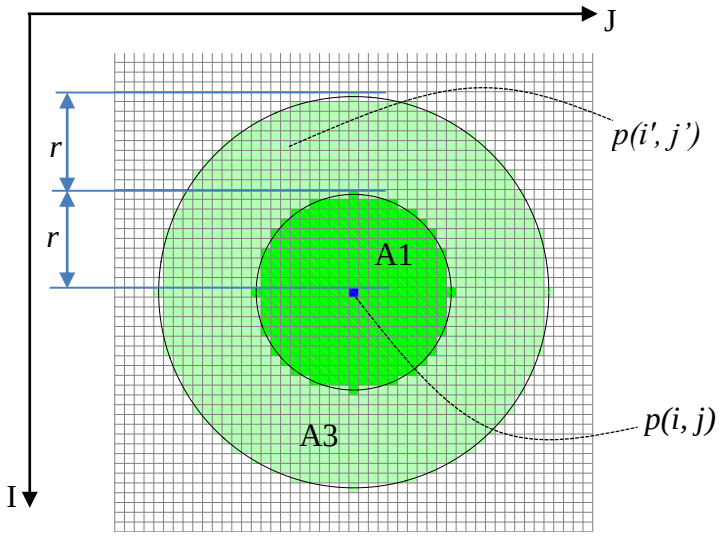


Figure 3-8 Illustration of the annulus area.

Similar to lightness, saturation can also be measured in a family of cylindrical color models, among which HSV and HSL are most commonly used. The S channel of HSV is used in this study to measure saturation, reasons being that it gives low saturations to white pixels. This is an important property in both daytime and nighttime detection. In nighttime, the lens area of both traffic lights and street lights can both be overexposed into white, making them undistinguishable by saturation. Therefore, the saturation of white pixels should be suppressed to give more attention

to the annulus area of the lights. In daytime, especially in sunny days, it is similarly important to suppress the saturation of the background sky as well as other overexposed objects due to reflection of the strong sun light. Figure 3-9 gives an illustration of the above situations. Another advantage of using HSV in nighttime is that it enforces delineation between the lens area and the halo area in terms of saturation, which is helpful to more accurate localization.

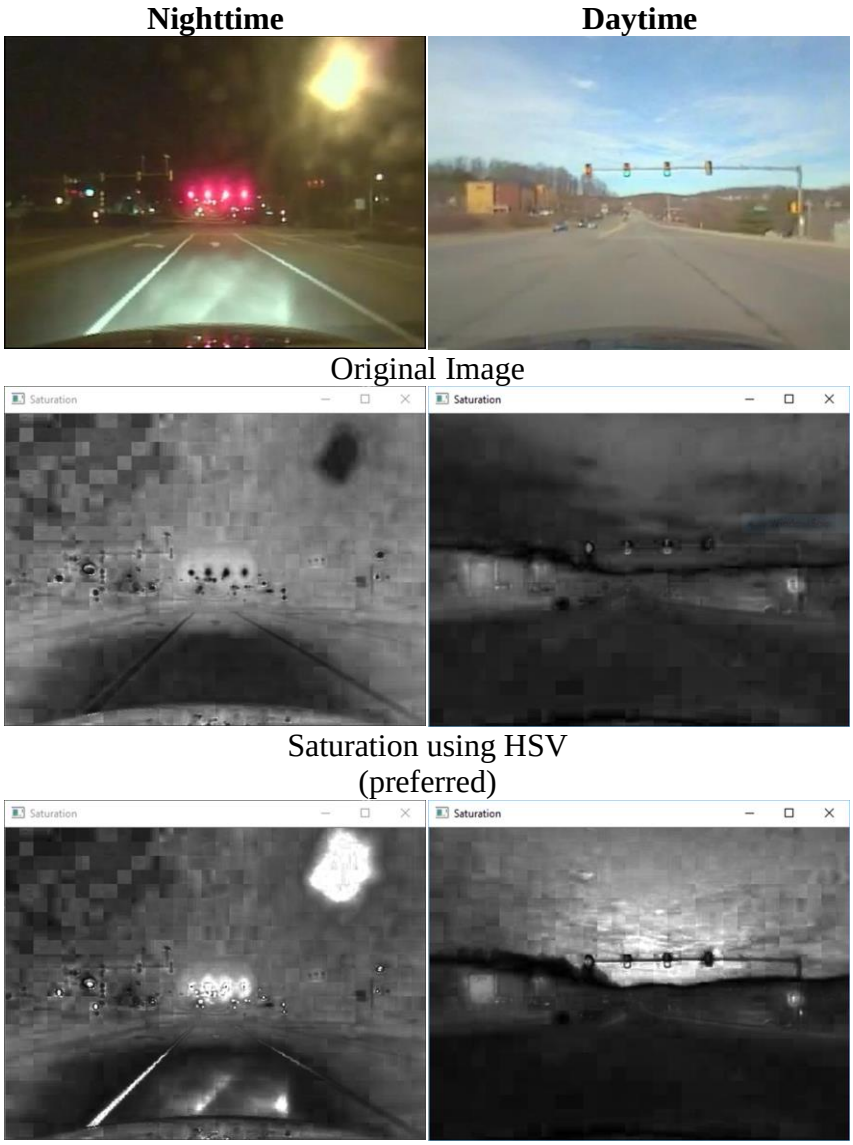


Figure 3-9 Saturation difference between HSV and HSL in nighttime.

3.2.2 Localization

A conspicuity map highlights the potential centers of traffic signal lenses but is not a direct binary mask of where the lenses are. In Figure 3-5, there are other conspicuous regions that do not belong to a traffic lights. Fortunately, locating candidate lenses can be done in several simple and effective steps, based on the fundamental assumption that the traffic lights exist in the scene and are among the most conspicuous objects. Of course, such assumption can be violated in many cases, such as when the image contains not traffic lights, when the traffic lights were viewed from a sharp angle, or simply when there are other brighter objects competing in the scene. These cases are arguably universal difficulties for all purely image based approaches. However, with the localization procedures proposed below, even some of the above difficulties can be automatically sensed and correctly responded to.

Overview

Figure 3-10 gives an overview of the localization procedures. In the flowchart, several abbreviations are used to make the diagram concise. The full descriptions are as following and details are explained in later paragraphs:

- CM – Conspicuity map (multi-radius aggregated version, see Equation 3.2)
- WCM – Working conspicuity map
- PPM – Peak pixel mask
- C_i – Connected component set at iteration i
- S_i – Traffic signal candidate set at iteration i
- S – Overall traffic signal candidate set
- N_{top} – The number of top conspicuity candidates to locate for the image
- N_{pi} – The maximum number of new candidates expected per iteration

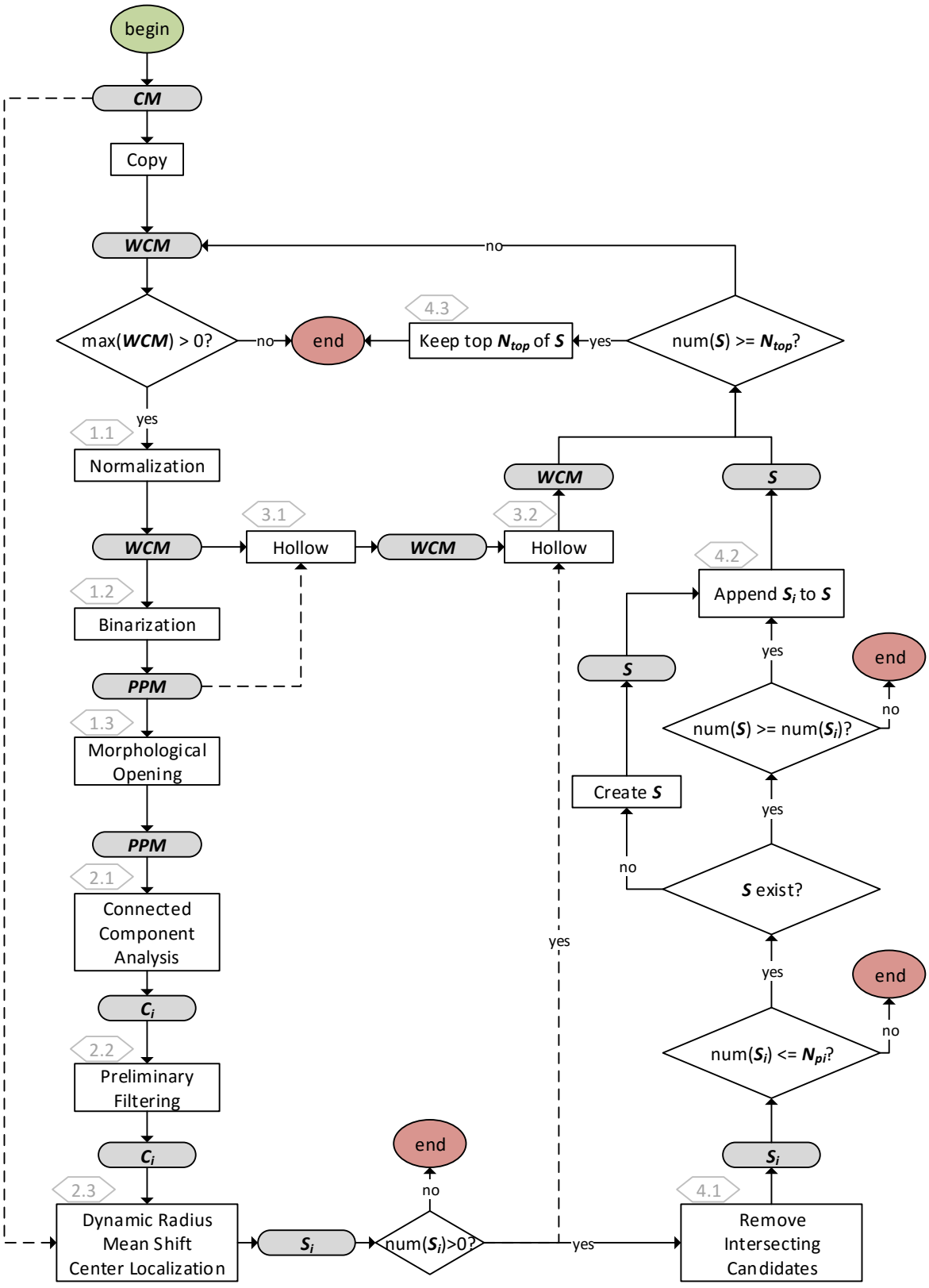


Figure 3-10 Flowchart of the localization procedures.

In general, the procedures go through multiple iterations to extract candidates at various levels of peaks. In each iteration, the working conspicuity map are normalized to have a maximum value of 1 (step 1.1). Then, the pixels whose normalized conspicuity values are above a pre-defined threshold, namely the peak pixels, are flagged to form a peak pixel mask (step 1.2). Since this peak pixel mask can be noisy, a morphological opening operation is performed to reduce noise (step 1.3). A connected component analysis is performed on the denoised peak pixel mask so peak pixels connected to each other are grouped into separate components (or blobs) (step 2.1). At this points, each component region is considered a good approximation of a candidate. However, recall that the conspicuity value represents the likelihood of a pixel being the center of a signal lens, the pixels in each components are likely to cover only a small center portion of an actual signal lens. Therefore, a local refined search using an extended mean shift algorithm is used to find the conspicuity mass center as the candidate center and the corresponding r for that center pixel is used as the radius of the candidate (step 2.3). The refined candidate regions are typically larger than the initial component region. Note, between step 2.1 and step 2.3, there is a preliminary filtering step 2.2, which filters out components that are unlikely to be related to a candidate. The filtering is based on the aspect ratio and the pixel density of each component's bounding box. After step 2.3, if new candidates are detected, they will be reduced to make sure no two candidates intersect or touch each other. The larger candidate is kept if any intersection occurs. The new candidates are then added to a cumulative set of candidates if several checks are passed (step 4.2). The checks are to make sure in every iteration, the amount of new candidates is reasonable (will be further explained shortly).

Besides detecting new candidates, the working conspicuity map should be updated accordingly so it can be used in the next iteration. The idea of updating is to turn pixels covered

by the peak pixel mask (before denoising) and by the new candidate regions to zero so these regions will not be repeatedly considered in future iterations (steps 3.1 and 3.2). Following this routine, the theoretical behavior of the iteration is to find out local maximum regions in a descending order of conspicuity level. An analog to this process is to imagine the conspicuity map as a terrain where the value corresponds to the height. In an ideal scenario, we expect a few tall mountains representing the actual traffic lights and the rest majority being small hills or plain. Each iteration removes the existing tallest areas (and more precisely, drilling wells in those areas) as new candidates. Actual traffic lenses are expected to be captured by the first one or two iterations. The further down the iteration, the noisier candidates are likely to be detected at a larger amount.

As briefly mentioned earlier, there are checks between steps 4.1 and 4.2. The first check after step 4.1 is to assess how relevant the new candidate set is according to its size. When the size exceeds a certain threshold, it physically means none of these candidates stand out among their peers and hence are irrelevant detections. Such condition will not only happen at a late iteration when only low conspicuity regions remain, but also will happen in the first iteration with a challenging scene containing many objects that are equivalently conspicuous as actual traffic lights. In the first case, the algorithm stops correctly to avoid further iterations. In the second case, the algorithm halts reasonably because it automatically senses the violation of the fundamental assumption that traffic lights are the most conspicuous. The check immediately before step 4.2 is similar but in a relative manner. The idea is, if the number of candidates detected in a later iteration, and hence with lower conspicuity, is larger than the total number of existing high conspicuity candidates, the new candidate set is unlikely to be relevant. These checks can help the algorithm to exit effectively with only a few iterations.

Some implementation details are given below for certain steps.

Peak Pixel Mask

Equations 3.13 – 3.15 are used in normalization (step 1.1), binarization (step 1.2), and denoising (step 1.3) to generate the peak pixel mask, respectively. In Equation 3.14, the threshold provides a control of the strictness of peak selection during each iteration. The closer the threshold is to 1, the fewer peaks are selected and the iteration moves down the conspicuity level more slowly. In Equation 3.15, the morphological opening consists of an erosion and a dilation of the peak pixel mask. Erosion assigns the minimum value of a pixel's neighborhood to that pixel and dilation assigns the maximum value instead. The neighborhood is defined in a structure element, i.e., a template matrix whose center element represents the pixel being calculated and the other elements flagging neighborhood pixels by the value of 1. After erosion, regions in the peak pixel mask will shrink and some small regions may disappear depending on how large the neighborhood is. After dilation, remaining regions will be inflated back to the original size. As a result, small regions (typically noises) are removed while large regions are reserved at the same size.

$$WCM_{i,j} \leftarrow \frac{WCM_{i,j}}{\max\left(WCM_{i,j}\right)} \mid \max\left(WCM_{i,j}\right) > 0 \quad (3.13)$$

$$PPM_{i,j} = \begin{cases} 1, & if\ WCM_{i,j} > h_{peak} \\ 0, & otherwise \end{cases} \quad (3.14)$$

$$PPM_{i,j} \leftarrow f_{opening}\left(PPM_{i,j}, se\right) \quad (3.15)$$

where,

$WCM_{i,j}$ = the normalized conspicuity value of pixel *p(i, j)* in the working conspicuity map,

$PPM_{i,j}$ = the pixel *p(i, j)* in the peak pixel mask,

h_{peak} = a threshold of normalized conspicuity value in the range of [0,1]. Typically chosen as 0.9,

$f_{opening}(\cdot)$ = the morphological opening function,

se = A structure element for morphological opening, typically chosen as a disk neighborhood of radius 1.

Candidate Extraction

In step 2.1, 8-connectivity is used for the connected component analysis. One pixel is considered connected to another if it is in any of the eight immediate neighborhood positions of the other pixel. The result of step 2.1 is a list of pixel groups, called components. Each component gives an estimated region of where the center of a candidate lens should be.

In order to more accurately locate the final center of each candidate and determine its size (i.e., lens radius), a dynamic radius mean shift (DRMShift) algorithm is proposed (step 2.3). The basic mean shift algorithm is an iterative approach to find the center position of a region with a fixed size over a set of weighted points. The resulting position minimizes the distance between the centroid and the mass center of the points enclosed in that region. When applied on the conspicuity map, the points are the pixels and the weights are the conspicuity values. However, because each pixel in the conspicuity map (Equation 3.2) is associated with an optimal radius (r) that yields the optimal conspicuity value, moving the center of the search region also implies changing the optimal radius accordingly. Therefore, the proposed DRMShift algorithm extends the basic mean shift by allowing the size of the search region to change during the iterative search. A segment of pseudo code is given in Algorithm 3-1 for DRMShift. Range checking for pixel indices is not reflected in this code so it conveys the main idea in a concise manner.

Algorithm 3-1

DRMShift

Inputs:

* Initial center pixel position [$\mathbf{i_0}$,$\mathbf{j_0}$]
* Conspicuity map **CM**
* Radius map **RM**
* Converge threshold $\mathbf{h_c}$
* Maximum number of iteration $\mathbf{n_{max}}$

Steps:

```
1  i = i0;
2  j = j0;
3  d = hc;
4  while nmax > 0 and d >= hc
5    nmax ← nmax − 1;
6    m00 = 0;
7    m01 = 0;
8    m10 = 0;
9    r = RM[i,j];
10   for i' = i − r to i + r
11       for j' = j − r to j + r
12           if |p(i,j), p(i',j')| <= r
13               m00 ← m00 + CM[i',j'];
14               m01 ← m01 + i' * CM[i',j'];
15               m10 ← m10 + j' * CM[i',j'];
16           end if
17       end for
18   end for
19   if m00 > 0
20       itmp = i;
21       jtmp = j;
22       i = m01/m00;
23       j = m10/m00;
24       d = |p(i,j), p(itmp,jtmp)|;
25   end if
26 end while
```

Outputs:

* Final center pixel position [**i**,**j**]
* Final region radius, **r**

Between step 2.1 and step 2.3 in Figure 3-10, there is an optional preliminary filtering step (2.2) which checks the geometric properties of each component before they can be considered for refined candidate extraction. Two properties, the aspect ratio (*asp*) and the pixel density (*pd*) of the bounding box of the component are checked. The aspect ratio is defined as the ratio of the short edge to the long edge and the pixel density is defined as the number of peak pixels as a ratio of the total number of pixels (including non-peak pixels) enclosed in the bounding box. The minimum thresholds for both the aspect ratio and the pixel density are by default 0.6 to be forgiving to imperfect image quality.

3.3 State Classification

A 2D histogram matching approach is employed for classifying the state (color) of each detected candidate. This approach is essentially a simple learning approach, which at the first thought is against the generic design principle that requires little dependency on training data collected with the same camera used in testing. However, since candidates are detected and assumed to be true traffic signal lenses, the primary functionality of the classifier is to make a choice among three possible traffic signal colors. Therefore, the training data do not need to be highly representative for the test data. As long as the training data capture a good delineation between the three traffic signal colors. In other words, the training data can be images collected using cameras other than the one used for collecting the test data.

More specifically, a 2D histogram of the "a" channel and the "b" channel from the CIELab space is calculated as the matching feature. Equations 3.16 – 3.18 formulate the calculation of the histogram. In fact, there are two histograms that should be calculated for each candidate, one for the pixels in the disk (lens) area (A1) and another for the pixels in the annulus area (A3), so color information in the nighttime when "halo effect" occurred could be captured. With a training sample,

only one of this histograms is calculated because the area containing the color is known already during the manual annotation. For a particular signal color, a trained histogram is calculated as the average of histograms of all the training samples. Example trained histograms are illustrated in Figure 3-11.

$$H_{k,l \mid mode} = \frac{\sum \sigma_{i',j' \mid mode} \times \beta_{i',j' \mid k,l}}{\sigma_{i',j' \mid mode}} \quad (3.16)$$

$$\sigma_{i',j' \mid mode} = \begin{cases} 1, & if\ mode = A1\ and\ |p(i,j), p(i',j')| \le r \\ 1, & if\ mode = A3\ and\ r < |p(i,j), p(i',j')| \le 2r \\ 0, & otherwise \end{cases} \quad (3.17)$$

$$\beta_{i',j' \mid k,l} = \begin{cases} 1, & if\ a_{i',j'} \in Bin_k^a\ and\ b_{i',j'} \in Bin_l^b \\ 0, & otherwise \end{cases} \quad (3.18)$$

where,

$H_{k,l \mid mode}$ = the value of the histogram at bin (*k*, *l*) under the specified mode,

$mode$ = A1 for histogram in the disk area and A2 for the histogram in the annulus area,

$\sigma_{i',j' \mid mode}$ = the flag indicating whether *p(i', j')* falls in to the specified area of *p(i, j)*,

$\beta_{i',j' \mid k,l}$ = the flag indicating whether the "a" value and the "b" value of *p(i', j')* fall in to the k^{th} bin of "a" (Bin_k^a) and l^{th} bin of "b" (Bin_l^b).

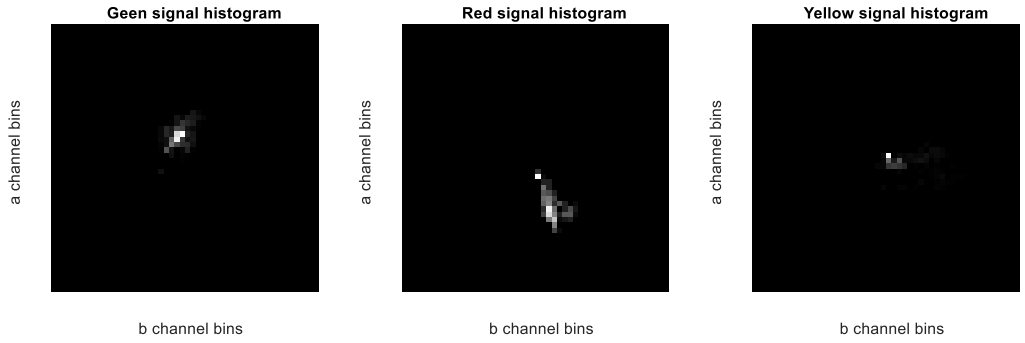


Figure 3-11 Example trained a-b histograms.

When testing, the two histograms of each candidate's lens and annulus areas are calculated and compared to each signal color's trained histogram to derive a similarity score. The similarity score is calculated using Equations 3.19 – 3.20. The signal color with the highest similarity score is chosen as the state of the candidate.

$$sc_{cand\,|\,color} = \max\left(sc\left(H_{cand\,|\,mode}, H_{color}\right)\right) \;|\; mode \in \{A1, A3\} \quad (3.19)$$

$$sc(H, H') = 1 - \frac{2 \times \operatorname{acos}\left(\dfrac{\sum H_{k,l} \times H'_{k,l}}{\sqrt{\sum H_{k,l} \times H_{k,l}} \times \sqrt{\sum H'_{k,l} \times H'_{k,l}}}\right)}{\pi} \quad (3.20)$$

where,

$sc_{cand\,|\,color}$ = the similarity score of a candidate as a particular signal color,

$H_{cand\,|\,mode}$ = the histogram of the candidate. When the average saturation in A1 is higher than in A3, *mode* = A1; otherwise *mode* = A3,

H_{color} = the trained histogram of a signal color.

3.4 Spatiotemporal Framework

In this section, key stages of the spatiotemporal framework are given with details. The implementation of map projection is first explained as a fundamental stage of going from spherical coordinates to plane coordinates so distance can be calculated conveniently. Although GPS positions of the frames and signalized intersections are required, the method only asks for two relaxed quality criteria: 1) the position data should be available at an adequate frequency and within the borders of the traveled highway and 2) the traffic signal locations should be mapped no farther than 5 meters outside of the intersection area between two highways. With projected coordinates, vicinity calculation methods are explained regarding finding the closest signalized intersection for each frame. An extended kd-tree search algorithm is proposed to optimize the vicinity calculation speed. A third section describes the classification of movements at each signalized intersection according to the trajectory data. The last two sections give comprehensive descriptions of the temporal coordination of TSR as each signalized intersection.

3.4.1 Map Projection

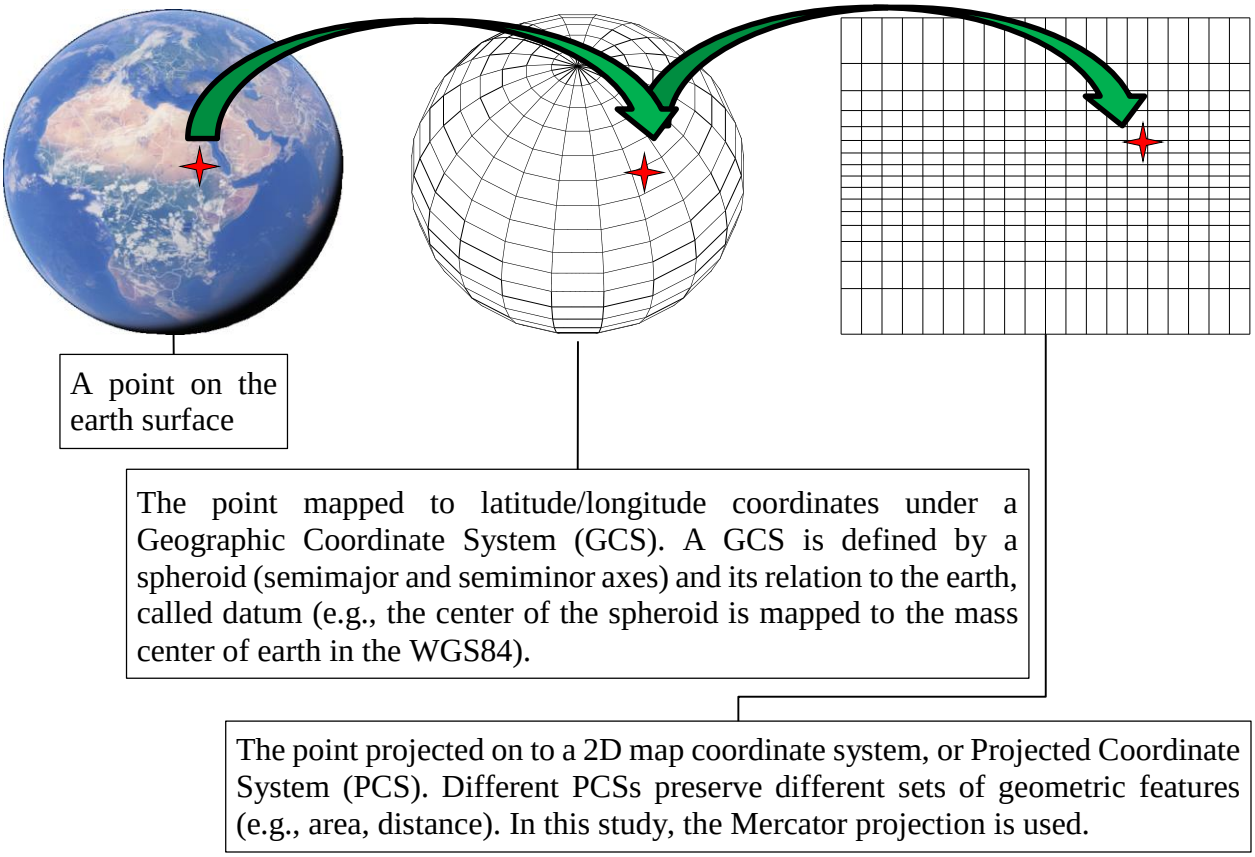


Figure 3-12 Illustration of position data conversion.

Raw spatial data (of video frames and signalized intersections) are assumed to be presented in latitudes and longitudes under the World Geodetic System 1984 (WGC84), unless otherwise specified. WGC84 is an ellipsoid geographic coordinate system (GCS) that does not directly support planar geometric calculation on local regions of the earth surface. In order to conduct planar spatial analyses, such as distance calculation and moving direction judgement, latitudes and longitudes need to be projected onto a 2D projected coordinate system (PCS). Figure 3-12 illustrates the pipeline of a point on the earth surface being transferred to a GCS and then to a PCS. In this thesis, the Mercator projection, a cylindrical map projection, is chosen.

Actual implementation of the Mercator projection follows Equations 3.21 - 3.23. The basic idea of the projection is to map a point on the GCS spheroid to a point on the cylinder wrapped around the spheroid.

$$[x, \quad y] = R_{earth} \times \left[\lambda, \quad \ln\left(\tan\left(\frac{\pi}{4} + \frac{|\chi|}{2}\right)\right)\right] \qquad (3.21)$$

$$\chi = 2\,\mathrm{atan}\left(\tan\left(\frac{\varphi}{2} + \frac{\pi}{4}\right) \times \left(\frac{1 - ecc * \sin(\varphi)}{1 + ecc * \sin(\varphi)}\right)^{\frac{ecc}{2}}\right) - \frac{\pi}{2} \qquad (3.22)$$

$$[\lambda, \quad \varphi] = [lon, \quad lat] \times \frac{\pi}{180} \qquad (3.23)$$

where,

x = the projected x coordinate, in meters,
y = the projected y coordinate, in meters,
R_{earth} = the average radius of the earth (default = 6378137), in meters,
λ = longitude, in radians,
φ = latitude, in radians,
lon = longitude, in degrees,
lat = latitude, in degrees,
χ = conformal latitude, in radians,
ecc = eccentricity of the GCS spheroid (default = 0.081819190842621486 for WGS84), no unit.

The Mercator projection is known to introduce increasing geometric distortion as the projected point goes further from the default projection origin at [latitude = 0, longitude = 0]. In order to overcome this issue, a new origin of projection should be chosen so it is relatively centered in the region of analysis. In this study, the new origin is chosen as the centroid of the positions of all frames. Accordingly, the longitude and the conformal latitude (in radians) are transformed using Equations 3.24 - 3.28 before they can be plugged in Equation 3.21.

$$\begin{bmatrix} x_{cart} \\ y_{Cart} \\ z_{cart} \end{bmatrix} = \begin{bmatrix} \cos(\chi) \times \cos(\lambda) \\ \cos(\chi) \times \sin(\lambda) \\ \sin(\chi) \end{bmatrix} \quad (3.24)$$

$$\begin{bmatrix} x_{cart\_R} \\ y_{Cart\_R} \\ z_{cart\_R} \end{bmatrix} = \begin{bmatrix} \cos(\chi_0) \times \cos(\lambda_0) & \cos(\chi_0) \times \sin(\lambda_0) & \sin(\chi_0) \\ -\sin(\lambda_0) & \cos(\lambda_0) & 0 \\ -\sin(\chi_0) \times \cos(\lambda_0) & -\sin(\chi_0) \times \sin(\lambda_0) & \cos(\chi_0) \end{bmatrix} \times \begin{bmatrix} x_{cart} \\ y_{Cart} \\ z_{cart} \end{bmatrix} \quad (3.25)$$

$$h = \sqrt{x_{cart\_R}^2 + y_{cart\_R}^2} \quad (3.26)$$

$$\lambda_R = \begin{cases} \text{atan}\left(\frac{y_{cart_R}}{x_{cart_R}}\right), & if\ x_{cart\_R} \geq 0 \\ \text{atan}\left(\frac{y_{cart_R}}{x_{cart_R}}\right) + \pi, & if\ x_{cart\_R} < 0\ and\ y_{cart\_R} \geq 0 \\ \text{atan}\left(\frac{y_{cart_R}}{x_{cart_R}}\right) - \pi, & if\ x_{cart_R} < 0\ and\ y_{cart_R} < 0 \end{cases} \quad (3.27)$$

$$\chi_R = \text{atan}\left(\frac{z_{cart_R}}{h}\right) \quad (3.28)$$

where,

x_{cart} = Cartesian x coordinate, no unit,
y_{cart} = Cartesian y coordinate, no unit,
z_{cart} = Cartesian z coordinate, no unit,
λ = longitude, in radians,
χ = conformal latitude, in radians,
x_{cart_R} = rotated Cartesian x coordinate, no unit,
y_{cart_R} = rotated Cartesian y coordinate, no unit,
z_{cart_R} = rotated Cartesian z coordinate, no unit,
λ_0 = longitude of the new projection origin, in radians,
χ_0 = conformal latitude of the new projection origin, in radians,
λ_R = rotated longitude, in radians,
χ_R = rotated conformal latitude, in radians,

3.4.2 Vicinity Calculation

A frame is likely to contain a traffic signal only when the vehicle was close to a signalized intersection. Therefore, by calculating the distance from the vehicle position of each frame (or "frame position" for short) to its nearest signalized intersection, a traffic signal vicinity profile can be generated (e.g., Figure 3-13). By setting a vicinity threshold (e.g., 50 m as indicated by the horizontal red line in Figure 3-13), a series of profile valleys can be isolated, representing candidate

batches of frames that might have passed a signalized intersection. The valley bottom indicates the nearest frame to the intersection (but not necessarily the traffic signal) and is called the *anchor frame* (indicated by a red box) in the following discussion. When the valley bottom is flat, the latest bottom frame is chosen as the anchor frame. Frames before this anchor frame are called *upstream frames* while those after the anchor frame are called *downstream frames*. Intuitively, traffic signals are expected to be captured in most upstream frames (up to a certain distance) and a few of immediate downstream frames, if not none. Detailed algorithms regarding the temporal coordination of TSR starting from each anchor frame will be explained in Sections 3.4.4 and 3.4.5. The position data of the signalized intersections are requested from the OpenStreetMap™ server within a buffered bounding box around all frame positions (see Section 4.4 for details).

Note, constructing the vicinity profile is a classical nearest neighbor problem in a low dimensional space (e.g., 2D). A slightly more formal definition of the problem is, given two finite sets of points $T \subset R^2$ (target points) and $Q \subset R^2$ (query points), find for each query point $q_i \in Q$ the nearest target point $t_j \in T$ so that $dist(q_i, t_j) \leq dist(q_i, t_k)\ \forall\ t_k \in T$. Solving this problem with a naïve all-pair distances algorithm has a time complexity of $O(N_Q \times N_T)$, where N_Q and N_T are the numbers of points in Q and T, respectively. In our case, the query points are frame positions and the target points are signalized intersections, so $N_Q \gg N_T$.

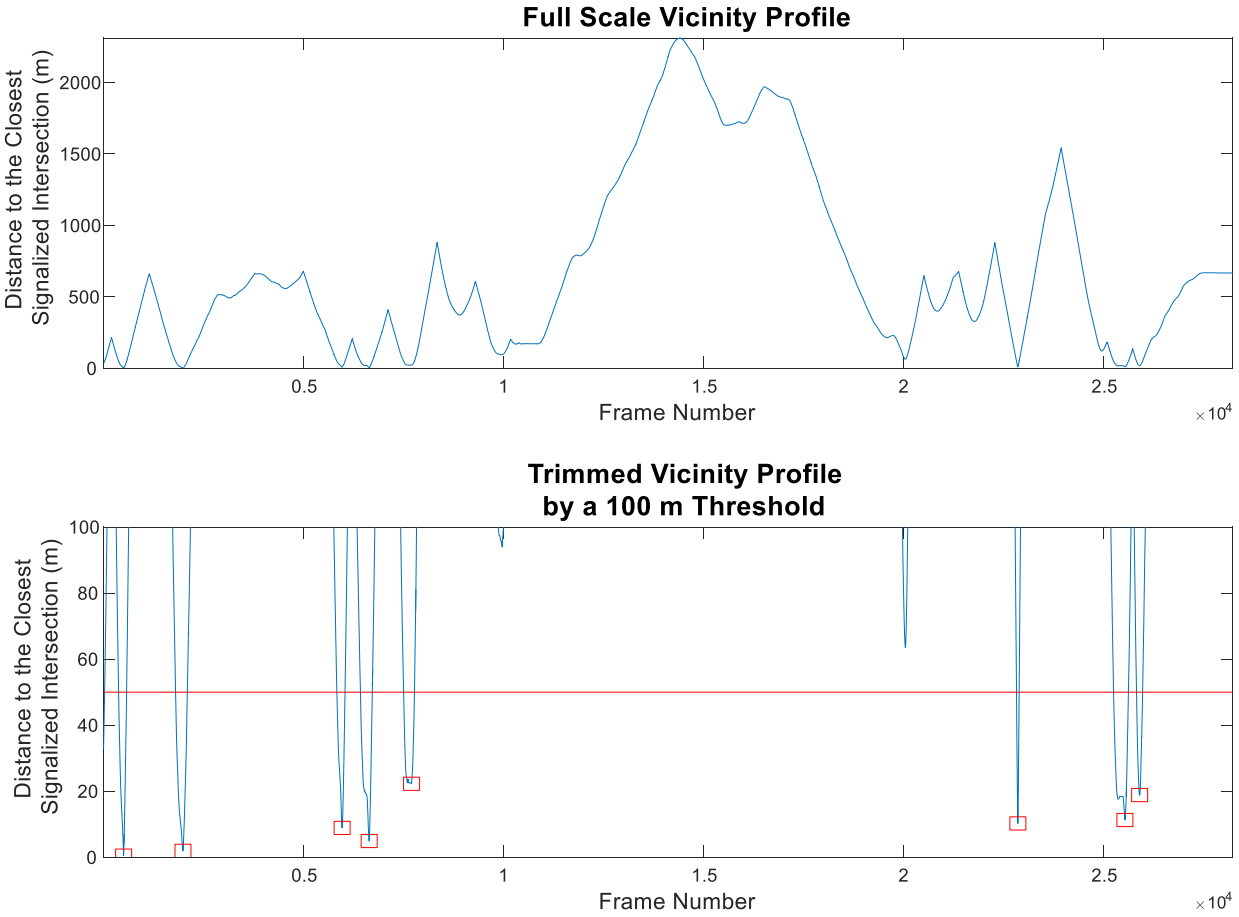


Figure 3-13 Illustration of a traffic signal vicinity profile.

Basic KD-Tree Method

An efficient and commonly used algorithm is based on a kd-tree indexing of the target points *(63)*. A kd-tree is a multidimensional binary tree whose tree nodes are the points being indexed. In the following discussion, nodes and points are used interchangeably. Each non-leaf node has a left child node and a right child node, serving as the roots of the left and the right subtrees, respectively. The key property of a kd-tree is that for each non-leaf node, the nodes in the left subtree is to one side of the node and those in the right subtree is to the other side of the node in a chosen dimension. The dividing dimension is typically selected to balance the numbers of nodes in the two subtrees. The amortized time complexity of constructing a kd-tree is

$O(N_T \times \log(N_T))$ and that of querying the nearest target of a given point is $O(\log(N_T))$. Therefore, the overall time complexity of constructing the vicinity profile can be reduced to $O\left(\left(N_T + N_Q\right) \times \log(N_T)\right)$. This approach has been effectively employed by Fairfield and Urmson in their real-time application *(4)*.

Extended KD-Tree Method

In fact, for offline analysis, additional speed improvement can be made by changing the order of query and assuming some spatiotemporal characteristics of the frame positions and the signalized intersections. *The basic idea is, if two frames have the same nearest signalized intersection and the time between these two frames was shorter than a threshold value τ_{min}, all frames in between would have the same nearest signalized intersection* (Condition 3-1).

It is convenient to borrow the concept of Voronoi diagram to explain the physical meaning of Condition 3-1 and determine how τ_{min} should be chosen. A Voronoi diagram is a division of the space into connecting regions according to a given set of points in that space, called sites. Each site is associated with one resulting region so that the site is the nearest site to any point in that region. Imagine the signalized intersections as the sites. A frame is nearest to a signalized intersection if and only if it falls in the Voronoi region of that signalized intersection. The only case when a frame can be nearest to more than one signalized intersection is when it falls on the boundary between two or more Voronoi regions. Therefore, if two frames have the same nearest signalized intersection, they must be in the same Voronoi region (boundary inclusive). The only possible case for any frame in between having a different nearest signalized intersection is when the vehicle exited the current Voronoi region and reentered during the time between the two frames. When the time is adequately short, this case is mostly impossible. In order to define this time

threshold, the minimum time a vehicle would have stayed in the Voronoi region of any signalized intersection is set as the τ_{min}. A conservative and practical value of τ_{min} can be chosen as 1.5 seconds, corresponding to about 20 frames with a frame rate of 14 FPS.

Therefore, rather than searching the nearest signalized intersection for each frame position using the kd-tree, we can recursively search for the nearest signalized intersection of the two end frames of a batch of frames. If Condition 3-1 is met, all frame in between will be populated with the same nearest signalized intersection without kd-tree search; otherwise, the nearest signalized intersection is searched using the kd-tree for the frame in the middle and the batch is divided into two sub batches for recursion. The time complexity of this algorithm is $O\left(\left(N_T + \frac{N_Q}{F(\tau_{min})}\right) \times \log(N_T)\right)$, where $F(\tau_{min})$ denotes the number of frames corresponding to the chosen τ_{min}. It should be noted, since N_Q is large and presents the efficiency bottleneck of the calculation, scaling it down to $\frac{N_Q}{F(\tau_{min})}$ is a significant speed improvement in practice.

3.4.3 Movement Classification

An additional piece of useful information that can be derived for each clip of candidate frames is the movement of the vehicle, i.e., left turn, thru, or right turn. Specifically, the angle between two vectors is used to classify the movement. The first vector (v_1) goes from the first upstream frame (F_{start}) to a turning point frame (F_{tp}) and the second vector (v_2) goes from F_{tp} to the last downstream frame (F_{end}). F_{tp} is defined as the frame with the maximum perpendicular distance to the baseline vector (v_b) between F_{start} and F_{end}. If the angle, θ, between v_1 and v_2 is no larger than a threshold θ_{min}, the movement is classified as a thru movement. If θ is larger than θ_{min} and measured counterclockwise from v_1 to v_2, the movement is classified as a left turn. Otherwise, the

movement is classified as a right turn. Mathematically, Equation 3.29 is used for movement classification. Examples are given in Figure 3-14. The θ_{min} used in this study is 20 degrees, which yielded 100% accuracy for all testing data.

$$\begin{cases} \theta = \cos^{-1}\dfrac{v_1 \cdot v_2}{\|v_1\| * \|v_2\|} * \dfrac{180}{\pi} \\ V_{cross} = V_1 \times V_2 \\ movement = \begin{cases} right\ turn, & if\ \theta > \theta_{min}\ and\ V_{cross}(3) > 0 \\ through, & if\ \theta \le \theta_{min} \\ left\ turn, & if\ \theta > \theta_{min}\ and\ V_{cross}(3) < 0 \end{cases} \end{cases} \qquad (3.29)$$

where,

v_1 = the 2D vector from F_{start} to F_{tp},
v_2 = the 2D vector from F_{tp} to F_{end},
θ = the angle between v_1 and v_2, in degrees,
$v_1 \cdot v_2$ = the dot product between v_1 and v_2,
$\|v_{1/2}\|$ = the length of v_1 (or v_2),
$V_{1/2}$ = the homogeneous 3D coordinates of v_1 (or v_2), i.e., $V_{1/2} = [v_{1/2}(1), v_{1/2}(2), 1]$,
V_{cross} = the cross product between V_1 and V_2, also a 3D vector,
θ_{min} = the minimum angle to be recognized as a turning movement, in degrees.

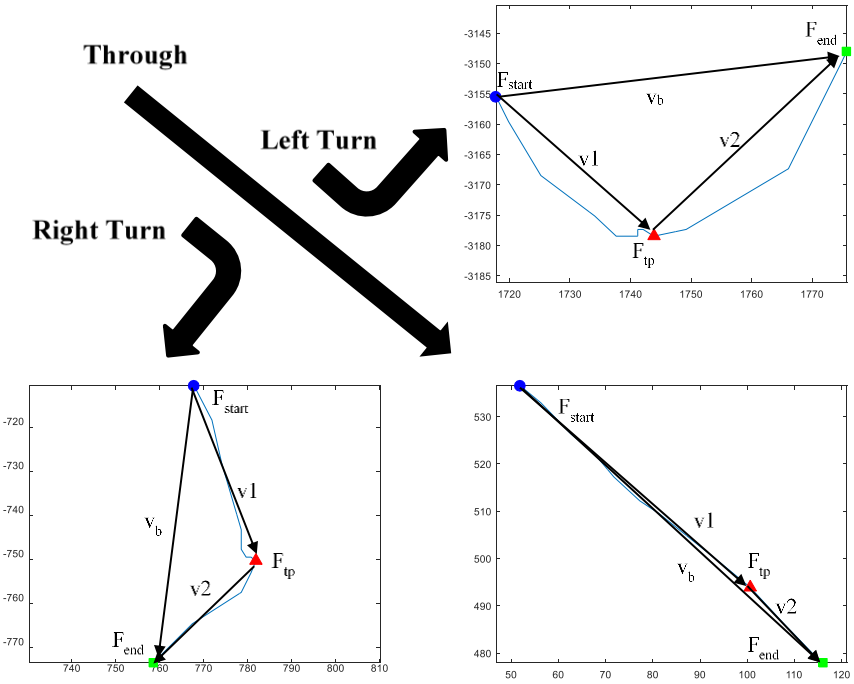


Figure 3-14 Determination of intersection movements.

3.4.4 Short Range Initialization

In addition to locating candidate frames for TSR, another motivation of the spatiotemporal framework is to utilize the temporal relationship between frames to improve the recognition results. A good strategy that is unique to offline analysis is to start detection in near frames and use stable detection results to assist tracked detection for distant frames. As will be shown in Section 5.2, detection performance can be affected by the target size in the image. The closer to the traffic signals, the larger the targets are and more accurately the detector works. Therefore, for each intersection traversal, an ideal starting point for TSR is the closest frame where traffic signals are still visible. However, as explained previously, the relaxed accuracy requirement of the traffic signal map and the GPS positions of frames would not support a precise calculation of the distance between a frame and the target traffic signals. The vicinity profile only gives a rough estimation of the closeness to a signalized intersection and the anchor frames are not necessarily the ideal starting points. As a result, rather than seeking for a perfect starting frame, a short range of frames are proposed to be used as an initialization set. For each of the frames in this short range, independent detection and classification are perform for the whole frame (or within a general region of interests). Candidates are associated using a dense optical flow based tracking algorithm.

Determine Short Range

The short range for initialization is defined based on the anchor frame. A number of upstream frames and downstream frames from the anchor frame are extracted as the short range based on their cumulative trajectory distances from the anchor frame. The cumulative distance between two frames is defined as the sum of straight line distances between all pairs of successive frames within these two frames. The straight line distance is the Euclidean distance in the projected [x, y] coordinate system (see Section 3.4.1). By default, frames within a trajectory distance of 10

m upstream or 5 m downstream from the anchor frame are extracted. These trajectory distance ranges can be adjusted to make a tradeoff between coverage and computational cost. By increasing the ranges, more frames will be considered for initialization and the chance of obtaining stable tracks of candidates is theoretically increased, but more computation efforts are needed as for each of these frames a whole frame detection at a wide range of radius scales will be performed.

Initial detection in this short range is a two-pass process. Both passes and the later long range tracked recognition stage are based on a dense optical flow (DOF) algorithm that estimates the movements of pixels from one image to another, enabling the projection of a candidate from one frame onto another frame as a position and size reference. A brief overview of the DOF based projection is given below before explaining the two-pass initialization process.

Dense Optical Flow Based Projection

A good range of DOF algorithms exist and the Farneback's method was chosen for its accommodation to camera vibrations *(64)*. The basic idea of the Farneback's method is to compare the similarity of pixel neighborhood in two grayscale images and the pixel neighborhood is modeled using polynomial expansion. Detailed explanation of this method is out of the focus of this research and readers of interests are advised to study the original paper *(64)*. In this section, some examples are shown to give a sense of how Farneback's method is used.

Figure 3-15 and Figure 3-16 illustrate the two-way projections of the same pair of successive frames using the Farneback DOF. In both projections, the resulting traffic signal positions and sizes closely match the actual positions and sizes in the target frame.

When signal colors change between frames, as shown in Figure 3-17 and Figure 3-18, the projected signal lenses are still reasonably preserved in the correct relative position in the projected

signal faces. However, the projection is more accurate in daytime (Figure 3-17) than in nighttime (Figure 3-18), because the optical flow is calculated using grayscale images and the image texture is richer in daytime to provide a better pixel neighborhood constraint. As can be seen in Figure 3-18, the green traffic lights are actually projected on the positions of the actual red lights, since in grayscale images, lights at both positions look almost identical. Nevertheless, the projected positions are still within a reasonable range of the ground truth positions for tracking purposes.

When the target leaves the view from one frame to another (Figure 3-19), the DOF also indicates such fact by giving negative (red) y flow to the disappearing target as it moves out of the upper bound of the image.

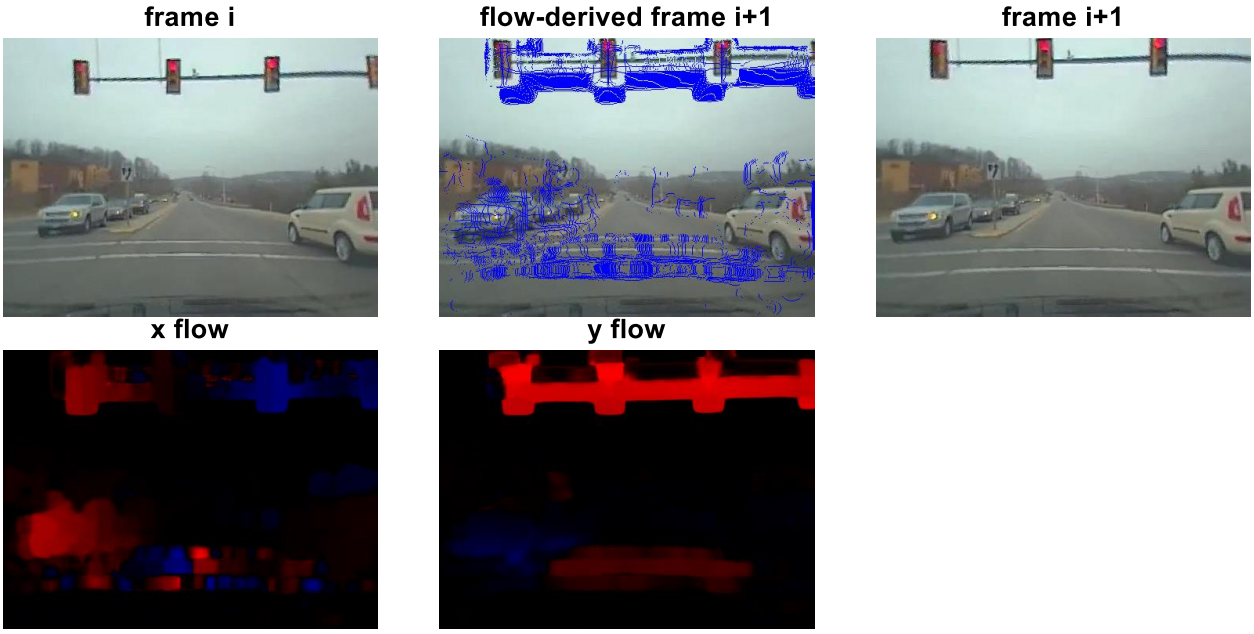


Figure 3-15 DOF based projection in normal forward motion (Frames 00053-338 to 339).

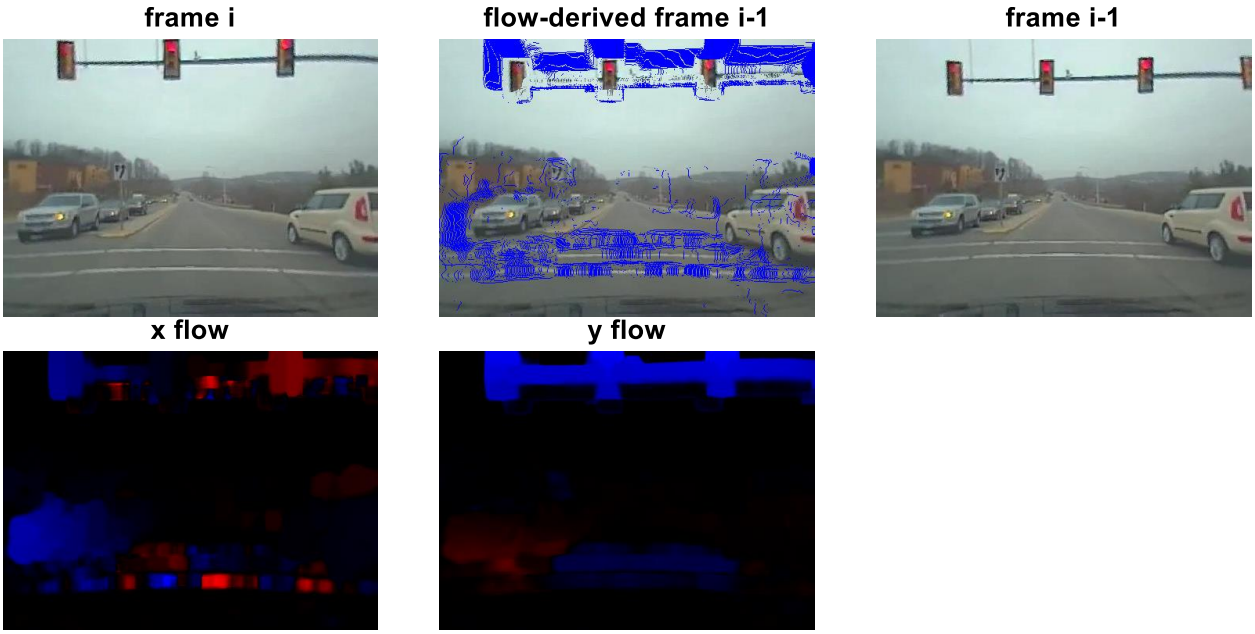


Figure 3-16 DOF based projection in normal backward motion (Frames: 00053-339 to 338).

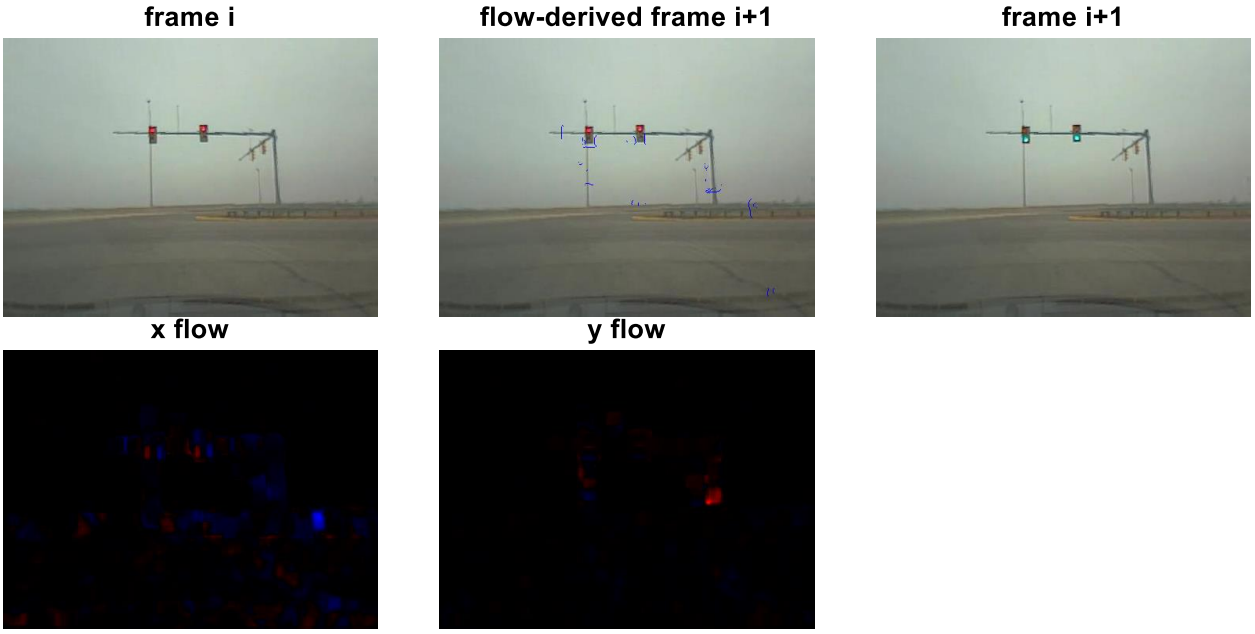


Figure 3-17 DOF based projection with change of signal states in daytime (Frames: 00053-21070 to 21069)

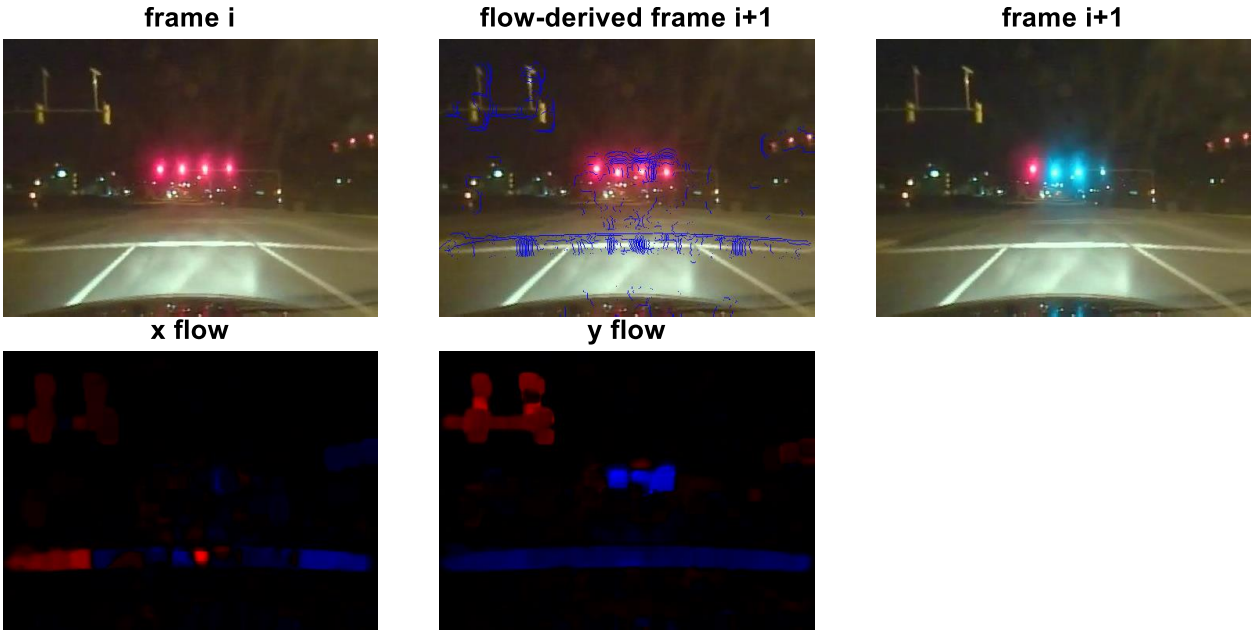


Figure 3-18 DOF based projection with change of signal states in nighttime (Frames: 00041-449 to 448)

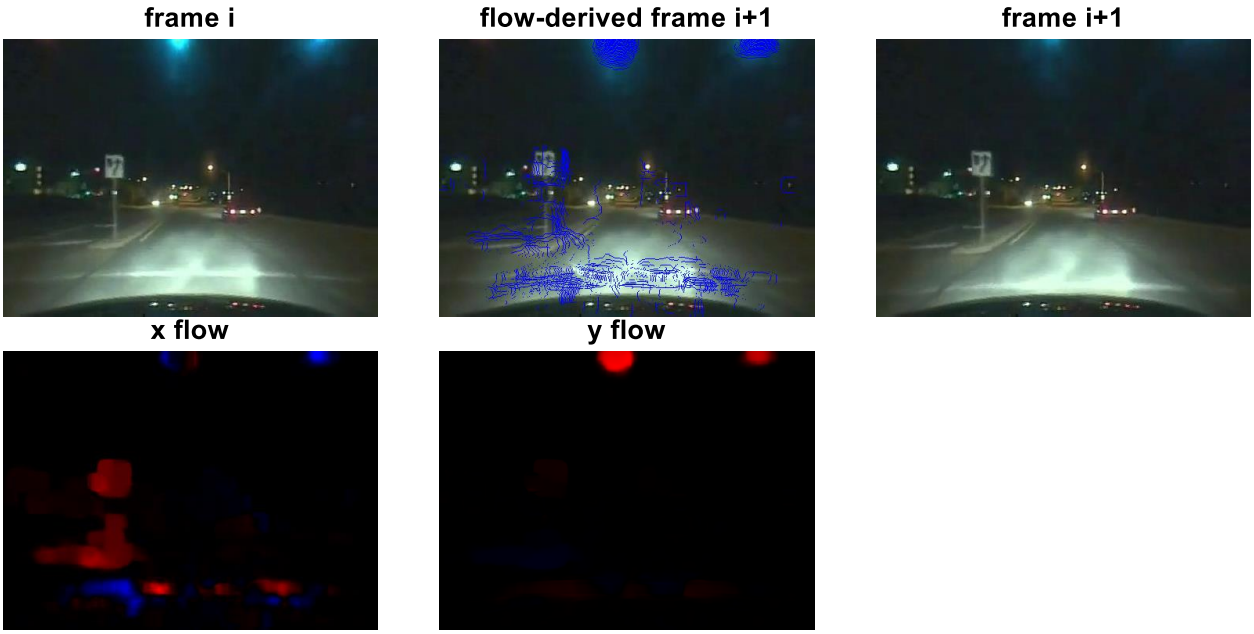


Figure 3-19 DOF based projection with target leaving the view (Frames: 00041-540 to 541)

In general, the Farneback DOF provides a robust motion estimate for the purpose of tracking traffic signal candidates, even in the case of signal color change and target disappearance. For candidate level projection (in contrast to pixel level), given a candidate in frame j and the optical flow from frame j to frame i, the projection of the candidate in frame i is calculated using Equations 3.30 – 3.34. It is possible for the projected candidate to be partially or entirely out of the image frame, especially during forward motion as illustrated Figure 3-19. These out of boundary projections can be checked against the dimension of the image and properly flagged.

$$cx_i = \frac{\min(x_i) + \max(x_i)}{2} \quad (3.30)$$

$$cy_i = \frac{\min(y_i) + \max(y_i)}{2} \quad (3.31)$$

$$r_i = \frac{\max[\max(x_i) - \min(x_i), \max(y_i) - \min(y_i)]}{2} \quad (3.32)$$

$$x_i = x_j + flow_{x|j\to i} \quad (3.33)$$

$$y_i = y_j + flow_{y|j\to i} \quad (3.34)$$

where,

cx_i = the x position of projected candidate center in frame i,
cy_i = the y position of projected candidate center in frame i,
r_i = the radius of project candidate in frame i,
x_i = the x coordinate of a point in the projected candidate in frame i,
y_i = the y coordinate of a point in the projected candidate in frame i,
x_j = the x coordinate of a point in the original candidate in frame j,
y_j = the y coordinate of a point in the original candidate in frame j,
$flow_{x|j\to i}$ = the optical flow in the x direction from frame j to frame i,
$flow_{y|j\to i}$ = the optical flow in the y direction from frame j to frame i,

First Pass – Detect and Associate

In the first pass of the short range initialization, the algorithm starts from the downstream frame of the short range and goes backward until the upstream frame. In this pass, not only will candidates be independently detected (and classified) in each individual frame, but also will candidates be associated in tracks across frames. A segment of pseudo code is given in Algorithm

3-2. The general idea is, after detecting and classifying a set of candidates in the current frame, each of these candidates is associated with either 1) an existing track in the history or 2) a new track starting from this candidate. For each existing track, a projected candidate based on DOF is tentatively set as the track's candidate (with the flag "DOF") for the current frame. If a detected candidate can be associated with this track, the detected candidate replaces the DOF based tentative candidate. The result of the first pass is a set of tracks. Each track contains candidates associated across all or an upstream portion of the short range frames. For each covered frame, the candidate in the track can be either a detected candidate or a DOF based candidate. A DOF based candidate may be out of the image view, simply acting as a dummy node in the track. A visualization of the first pass result is given in Figure 3-20, where each row presents a track and the frame indices increase as the tracks move downstream. A solid circle represents a detected candidate and a halo circle represents a DOF based candidate.

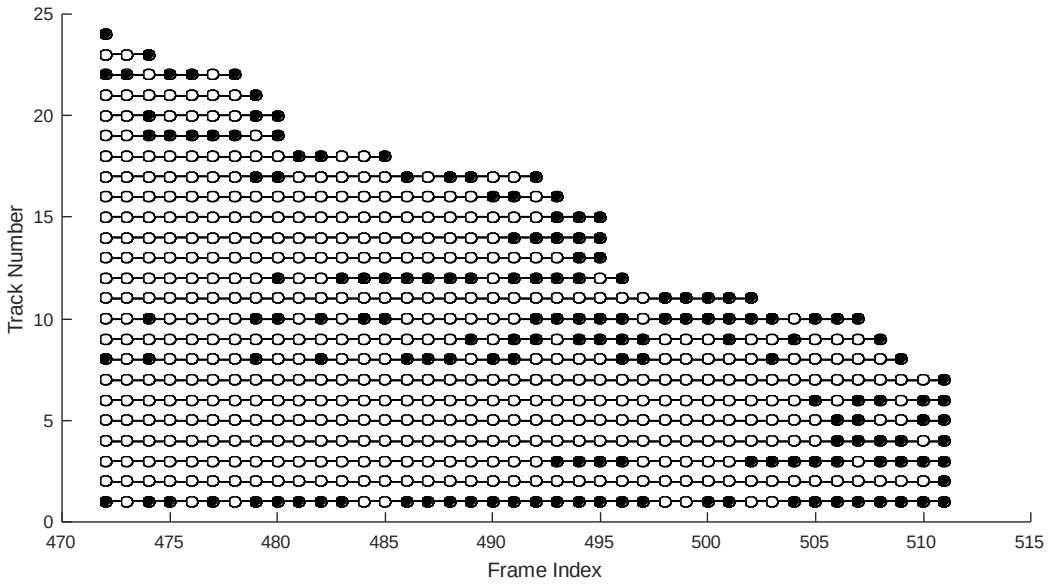


Figure 3-20 Demonstration of the first pass results of short range initialization.

Algorithm 3-2

First Pass of Short Range Initialization

Inputs:

* Upstream frame index $\mathbf{i_u}$
* Downstream frame index $\mathbf{i_d}$

Steps:

```
initialize a track set T to {};
for frame index i = i_d to i_u
  CD ← detected and classified candidates in frame i;
  if T is not empty
    CT ← {};
    for T_t in T
      cand_{t|i+1} ← T_t's candidate for frame i+1;
      cand_{t|i} ← projection of cand_{t|i+1} in frame i using DOF;
      add cand_{t|i} to CT;
    end for
    S ← a matrix of score zeros;
    for CD_d in CD
      for CT_t in CT
        S_{d,t} ← association score between CD_d and CD_t;
      end for
    end for
    for S_{d,t} in S
      if S_{d,t} is not the maximum in S_{:,t}
        S_{d,t} = 0;
      end if
    end for
    for S_{d,t} in S
      if S_{d,t} is the maximum in S_{d,:} and S_{d,t} != 0
        set CD_d as T_t's candidate for frame i;
        take CD_d out of CD;
      end if
    end for
    for T_t in T
      if T_t has no candidate for frame i yet
        set CT_t as T_t's candidate for frame i with flag "dof";
      end if
    end for
  end if
  for CD_d in CD
    create a new track T_new in T;
    set CD_d as T_new's candidate for frame i;
  end for
end if
```

Outputs:

* Track set **T**

In line 14 of the pseudo code, an association score is calculated between a detected candidate and a DOF based candidate. Such association considers both the state machine of traffic signals and the distance between these two candidates in the image coordinate system. Precise calculation is formulated in Equations 3.35 – 3.38. In these equations, all possible colors of a candidate are considered, weighted by their corresponding classification scores. The first candidate is the one related to an earlier state of signal color and the second candidate is the later. In the case of backward DOF, the DOF based candidate represents the state of a later (downstream) frame, although its position is projected onto the current frame using DOF. So, the first candidate in the equations is the detected candidate and the second candidate is the DOF based candidate. When the DOF based candidate is projected from an upstream frame, as will be the case in the second pass and later long range downstream tracked detection, the first candidate is the DOF based candidate and the second candidate is the detected candidate.

$$as = \max\left(as_{c0,c1}\right) \mid c0, c1 \in \{red, green, yellow, unknown\} \quad (3.35)$$

$$as_{c0,c1} = \max\left(pcs_{c0,c1}, 0\right) \times e^{ccs_{c0,c1}} \quad (3.36)$$

$$pcs_{c0,c1} = \begin{cases} 1 - pc, & if c0 = c1 \in \{red, green, yellow\} \\ 1 - pc/3, & if\ c0 \to c1 \in \{green \to yellow, yellow \to red\} \\ & and\ pc_x \le 0\ and\ pc_y \le 0 \\ 1 - pc/6, & if\ c0 \to c1 = red \to green \\ & and\ pc_x \ge 0\ and\ pc_y \ge 0 \\ 0, & otherwise \end{cases} \quad (3.37)$$

$$ccs_{c0,c1} = \sqrt{cs_{c0} \times cs_{c1}} \quad (3.38)$$

where,

as = the final association score between two candidates,
$c0$ = the assumed color state of the first candidate,
$c1$ = the assumed color state of the second candidate,
$as_{c0,c1}$ = the association score assuming the color change c0 → c1,,
$pcs_{c0,c1}$ = the position change score assuming the color change c0 → c1,
$ccs_{c0,c1}$ = the color state change score assuming the color change c0 → c1,
pc = the distance between the centers of the two candidates divided by the minimal radius of the two candidates,
cs_{c0} = the color classification score for color state c0 of the first candidate,
cs_{c0} = the color classification score for color state c1 of the second candidate.

Second Pass – Select and Prune

After the first pass, a specified number of tracks are selected and pruned in the second pass. In terms of selecting the tracks, a stability score is defined for each track as the sum of the highest color classification scores of its detected candidates (i.e., excluding DOF based candidates). Such stability score accounts for both the persistence of detection history and the detection reliability. The tracks with the top N (default to 5) stability scores are selected for further pruning. In the pruning process, the algorithm started from the upstream frame and goes forward until the downstream frame. For each track, the algorithm tries to replace each DOF based candidate with a new detection within a restricted region around that DOF based candidate. When the most downstream candidate in a track is met, the algorithm attempts to use forward DOF to further track and detect new candidates if the end of the short range is not reached. This attempt stops once no more detection is reported in the next frame. A segment of pseudo code is given in the following. Results are shown in Figure 3-21, with the selected and pruned tracks highlighted in red.

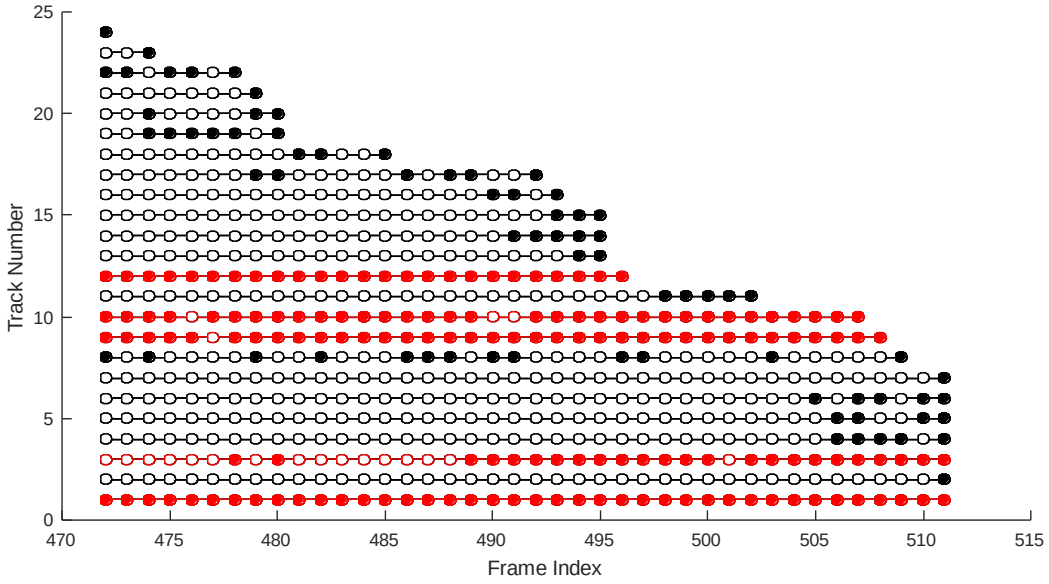


Figure 3-21 Demonstration of the second pass result of short range initialization.

Algorithm 3-3

Second Pass of Short Range Initialization

Inputs:

* Upstream frame index $\mathbf{i_u}$
* Downstream frame index $\mathbf{i_d}$
* Track set **T** from the first pass

Steps:

```
1  Reduce T with up to N top tracks left;
2  for frame index i = iu to id
3    for Tt in T
4      if Tt has a candidate for frame i
5        candt|i ← Tt's candidate for frame i;
6      else if Tt has a candidate for frame i-1;
7        candt|i-1 ← Tt's candidate for frame i-1;
8        candt|i ← projection of candt|i-1 in frame i using DOF;
9        flag candt|i with "DOF";
10     end if
11     if candt|i is not null and flagged "DOF"
12       CD ← detected and classified candidates in frame i within
              a region co-centered with candt|i but with twice
              of radius;
13       asmax = 0;
14       dmax = -1;
15       for CDd in CD
16         as = association score between CDd and candt|i;
17         if as > asmax
18           asmax = as;
19           dmax = d;
20         end if
21       end for
22       if asmax > 0
23         set CDdmax as Tt's candidate for frame i;
24       end if
25     end if
26   end for
   end for
```

Outputs:

* Track set **T**

3.4.5 Long Range Tracked Recognition

With pruned tracks, frame ranges further upstream and downstream out of the short range will be processed in a tracked manner. For each track, long range downstream tracked recognition only takes place when the head of the track reaches the downstream end of the short range. As the tracked recognition goes backward and forward on both ends, new DOF based candidates are projected on the fly to provide a tracked region for detection. The long range recognition stops if no detection in the tracked region can be found. Detailed algorithm is given in Algorithm 3-4. Figure 3-22 gives an example of the long range tracked result. The blue portions of the selected red tracks correspond to long range recognition results upstream and downstream of the short range.

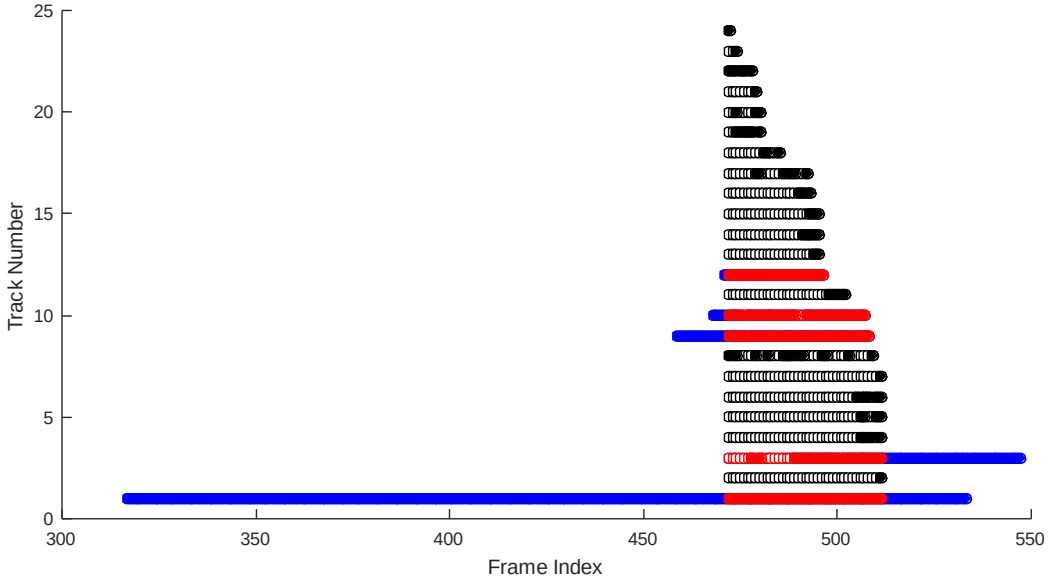


Figure 3-22 Demonstration of the long range tracked recognition result.

Algorithm 3-4

Long Range Tracked Recognition

Inputs:

* Upstream frame index $\mathbf{i_u}$
* Downstream frame index $\mathbf{i_d}$
* Track set **T** from the short range initialization

Steps:

```
1  I = {iu, id};
2  steps = {-1, 1};
3  for dir = 0 to 1 % assuming 0-indexing
4    halt = false;
5    while not halt
6      halt = true;
7      prei = Idir;
8      curi = Idir + stepsdir;
9      Idir ← curi;
10     for Tt in T
11       if Tt has a candidate for frame prei
12         candt|prei ← Tt's candidate for frame prei;
13         candt|curi ← DOF based projection of candt|prei in
                        frame curi;
14         CD ← detected and classified candidates in frame
                curi within a region co-centered with candt|curi
                but with twice of radius;
15         asmax = 0;
16         dmax = -1;
17         for CDd in CD
18           as = association score between CDd and candt|curi;
19           if as > asmax
20             asmax = as;
21             dmax = d;
22           end if
23         end for
24         if asmax > 0
25           set CDdmax as Tt's candidate for frame curi;
26           halt = false;
27         end if
28       end if
29     end for
30   end while
31 end for
```

Outputs:

* Track set **T**

3.5 Summary

A comprehensive methodology is proposed in this chapter, highlighting a generic TSR module that works on individual frames and a sophisticated spatiotemporal framework that considers efficient and reliable TSR in the processing of a lengthy video. The proposed detector relies on no empirical parameters from training data, but is still controllable in an intuitive way according to the expected effects of various features of conspicuity. The proposed color classifier uses sample data to train histograms for different signal colors, but the classification decision is made in a relative way between three expected colors, so the sample data can be totally independent of the testing data in terms of the cameras being used. The spatiotemporal framework helps to zoom into relevant frames in a lengthy video and allow temporally coordinated TSR. The framework does not rely on highly accurate position data, because the temporal coordination considers a buffered short range for initialization. Long range tracked recognition is expected to be fast and reliable based on stable detection tracks obtained from the short range initialization. The next chapter describes data collection for testing the proposed methodology.

CHAPTER 4

DATA DESCRIPTION

4.1 Overview

A total of 21 videos are used to evaluate the proposed methodology. These videos were collected as part of a Head Pose Validation (HPV) data set by the SHRP 2 data collection team and shared via Oak Ridge National Laboratory (ORNL) as sample data for this research. All videos were recorded through traveling an identical route in different days and times. The route was 18 miles long and nearly half of the mileage was on freeway (US Highway 460, Blacksburg, Virginia). Figure 4-1 gives a map visualization of this route alongside a rectified sample frame of approaching a signalized intersection. Each traversal of the route was around 30 minutes and encountered 7 signalized intersections for 8 times (i.e., one intersection was passed twice in different directions). As a result, a total of 168 navigations through signalized intersections were recorded, covering different types of movements and lighting conditions (Figure 4-2). For each video, a log file with other channels of sensor data was also provided. This log file contains GPS based latitude, longitude, and speed readings, 3D vehicle acceleration rates, and ambient exterior light level. Further details of these data and data reductions are given in the rest of this Chapter.

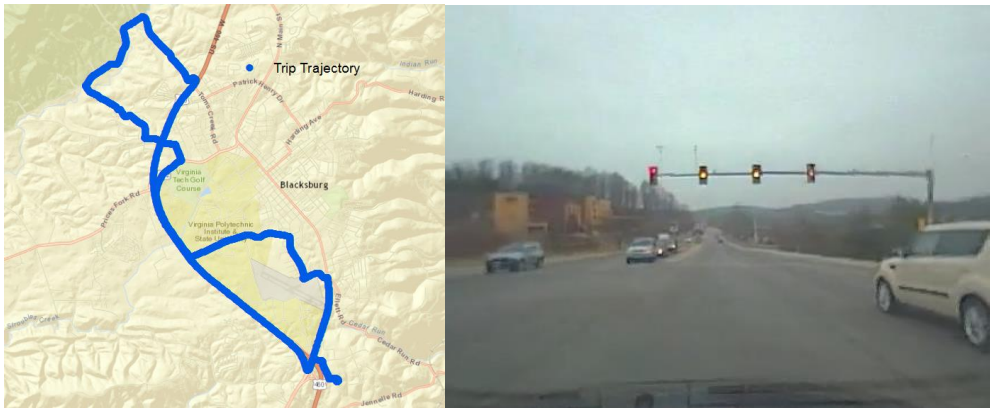


Figure 4-1 The HPV trial route and a sample frame.

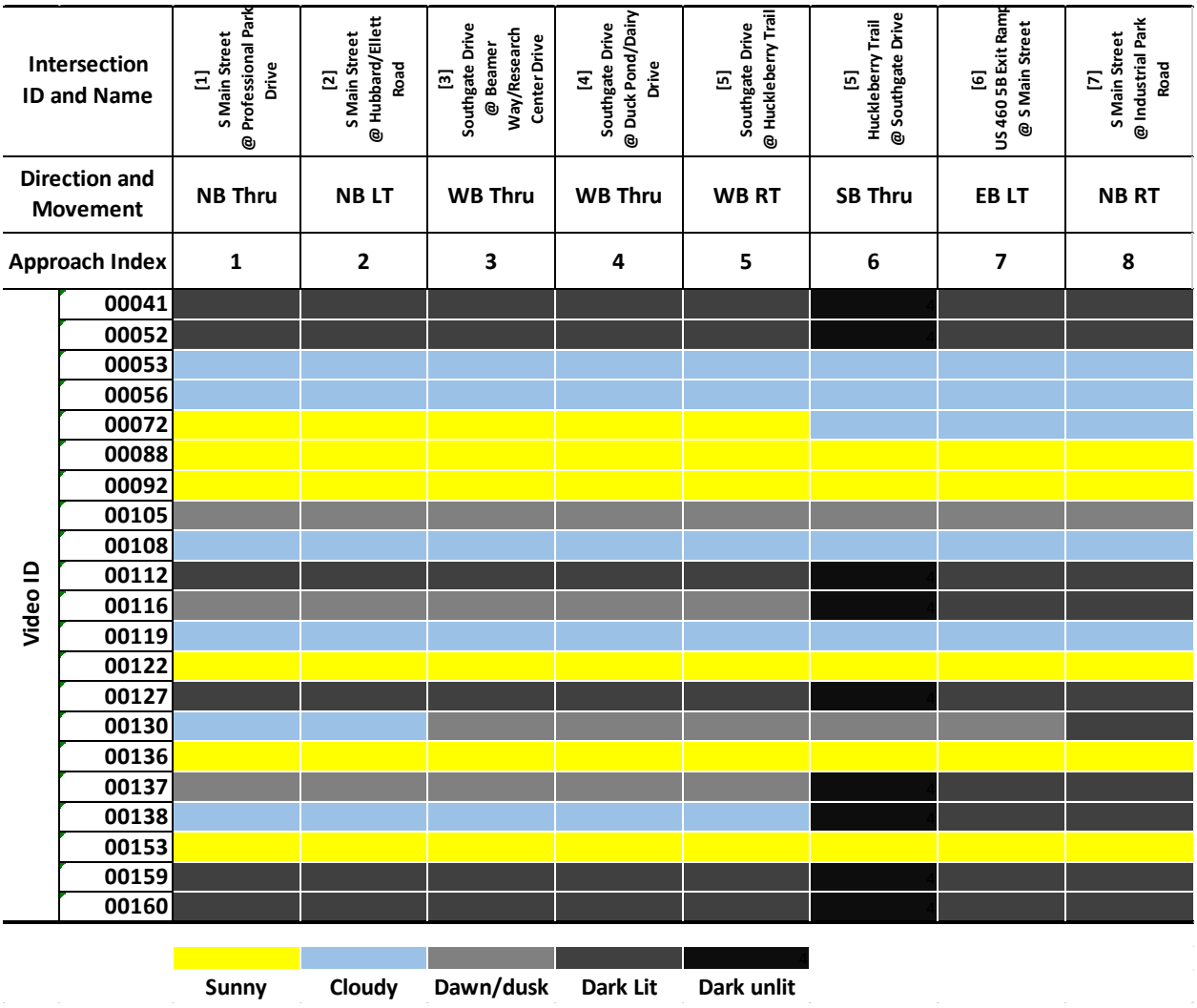


Figure 4-2 Signalized intersection navigations and light conditions.

The five lighting conditions in Figure 4-2 were manually accessed by human reviewers. A quantitative summary of the frames labeled under these five lighting conditions is shown as average lightness histograms in Figure 4-3. The lightness was measured using the L channel of the CIELab color model.

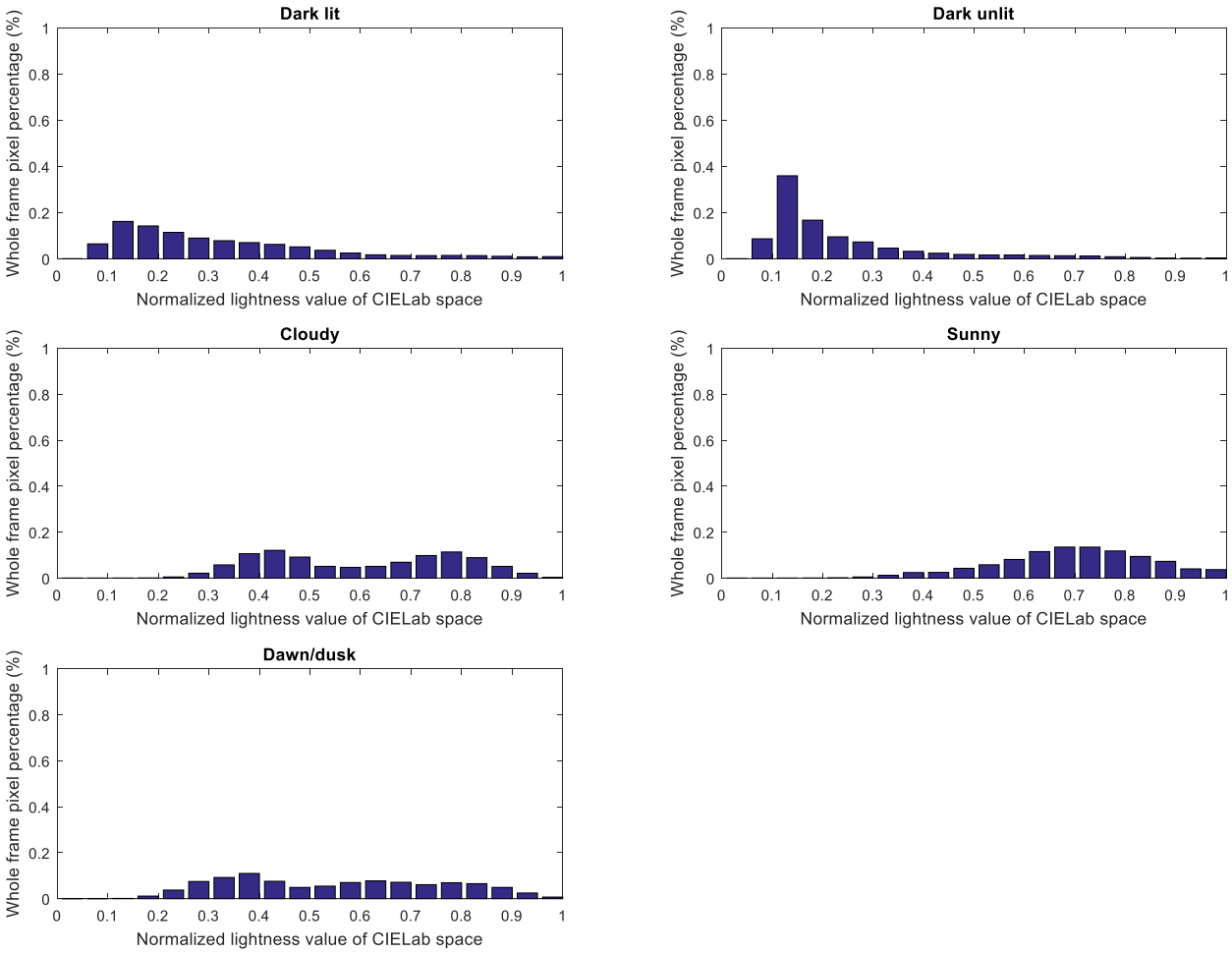
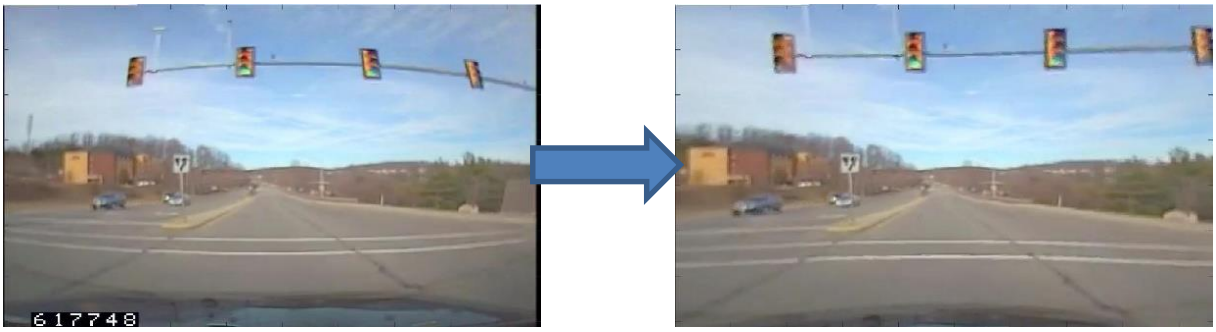


Figure 4-3 Average whole-frame lightness histograms of various lighting conditions.

4.2 Videos

The videos were recorded by a camera behind the windshield near the rare-view mirror. Colors were coded in RGB channels. The camera frame rate was 14 fps and the size of frame is 480x356 (width by height) pixels. An original field of view (FOV) was 83 degrees with the use of a fisheye lens, which introduced significant radial distortions to objects in the view. Camera calibration parameters were used to rectify the videos. Figure 4-4 demonstrates the conversion from an original distorted frame to a rectified frame. After rectification, a portion of the outer

pixels are warped out of the original frame and trimmed. As a result, the rectified frame has a slightly reduced FOV (around 70 degrees) with the same frame size. Note, rectification may not be possible in some cases when camera calibration information is not retrievable. For the proposed traffic signal recognition method, rectification is not a hard requirement, although it provides various image processing advantages, such as preservation of straightness of lines.



A Distorted Frame The Rectified Frame

Figure 4-4 Radial distortion with a relatively wide field of view.

It is worth highlighting two major challenges presented by these videos. The first challenge is related to low pixel resolution. Recall the 480x356 frame size with a rectified FOV of about 70 degrees. Such combination implies a relatively zoomed out view in which, even at a near distance, an object may still look far and small. In addition, the borders of objects are relatively blurred. Figure 4-5 illustrates the typical size and appearance of traffic signal lens at different upstream distance level. Not only is the number of pixels of the target objects limited, but also is the object outline unclearly defined. Another challenge comes from the wide range of lighting conditions. One of the artifacts caused by extreme environmental lights is color oversaturation. As illustrated in Figure 4-6, oversaturation can occur in many cases. In bright sunny days (Figure 4-6a), the photons emitted by the traffic signal lenses plus the ambient photons would surpass the upper intensity threshold of the camera's sensor range and cause white-out pixels. The condition is worse

at sunrise or sunset when the sun was behind the traffic signals along the camera's optical axis, creating a severe backlighting effect (Figure 4-6b). At night, the "halo" or "blooming" effect occurred where the pixels of a traffic signal lens were oversaturated, leaving recognizable colors only in the surroundings (Figure 4-6c). Cloudy daytime and similar light conditions, in contrast, are desired situations where the oversaturation problem was minimized. Besides oversaturation, variable lighting conditions also introduced wild variation of perceivable traffic signal colors. All the above challenges make the videos suitable for testing the robustness of any traffic signal recognition algorithms.

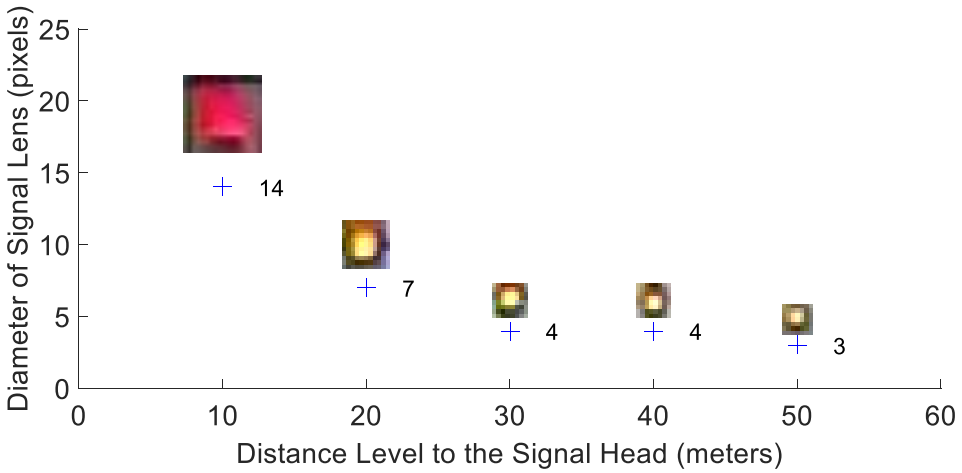


Figure 4-5 Amount of pixels of the signal lens as a function of the upstream distance.

(a) **(b)** **(c)**

Figure 4-6 Oversaturated pixels of the lenses.

4.3 Log File and Position Data

The log file associated with each video contains data from other sensors synchronized by millisecond timestamps. Table 4-1 summarizes the list of data channels as well as their approximate data frequencies. Among these data, only the GPS latitude and longitude readings (under the WGS84 coordinate system) are used as inputs to the proposed geo-filtering method. GPS data were reported less frequently than the other sensor data, because a GPS receiver needs to process signals from visible satellites and estimate the current position over a short course of time. The GPS receiver used in the SHRP 2 NDS study was a Fastrax UP500 *(65)*. This model of GPS receiver uses two satellite-based GPS augmentation systems, the Wide Area Augmentation System (WAAS) and the European Geostationary Navigation Overlay Service (EGNOS), to improve positioning accuracy. However, the accuracy of the GPS receiver could vary due to different factors, such as the number of visible satellites and the angles from the receiver to the satellites. According to the WASS specification, the position accuracy should be no worse than 7.6 meters (25 feet) at least 95% of the time and field measurements have shown horizontal accuracy better than 1 meter (3 feet 3 inches) and vertical accuracy better than 1.5 meters (4 feet 11 inches) *(66, 67)*. EGNOS was found to provide a similar range of accuracy *(68)*.

Table 4-1 Data Entries and Frequencies in the Log File

Sensor	Data entries	Approximate Data frequency
Inertial accelerate sensor	*Longitudinal (x) acceleration rate* *Lateral (y) acceleration rate* *Vertical (z) acceleration rate*	10 Hz
Gyroscopic sensor	*Angular velocity around the vertical axis*	10 Hz
GPS receiver	*Speed* *Heading* *Longitude* *Latitude* *3D Positional dilution of precision*	1 Hz
Ambient light sensor	*Ambient exterior light*	5 Hz

In order to populate the latitude and longitude coordinates to all frames, timestamps were first snapped to their nearest frame numbers and linear interpolation was performed to fill frames without direct GPS readings. More specifically, the GPS reading at timestamp T was assigned to frame N according to the following equation:

$$N = \left\lfloor \frac{T}{1000} * FR \right\rfloor + 1 \qquad (4.1)$$

where,

N = the number of the frame to be snapped to,
T = the timestamp, milliseconds,
FR = the video frame rate, FPS,
$\lfloor \cdot \rfloor$ = the floor function, e.g., $\lfloor 3.7 \rfloor = 3$.

Because the video frame rate was larger than the GPS reading frequency, only a relatively equally spaced fraction of the frames were assigned a GPS reading. In order to populate the frames without direct assignment, linear interpolation with respect to time (i.e., frame number) was used to derive latitude and longitude coordinates between every successive pair of directly assigned frames (Equation 4.2).

$$\begin{cases} \alpha_\Delta = \dfrac{I_{k+1} - I_k}{i - I_k} \\ lat_i = (1 - \alpha_\Delta) * lat_{I_k} + \alpha_\Delta * lat_{I_{k+1}} \\ lon_i = (1 - \alpha_\Delta) * lon_{I_k} + \alpha_\Delta * lon_{I_{k+1}} \end{cases} \qquad (4.2)$$

where,

I_k = the index of the kth frame with direct GPS reading,
I_{k+1} = the index of the k+1th frame with direct GPS reading,
i = the index of the ith frame, subject to $I_k < i < I_{k+1}$,
α_Δ = The relative time progress from frame I_k to frame i,
$lat_{i/I_k/I_{k+1}}$ = the latitude of the i/I_k/I_{k+1}th frame,
$lon_{i/I_k/I_{k+1}}$ = the longitude of the i/I_k/I_{k+1}th frame.

Figure 4-7 illustrates the interpolation procedure. Although no additional smoothness is achieved by linear interpolation, the density of the direct GPS readings can approximate the

curvatures of vehicle trajectories reasonably well. Also, a smoother interpolation is not necessarily more accurate and may even exaggerate GPS errors.

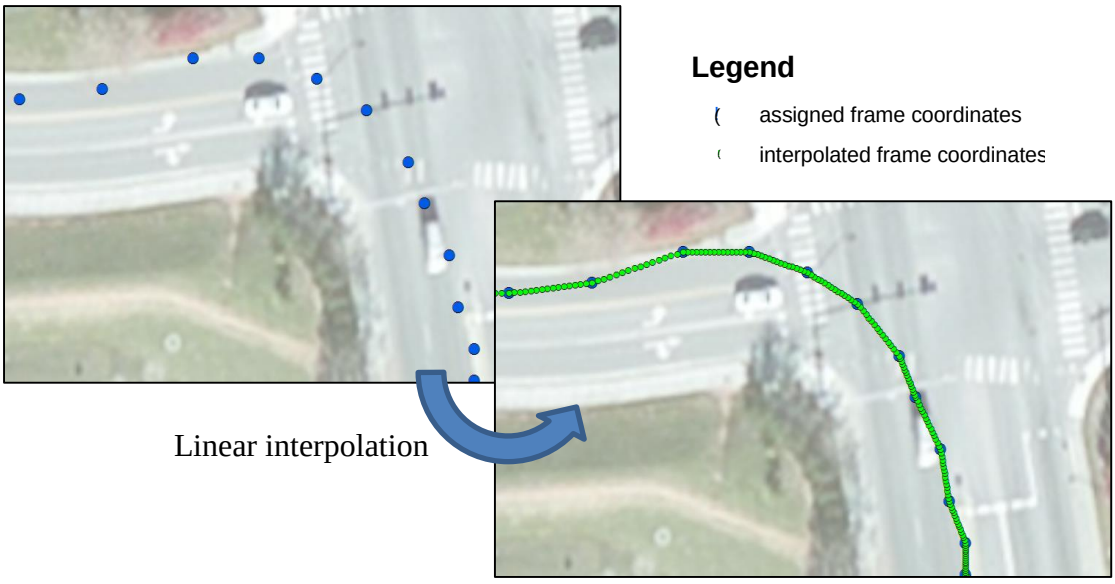


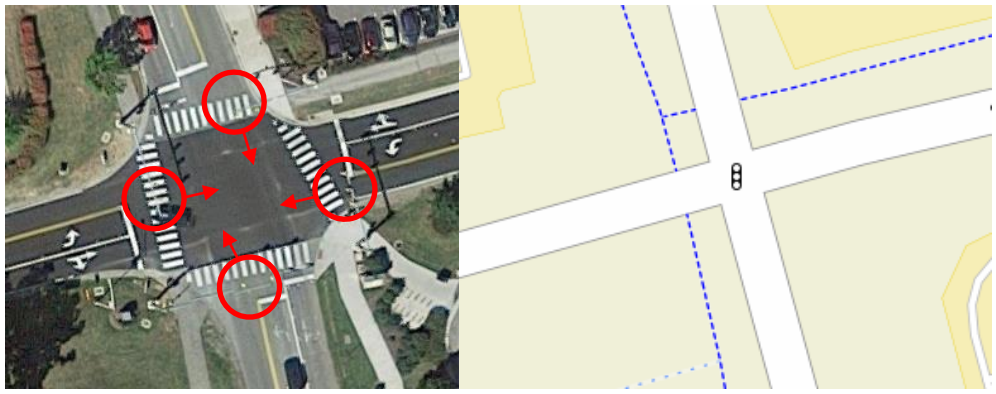
Figure 4-7 Interpolation of GPS coordinates.

4.4 Traffic Signal Map

OpenStreetMap™ (OSM) was chosen as a mapping data source. The OSM data consist of three prime elements: nodes, ways, and relations. Nodes are individual points to mark locations, e.g. intersections. Ways are a series of line segments connecting several nodes. Ways are used to create roads, paths, rivers, etc. An area can be represented by a closed way. Relations are groupings of ways or areas based on their logical relationship. These data can be queried over the internet using OSM's uniform resource locator (URL) based API. A query URL returns data within a latitude-longitude defined bounding box in the XML file format. In this study, the initial bounding box around the entire route of the video is resized by a factor of two in both dimensions to form

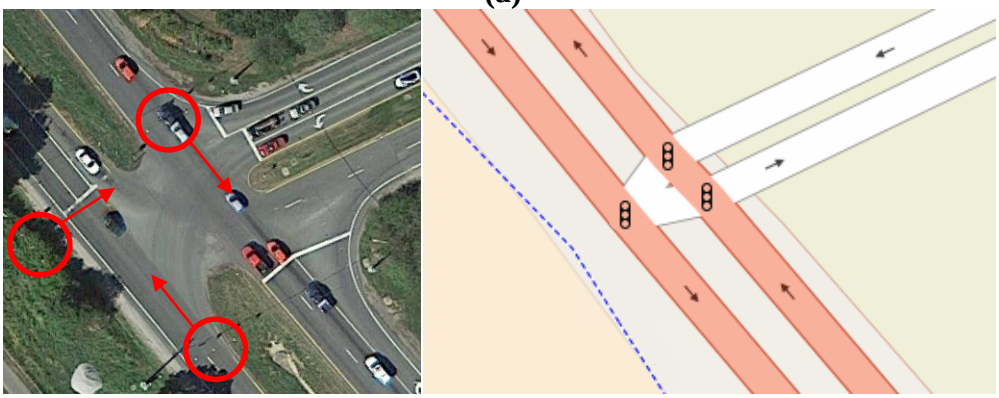
an expanded bounding box to query OSM data. The buffered bounding box can prevent missing data that are slightly outside the borders of the original bounding box.

Traffic signals are tagged as nodes with their "*highway*" attribute having the value of "*traffic_signals*" *(69).* According to the OSM documentation, "the mapping of traffic signals is an abstraction that the particular junction or way is regulated by traffic lights." Therefore, a traffic signal node is not conceptually related to a particular traffic control device. For example, in Figure 4-8a, a signalized intersection is represented as a node connecting four ways in OSM. This node is tagged as the only traffic signal node for that intersection, even though there are four sets of overhead traffic signals on the far side of the intersection for each approach. When multiple nodes are used to represent a more complicated highway intersection (e.g., Figure 4-8b), all nodes could be tagged as traffic signal nodes but their positions do not correspond to the actual traffic signals. Therefore, the OSM traffic signal nodes should not be used to locate actual traffic signals, rather, they should be used as a rough estimation of the intersection as a whole. Typically, when multiple traffic signal nodes are closer to each other than a certain distance (e.g., 50 m), their average position can be used as an approximation of the center of the intersection.



satellite image OSM data model

(a)



satellite image OSM data model

(b)

Figure 4-8 Illustration of OpenStreetMap™ traffic signal nodes.

For the purpose of verifying the accuracy of the OSM data, key points of the seven traversed intersections were manually located using satellite images on Google Maps™. Detail information is given in Table C-2 and Table C-3 in Appendix C. For each intersection, the manually extracted intersection center (Table C-3) and the average location of the OSM traffic signal nodes (Table 4-2) are compared. Figure 4-9 shows the comparison results. The OSM based estimations provide relatively good accuracy of less than 5 meters away from the manually coded positions, which is acceptable for a rough estimation of intersection vicinity from each video frame.

Table 4-2 OSM Traffic Signal Nodes of Traversed Signalized Intersections

Intersection	Traffic Signal Node ID	Latitude	Longitude
[1] S Main Street @ Professional Park Drive	*[274633606]*	37.197725	-80.401337
	[726778899]	37.197687	-80.401243
[2] S Main Street @ Hubbard/Ellett Road	*[216434379]*	37.209364	-80.399353
	[721834927]	37.209383	-80.399213
[3] Southgate Drive @ Beamer Way/Research Center Drive	*[216441656]*	37.217342	-80.419160
[4] Southgate Drive @ Duck Pond/Dairy Drive	*[721757100]*	37.216285	-80.423660
[5] Southgate Drive @ Huckleberry Trail	*[216459765]*	37.213135	-80.431888
	[726771247]	37.213092	-80.432138
	[1468455063]	37.213234	-80.432009
[6] US 460 5B Exit Ramp @ S Main Street	*[726671503]*	37.191650	-80.403763
	[726672094]	37.191697	-80.403894
[7] S Main Street @ Industrial Park Road	*[721834758]*	37.193972	-80.402747
	[721834885]	37.194042	-80.402874

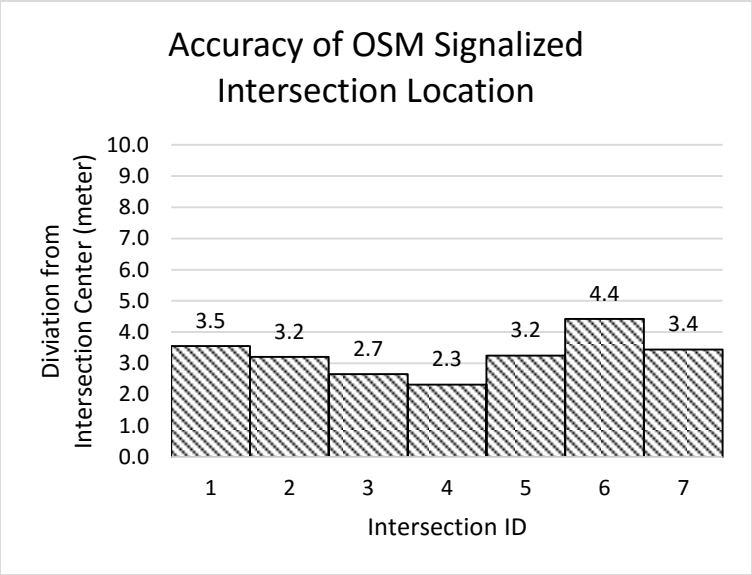


Figure 4-9 Accuracy of intersection center estimate based on OSM traffic signal nodes.

4.5 Signal State Ground Truth

Annotating traffic signals in each frame is the most critical data reduction effort. Resulting annotations will be used to evaluate the accuracy of the proposed methodology. Also, sampled annotations can be used as training data for some compared methods that require device dependent training samples.

In each frame, every active traffic signal lens (excluding pedestrian signals) facing the subject approach was annotated by a rectangular bounding box. The bounding box was drawn to enclose the lens area only. However, due to image quality (e.g., oversaturation), there was a certain degree of uncertainty when drawing the bounding box. Such uncertainty was inevitable, but was mitigated by enforcing a rule that the bounding box of the same signal lens should not increase in size as the annotation started from a nearest non-void frame and progress backwards. Starting from the nearest frame gave the annotator the largest and clearest target objects and, as the annotation

progressed backwards, the bounding boxes of the previous frame became a helpful gauge of the bounding boxes in the current frame. Annotation stopped when all signals were too small to locate or when an upstream distance was exceeded, whichever happened first. Occlusion and flashing mode could result in discontinued appearance of the same signal lens. In such cases, the bounding box was only drawn for frames when the lens was visible or lit.

In addition to the bounding box, each signal lens was also annotated with its signal color and an identification number corresponding to the signal head it belonged to (referred to as the signal head id hereafter). The signal head id is local to each approaching instance. Numbering of the signal head ids started from the *critical signal head*, defined as the signal head that regulated the traveled lane and movement of the subject vehicle. The critical signal head was given an id 0. After that, the id of every next signal head to the left and to the right of the critical signal head was decreased and increased by 1, respectively. For example, given four signal heads among which the second from the left is the critical signal head, their id sequence will be [-1, 0, 1, 2] starting from the left. Therefore, even in the same approach to the same intersection, if two approaching instances were in different lanes, the two resulting signal head id sequences would be differed by an integer. The advantage of such identification schema is the convenience of locating the critical signal head when movement specific analysis is needed. With the signal head id, bounding boxes across frames can be associated even when a signal state change happened.

In order to facilitate the extraction of the ground truth, an interactive computer program was written in Matlab™ to provide both visual control and text input functions. As illustrated in Figure 4-10, the interface provides a guiding box (dashed-line) that moves with the mouse cursor so the user is always aware of the current mouse position and the bounding box size. The user can change the width, the height, or both of the guiding box by certain combinations of keyboard

strokes and mouse scroll. The user can also zoom into the region of interest to get a focused view of the signal lens, which is very useful when the traffic signal was far away and only occupies a very small amount of pixels. When the guiding box correctly locate a signal lens, the user can simply left-click the mouse to confirm the annotation, which will bring up a text input dialog for the user to type in the traffic signal state and the signal head id. The confirmed ground truth will appear as a still solid-line box on the frame with information displayed beneath it. If a mistake is made, the user can delete the mistake by positioning the mouse cursor in the box and right clicking the mouse with the Ctrl key pressed. Considering the situation when the vehicle was stopping for the red light or yielding to conflicting traffic under the permissive mode, a large amount of frames may look almost the same. Therefore, a linear interpolation mode was given to the program to provide a certain level of automation. With this mode, the user only need to annotate two end frames and let the program interpolate annotations in between.

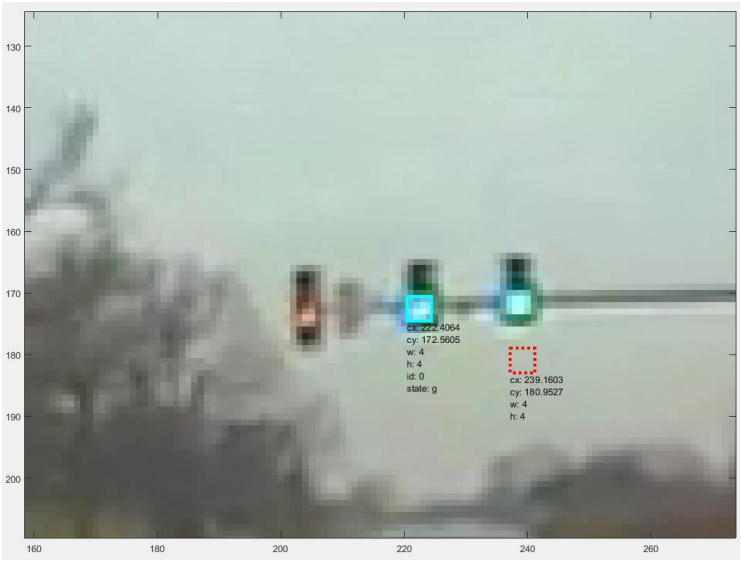


Figure 4-10 Visual interface for ground truth signal state extraction.

Table 4-3 summarizes the statistics of the extracted ground truth and Figure 4-11 visualizes the counts of annotations in different cross classifications in terms of instant signal state, in image size, and the lighting condition.

Table 4-3 Overview of Annotation Results

Statistic name	Statistic value
Total number of annotated frames	30529
Total number of annotations[(1)]	82528
Number of annotations by instant signal state[(2)]	
Red	28588
Green	51076
Yellow	2864
Number of annotations by radius of the maximum bounded circle (in pixels)	
1	7
1.5	1594
2	16633
2.5	19547
3	18160
3.5	14237
4	6403
4.5	3300
5	1440
5.5	466
6	305
6.5	210
7	102
7.5	67
8	39
8.5	12
9	6
Number of annotations by lighting conditions	
Sunny	25463
Cloudy	17121
Dawn/dusk	14141
Dark lit	23598
Dark unlit	2205

(1) An annotation is a bounding box around a traffic signal lens in one video frame with related information. One frame may contain multiple annotations and one physical traffic signal lens may correspond to a set of annotations across multiple frames.

(2) Instant signal state refers to the traffic signal color at the instant of the frame. For solid traffic signals (e.g., ordinary green, red, or yellow), the instant signal state during the corresponding interval is consistent. For flashing traffic signals, such as flashing yellow arrow, the instant signal state changes at a certain frequency during that interval. A flashing lens was only annotated in the frames when it was lit, resulting in non-continuous annotations.

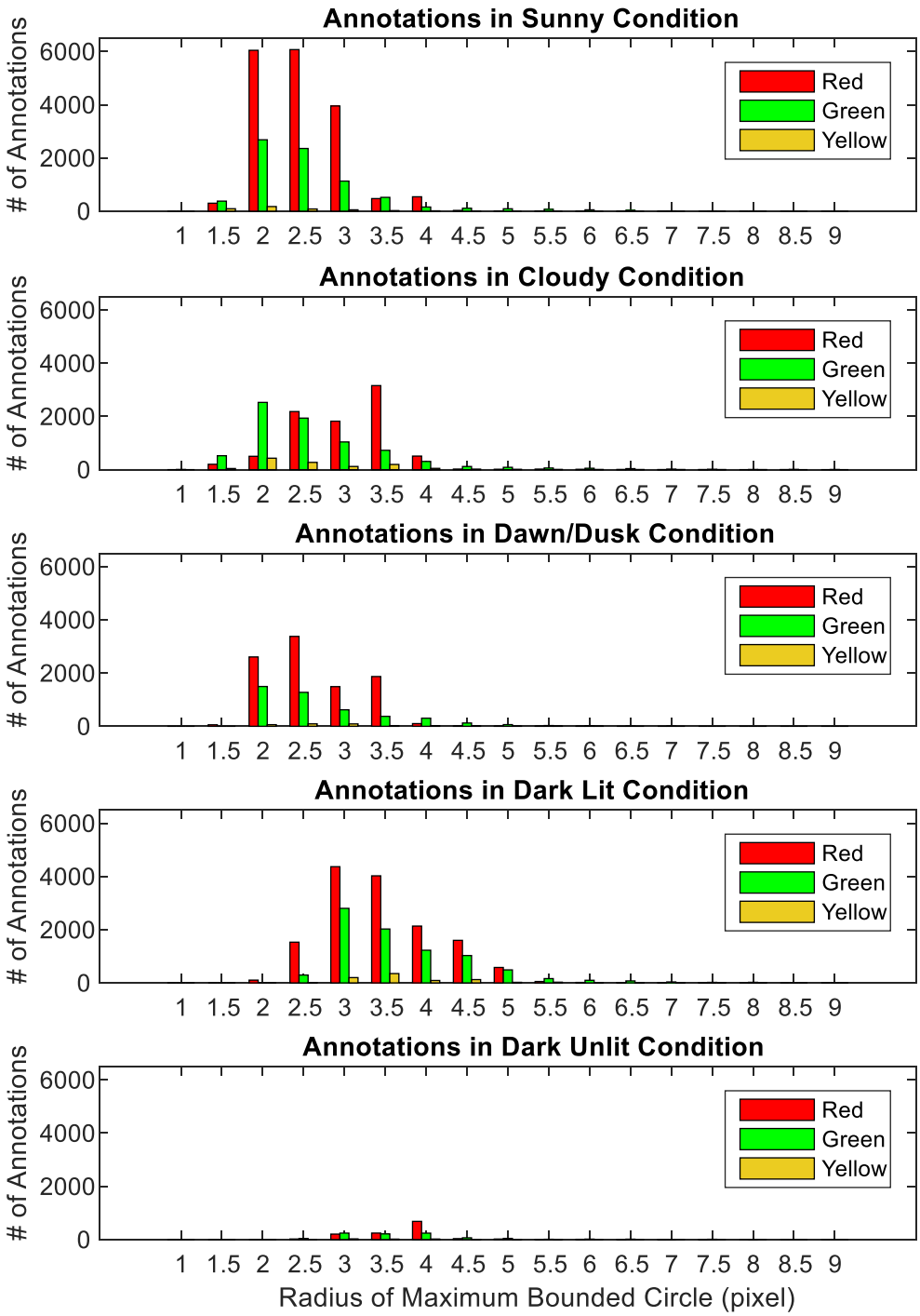


Figure 4-11 Annotation results by cross classification.

CHAPTER 5

EXPERIMENTS AND ANALYSES

5.1 Overview

Experiments were conducted on a 64-bit Windows 10 desktop machine. This machine was equipped with a 2.33-GHz Intel® Core™ 2 Quad CPU (Q8200) and 6 Gigabytes of random access memory (RAM). Also, this machine was constantly connected to the internet, which allowed on-the-fly data acquisition from the OSM servers during the experiment with the spatiotemporal framework. Algorithms were prototyped in the C++ programming language with reference to the OpenCV library for fundamental vision functions. Matlab™ wrapper functions (i.e., mex functions) were written so they could be executed and the results (intermediate and final) could be analyzed in the Matlab™ environment (version R2016a). Direct builds to Windows command line executables are also provided for production purposes. Algorithm parameters could be controlled with a configuration text file that follows the YAML data serialization standards.

5.2 Detection Performance

Since detection is the foundation of the entire recognition pipeline, experiments were performed to assess its accuracy and gain insights to calibrating parameters (e.g., the weights of conspicuity features). A baseline performance measure was given by running the detection algorithm with default parameters over all 30,529 annotated frames. Accuracy and recall are evaluated for various lighting and distance (in terms of target size) categories. Qualitative inspections into sample frames in low performance categories revealed helpful rules for tuning parameters. The baseline performance was also compared to two other closely related methods and the results reveal the robustness of the proposed algorithm.

5.2.1 Baseline Test

In order to gain a baseline assessment of the detector's performance, default parameters listed in Table 5-1 were used to run tests on all 30,529 annotated frames. The tests took over 40 hours to finish, averaging to a processing time of nearly 5 seconds per frame. However, this processing time included loading and saving intermediate data from and to computer hard drives to facilitate later tuned tests, which can be avoided in production.

Table 5-1 Default Parameters for Baseline Test

Parameter	Value
$\{w_L, w_{LC}, w_S\}$	{1, 1, 1}
RAD (the *r* range)	{1,2,3,4,5,6,7,8,9,10}
m (the A2 margin width)	0
h_{peak}	0.9
N_{top}	10
N_{pi}	20
asp_{min}	0.6
pd_{min}	0

Detected candidates were compared to manual annotations so true positives and false positives could be separated. A true positive is defined as a candidate whose center pixel resides in the minimum enclosed circle of any annotation box of the same frame. At this point, no color state was considered because the detection algorithm does not provide a state classification. The assessment of state classification will be addressed in Section 5.3.

Detection performance should be evaluated on a per-frame basis and under various lighting and distance categories. Because the theoretical behavior of the conspicuity based detection is to find the most conspicuous regions in the given frame, it makes little sense to evaluate whether individual traffic signal lenses were detected or not by considering them separately from the frame

context. For example, a traffic signal lens can be missed not because its absolute conspicuity value (which is in itself meaningless) is low but because it is lower than those of other regions in the frame. In other words, the detection performance should be an assessment of how many of the present traffic signals in a given frame can be correctly detected (i.e., the recall) as well as how many false objects are reported among all detections (i.e., the false positive rate = 1 - precision). In different lighting and distance (in terms of the size of the targets) conditions (refer to Figure 4-1 and Figure 4-11), the relative conspicuity of the target traffic signals can be affected differently. For example, in sunny days when the traffic signals are far away, the conspicuity of the targets can be compromised by pixel resolution and overwhelmed by near-by roadside object or the background sky. In some of these cases, finding the exact location of a traffic signal could be extremely ambiguous even to human reviewers (based on data reduction experience). Therefore, frames were evaluated under different scenario categories.

Table 5-2 summarizes the baseline test result in terms of frame counts under various scenario categories and performance ranges. Note, the header of each recall rate column states the exact number of true positives over all annotated signals in the frame, except for zero and full recalls (0 and 1, respectively). The precision columns are corresponding to lower bound precisions. For example, a frame under the 30% precision column is one on which the detection achieved a 30% or more precision.

Table 5-2 Frame Counts of Various Performance Ranges in Different Scene Categories

Light Condition - Max Target Radius	Recall Rate								Precision																	Total
	0	1/4	1/3	1/2	2/4	2/3	3/4	1	0%	10%	15%	20%	25%	30%	35%	40%	50%	55%	60%	65%	75%	80%	100%	
Cloudy	2434	228	934	884	152	557	69	1088	2434	56	19	48	120	220	10	84	344	1	55	251	83	1	2620	6346
2	564	54	270	113	30	145	3	138	564	3	4	13	22	55	2	20	95		12	46	20		461	1317
3	872	128	491	260	80	314	39	404	872	16	6	17	42	102	8	45	167	1	34	114	52	1	1111	2588
4	955	40	121	446	35	77	25	472	955	37	9	15	54	54		19	77		9	84	8		850	2171
5	25	6	29	23	4	9	1	21	25			1		2			1			2	1		86	118
6	11		9	28	3	4	1	26	11			1		3			1			3	1		62	82
7	5		8	13		4		14	5				1	3			1			1	1		32	44
8	2		2	1		2		5	2			1	1				1			1			6	12
9			4			2		8						1			1						12	14
Dark lit	2008	375	1168	763	385	1453	203	2035	2008	88	88	167	389	831	27	378	1343	6	234	896	281	7	1647	8390
2	9		17			8		1	9	3	3	4	5	6	1	2	1						1	35
3	469	180	518	89	180	819	58	704	469	31	34	57	164	304	6	169	495	3	144	286	163	2	690	3017
4	798	97	493	335	174	516	129	1001	798	34	30	62	147	370	16	168	530	3	58	478	85	5	759	3543
5	581	93	113	303	22	96	12	283	581	18	17	39	65	135	4	32	286		31	116	30		149	1503
6	79	4	17	24	8	11	4	35	79	1	3	4	5	11		6	24		1	12	3		33	182
7	47	1	8	9	1	3		9	47	1	1	1	3	5		1	6			3			10	78
8	25		2	3				2	25								1			1			5	32
9																								
Dark unlit	172		137	1		278		155	172	8	9	15	21	51	2	19	60		11	39	40		296	743
2																								
3	74		43			45		15	74	2	5	7	11	21		4	27		4	7	4		11	177
4	61		54			220		136	61	4	4	2	4	23	2	13	23		7	30	34		264	471
5	15		33	1		8		3	15	1		5	6	4		1	9			2	1		16	60
6	10		2			2		1	10			1		1							1		2	15
7	5		4			2			5	1				1		1							3	11
8	7		1			1			7					1			1							9
9																								
Dawn/dusk	3185	57	565	468	30	318	9	510	3185	26	19	52	138	251	7	67	643		39	341	25		349	5142
2	698	12	356	16	7	232	3	94	698	20	13	35	81	110	6	18	200		37	68	21		111	1418
3	1956	38	170	47	13	66	4	24	1956	3	2	6	21	54	1	14	144			26	3		88	2318
4	482	4	25	375	8	13	1	378	482	3	4	11	31	84		33	280		1	243	1		113	1286
5	34	3	9	27	1	5	1	9	34				5	2		1	17		1	2			27	89
6	13		5	1	1	2		5	13					1		1	2			2			8	27
7	2			2					2														2	4
8																								
9																								
Sunny	8977	2	309	443		84		93	8977	44	25	83	98	114		24	226			31			286	9908
2	2946		3	14					2946	2		4	2	4			3						2	2963
3	5362	2	262	215		64		62	5362	32	17	53	65	66		17	138			24			193	5967
4	520		34	179		15		14	520	9	7	24	24	33		5	64			4			72	762
5	78		7	26		3		5	78	1	1	2	5	9		1	12			2			8	119
6	51		2	7		2		5	51				2	2		1	6			1			4	67
7	19		1	2				5	19								2						6	27
8	1							2	1								1						1	3
9																								
Total	16776	662	3113	2559	567	2690	281	3881	16776	222	160	365	766	1467	46	572	2616	7	339	1558	429	8	5198	30529

In order to perceive trends of the results in Table 5-2, a plot of recall versus false positive rate (FPR = 1 – precision) was generated and showed in Figure 5-1. In this plot, each subplot belongs to a cross-category of lighting condition and target size, corresponding to each row in Table 5-2. The total number of frames in each category is printed as "N = ***" in gray color. In each subplot, the x-axis is the FPR and the y-axis is the recall. Each circle on the subplot correspond to all frames with the same performance, while the area of the circle are proportional to the percentage of frames with that performance in that category. Circles are colored more blue as they are closer to the left and upper borders (i.e., FPR = 0% and recall = 100%) and more red as they tend towards the other direction. Therefore, bluish circles corresponds to frames with desired performance while reddish circles are frames that reveal the inadequacy of the baseline setting.

Several trends can be observed in Figure 5-1. First, in almost all scenarios, the frames with at least one target being detected (recall > 0) are always near or above 50%. This implies that over 50% of frames in most scenarios fully or partially satisfy the assumption that the target traffic lights are among the most conspicuous objects in the scene. Second, there is little correlation between the recall and the precision. This complies well with the randomness of the scene complexity and hence of the satisfaction to the conspicuity assumption. When the conspicuity assumption is fully satisfied, the targets can be detected at a perfect recall (= 1) with zero false positives. As the satisfaction of the assumption decreases in various ways, the performance can be roughly anywhere in the plot area. When the assumption is fully violated, both recall and precision drop down to zero. Nevertheless, performance differs among different scenarios. In terms of lighting condition, cloudy days present the most detectable environment and as the target size increases, the total misses are constantly decreasing. This is intuitive because the cloudy days

introduce weak ambient light and help the traffic lights to stand out in the scene. In dawn/dusk, dark lit, and dark unlit conditions, the ambient lights are even weaker than in cloudy days, but due to the sudden increased uses of vehicle headlights and streetlights, the conspicuity of target traffic lights face more competition from these other light sources, even from the highly reflective surfaces such as traffic signs. Sunny days present the most challenging condition with constantly high percentage of total miss frames. In sunny days, the strong ambient light can interfere with the digital imaging of the lightness and even color saturation of traffic lights, such as overexposure. The conspicuity of reflective pavement markings and colorful roadside objects can also be elevated by the ambient light to confuse the algorithm. Though, due to the consideration of contrast in the conspicuity model, traffic lights could still be detected in a decent percentage of frames in challenging sunny conditions.

Without additional knowledge about the scene or spatiotemporal constraints for detection, the conspicuity model is maintaining a balanced accommodation to a wild randomness of scenarios. Further detection improvement can be attempted by adjusting the weights in the model to adapt to certain scenarios better (see Section 5.2.3) or by introducing constraints in a spatiotemporal framework (see Section 0). In order to reveal the advantage of the proposed method, two other detection approaches that also aimed at generic accommodation were compared in the next section.

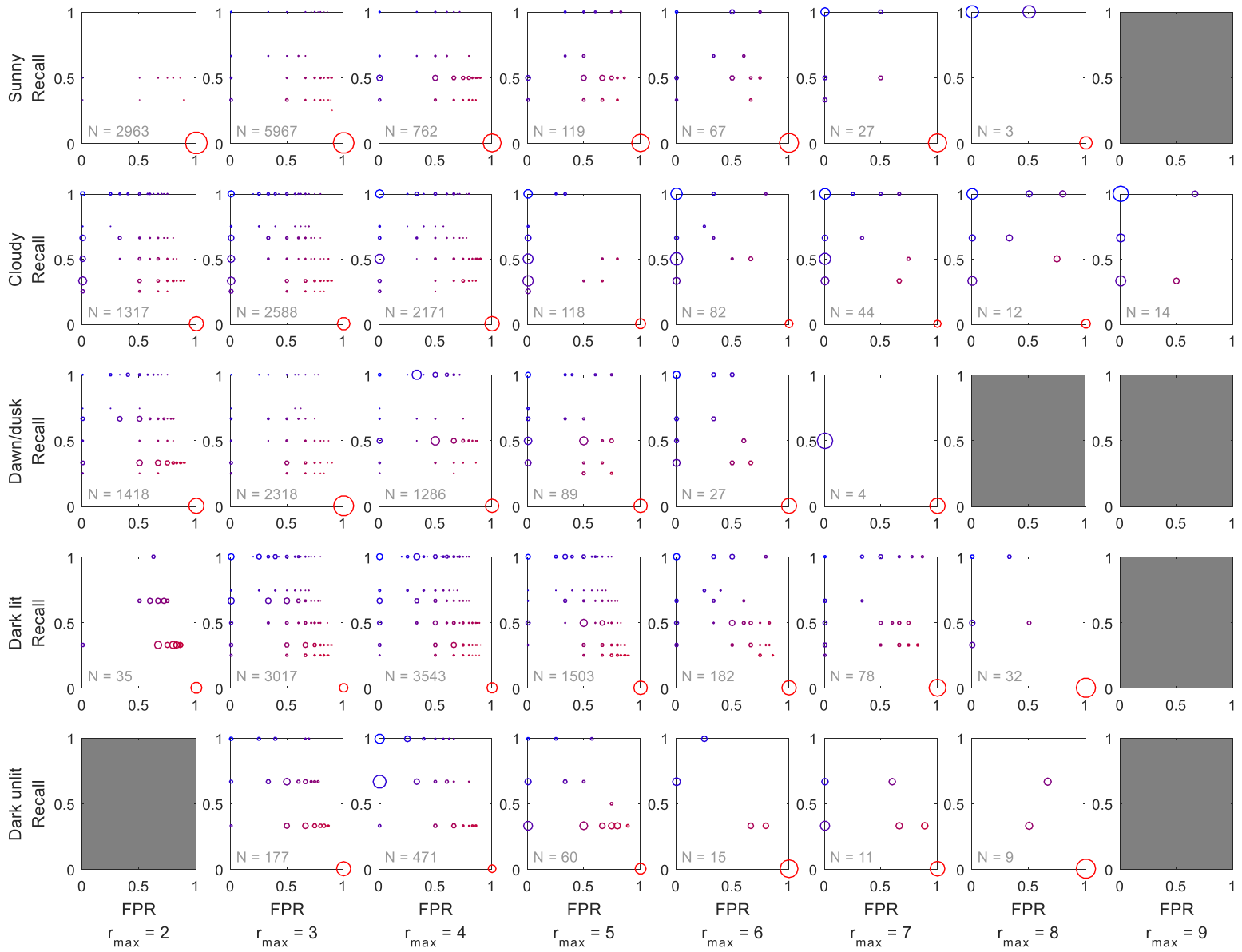


Figure 5-1 Recall v.s. false positive rate (= 1 - precision) of the proposed algorithm.

5.2.2 Comparison to Other Approaches

The two alternative approaches chosen for comparison are based on Siogkas et al. and Charette and Nashashibi *(28, 35)*. Siogkas et al. multiplied the "L" channel to the sum of the "a" and "b" channels of the CIELab space to create an initial feature map. Then, a fast radial symmetry transform was performed on this feature map to generate a heat map of local symmetry (corresponding to the circular shape of signal lenses). Binarization and connected component analysis were used on this heat map to locate the top five dark spots (corresponding to green lights) and the top five bright spots (corresponding to red lights). Charette and Nashashibi converted the input image to a grayscale image and applied the white top-hat transform (difference between an image and its morphological opening) on this grayscale image to highlight spotlights in the scene. Connected component analysis was used to locate the candidate regions. For conciseness, the approach based on Siogkas et al. is denoted as LAB-FRST and that based on Charette and Nashshibi is denoted as GRAY-TOPHAT.

The implementations of the above two methods are as following. For LAB-FRST, the fast radial symmetry transform (FRST) is based on the original implementation by the inventor *(70)*. The positive and negative parts of the CIELab based feature map are transformed separately. The radius range is chosen as 1 to 10 pixels, with 1 pixel step. The alpha parameter of the FRST algorithm is chosen to be 3 and the orientation flag is set to be true. Binarization is based on half of the maximum value in the transformed map and 8-connectivity component analysis is used to locate candidates and determine their sizes. For GRAY-TOPHAT, the implementation is in fact identical to the proposed algorithm, except that the conspicuity value is replaced by the top-hat value calculated on a grayscale image.

Figure 5-2 and Figure 5-3 show the recall versus FPR plots for LAB-FRST and GRAY-TOPHAT, respectively. For comparison, Figure 5-4 plots the average performance of each method under different categories. The proposed method has a higher average recall/lower average FPR than the other two methods do in most cases. For sunny condition, the performances are similar for all methods, again, confirming the challenge of sunny condition. However, as the target size increases, the proposed method and GRAY-TOPHAT show faster improvement in recall than LAB-FRST, while the proposed method also shows reduced FPR. In cloudy conditions, the proposed method constantly outperform the other two methods and the advantage becomes more significant as the targets get closer. In dawn/dusk, the proposed method is beaten by the GRAY-TOPHAT only when the target size is at 7 pixel in radius, but the sample size associated with that category is only 4, which does not grant statistically significant conclusion. In dark lit and unlit conditions, the proposed method always performs better than the other two methods, although the performance does not increase as the target becomes larger.

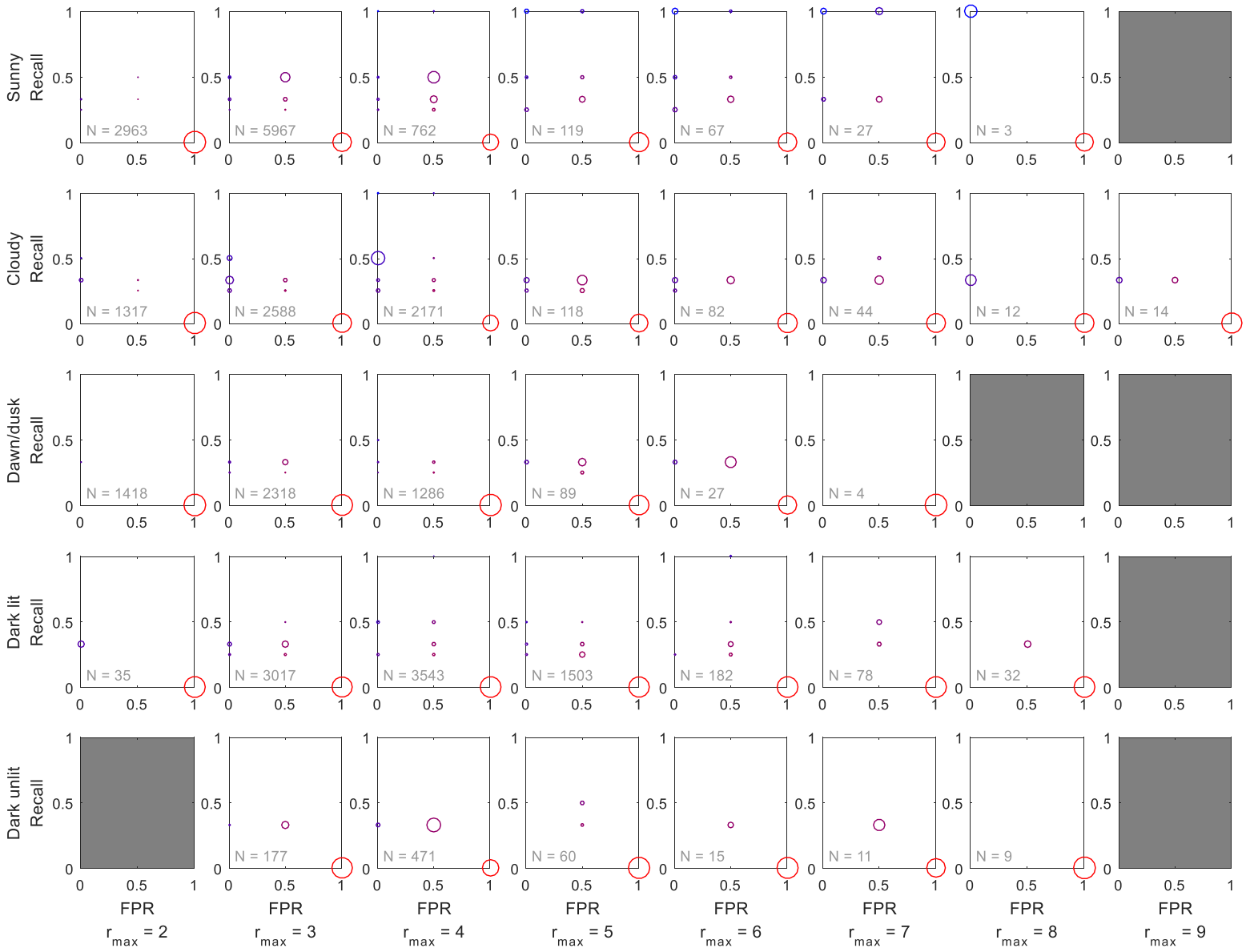


Figure 5-2 Recall v.s. false positive rate (= 1 - precision) of the LAB-FRST algorithm.

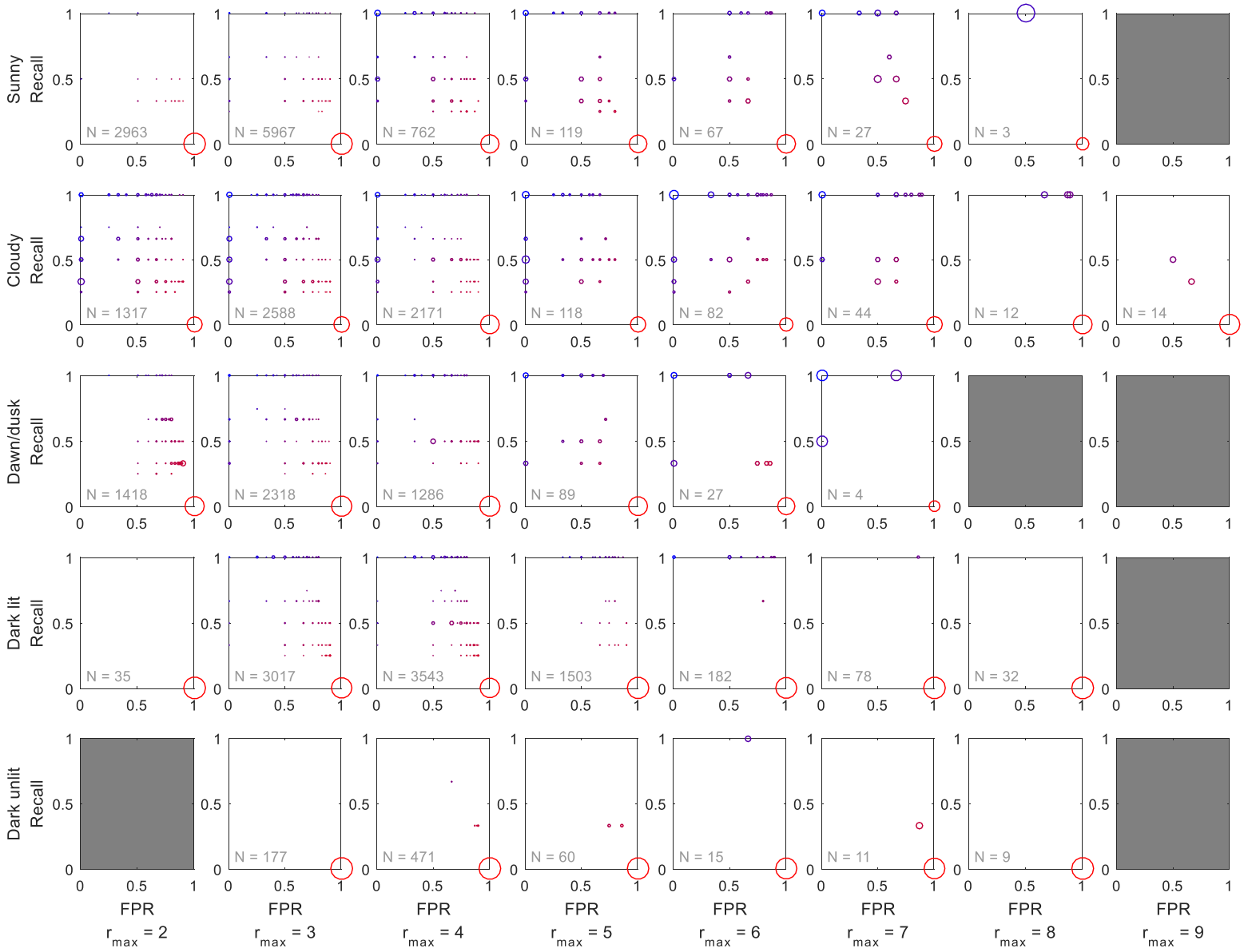


Figure 5-3 Recall v.s. false positive rate (= 1 - precision) of the GRAY-TOPHAT algorithm.

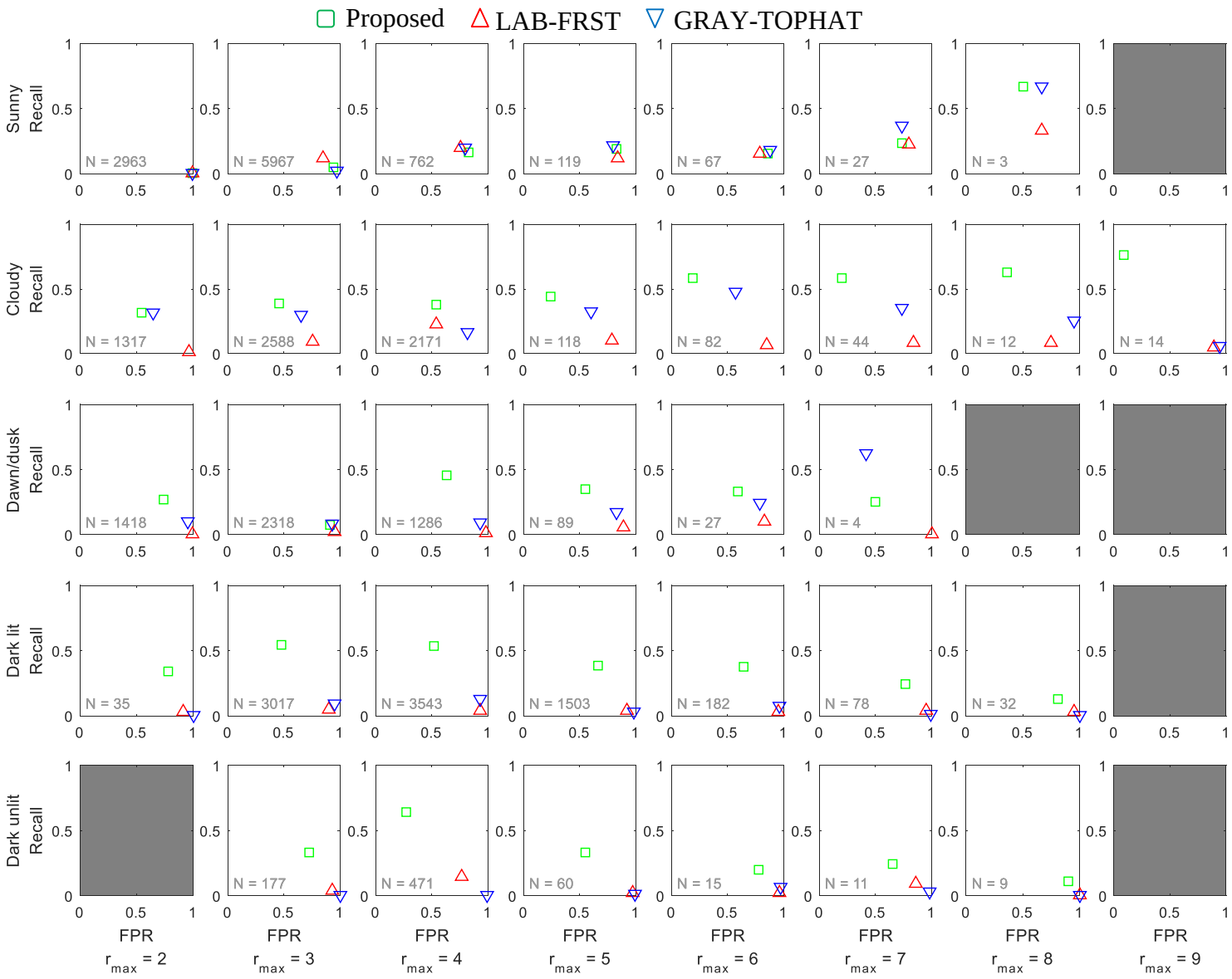


Figure 5-4 Comparison between different methods.

5.2.3 General Tuning Rules

Visual inspection was conducted on select frames to derive helpful tuning rules. Focuses were given to frames with target radius between 3 and 4 pixels with zero recall (as well as precision). All lighting conditions were considered for visual inspection.

In general, in all lighting conditions but sunny, low recalls and high false positive rates were found majorly related to the competitions from other light sources. Examples are given in Figure 5-5 - Figure 5-11, with the conspicuity map and the original image shown side by side. The original image is overlaid with the detected candidates (blue circles) and the annotated ground truths (red squares). Figure 5-5 and Figure 5-6 show two examples in dark lit conditions. The target traffic lights showed relatively high conspicuities in the image, but not as high as some of the other light sources. In fact, among the false positives, some are actually secondary traffic lights (Figure 5-5) and traffic lights for the cross street (Figure 5-6). In Figure 5-5, even the reflection of the target traffic lights on the hood cover were detected. Therefore, the algorithm was working in an expected way, but the complexity of the scene prevented the desired traffic lights to stand out. Without structural understanding of the scene, it is also hard for human to decide which light source is the subject signal control. Nevertheless, by tuning up the weight triplets to [1, 1, 4] and [1, 1, 2] (increasing the weight of average saturation) for the two images, respectively, all target traffic lights were picked up with reduced false reports on roadside street lights. This is implies an important rule of parameter tuning, that *during diming conditions where only light sources show up with high lightness, the major distinction between traffic lights and some of the other light sources is the color saturation, so the weight of average saturation should be increased to achieve better performance.* In dark unlit conditions like Figure 5-7, the major distraction could be introduced by traffic lights for the cross street and other highly reflective surfaces (road pavement

or signs). The rule of increasing the saturation weight also worked for this example. However, since the traffic lights for the cross street also have high saturation, only when the weight triplets were set to [1, 1, 2] were two out of three target red lights were detected. Further increasing the saturation weight to 3 gave higher conspicuity back to the green lights. In another dark unlit example in Figure 5-8, a weight triplet of [1, 1, 4] made the middle green light detectable.

However, increasing saturation weight is a general rule rather than a guaranteed remedy, because the scene complexity could violate the conspicuity assumption to an unknown extent. For example, in Figure 5-8 and Figure 5-9, the motion blur of the traffic lights breaks the assumption of the average or contrast areas (A1, A2, and A3). In Figure 5-10 (dawn/dusk), the color saturation of the target lights were actually underrepresented and an improved detection was achieved by increasing the weight of lightness contrast (i.e., [1, 2, 1]). In Figure 5-11, increasing the saturation weight detected only one of the two lights, but when combined with the adjustment of the lightness contrast weight (i.e., [1, 2, 4]) both lights were detected.

Figure 5-5 Dark lit example 1 (Frame ID: 00041-1803).

Figure 5-6 Dark lit example 2 (Frame ID: 00041-25891

Figure 5-7 Dark unlit example 1 (Frame ID: 00041-7545).

Figure 5-8 Dark unlit example 2 (Frame ID: 00041-22856).

Figure 5-9 Dawn/dusk example 1 (Frame ID: 00105-22284).

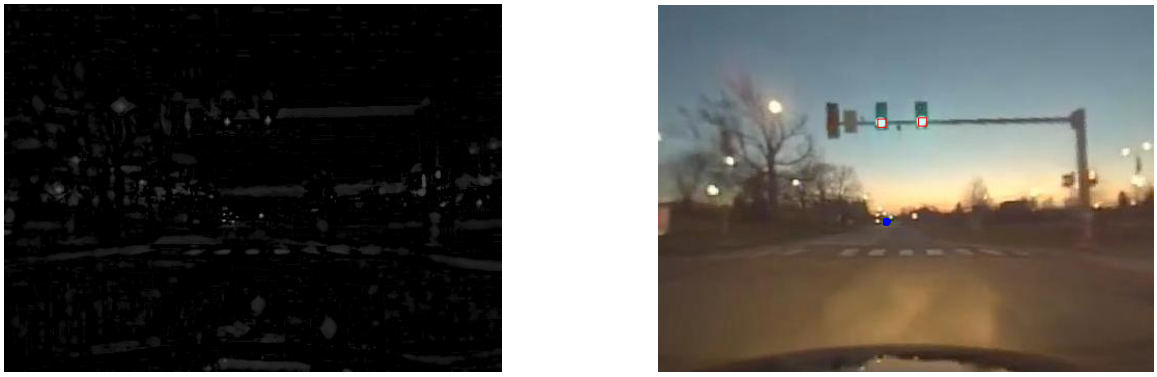


Figure 5-10 Dawn/dusk example 2 (Frame ID: 00137-5704).

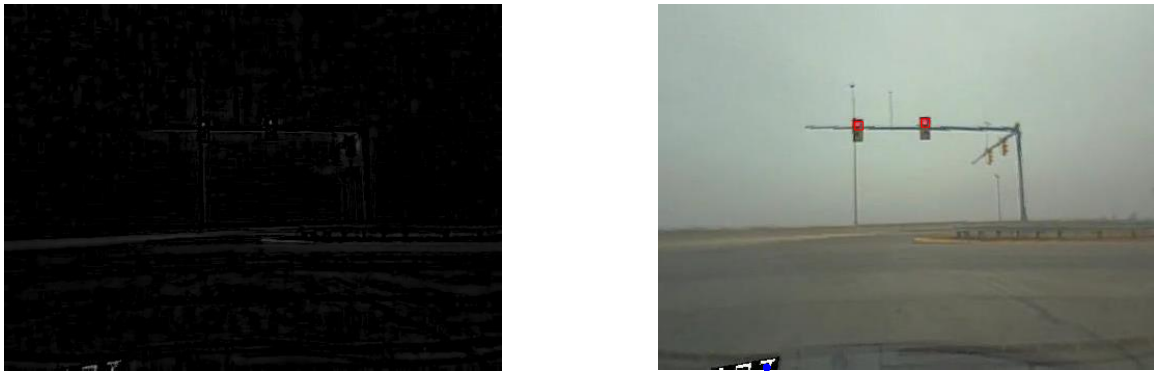


Figure 5-11 Cloudy example 1 (Frame ID: 00053-21007).

Lightness contrast showed more detection powers in sunny daytime. For example, as shown in Figure 5-12, the roadside objects and pavement markings has equivalent conspicuity as the target traffic light with equal weights between the three components of the conspicuity model. If the weight triplet is changed to [1, 3, 1] to increase the effect of lightness contrast, the middle traffic light will be detected.

Figure 5-12 Sunny example 1 (Frame ID: 00153-6185).

5.3 Classification Performance

Classification was tested on all true positive candidates from the base line detection result. In order to evaluate the sensitivity of the classifier, two sets of test were run with different training datasets. In the first test, the training data only contained 15 images randomly downloaded from the web. Among these images, five green lenses, six red lenses, and four yellow lenses were annotated and used to train histograms. The average radius of these lenses was about 18 pixels. In the second test, 26 sample frames from the HPV data were used, with ten green lenses, ten red lenses, and six yellow lenses. These frames were chosen from videos with cloudy condition so the color appearance was optimized. The average radius was about seven pixels. The classification results are given in Figure 5-13 and Figure 5-14 for these two tests, respectively. Similarly, the outputs are plotted in different combinations of lighting condition and target radius. The numbers of ground truth annotations for each signal color are given. The green triangle, red circle, and yellow triangle indicate the accuracy measures of green, red, and yellow, respectively classification results, respectively.

In Figure 5-13, a nearly ideal classification of green signals can be observed in all lighting conditions except sunny. Even in sunny days, the recall rates of green signals are generally high and no other signal colors were misclassified as green. In sunny days, both red and green signals can be confused with yellow signals, resulting in generally high FPR of yellow classifications. In cloudy days and dawn/dusk, confusion primarily happened between yellow signals and red signals. In dark lit condition, a major amount of yellow signals were misclassified as green signals, while red signals were well classified. This is intuitive because in dark lit conditions red signal lights are more distinguishable by color. In dark unlit conditions, the classification worked ideally for all

three signal colors. Overall, the classifier achieved 94.4% correct classifications for all tested candidates.

In Figure 5-14, when training was done using the samples in cloudy conditions from the HPV dataset itself, different performance changes happened in different lighting conditions. In sunny days, both green and red signals were easily misclassified as yellow signals. In cloudy conditions, red signals were easily misclassified as yellow signals. The overall performance improved in both dawn/dusk and dark lit conditions while the performance in dark unlit conditions almost remained the same.

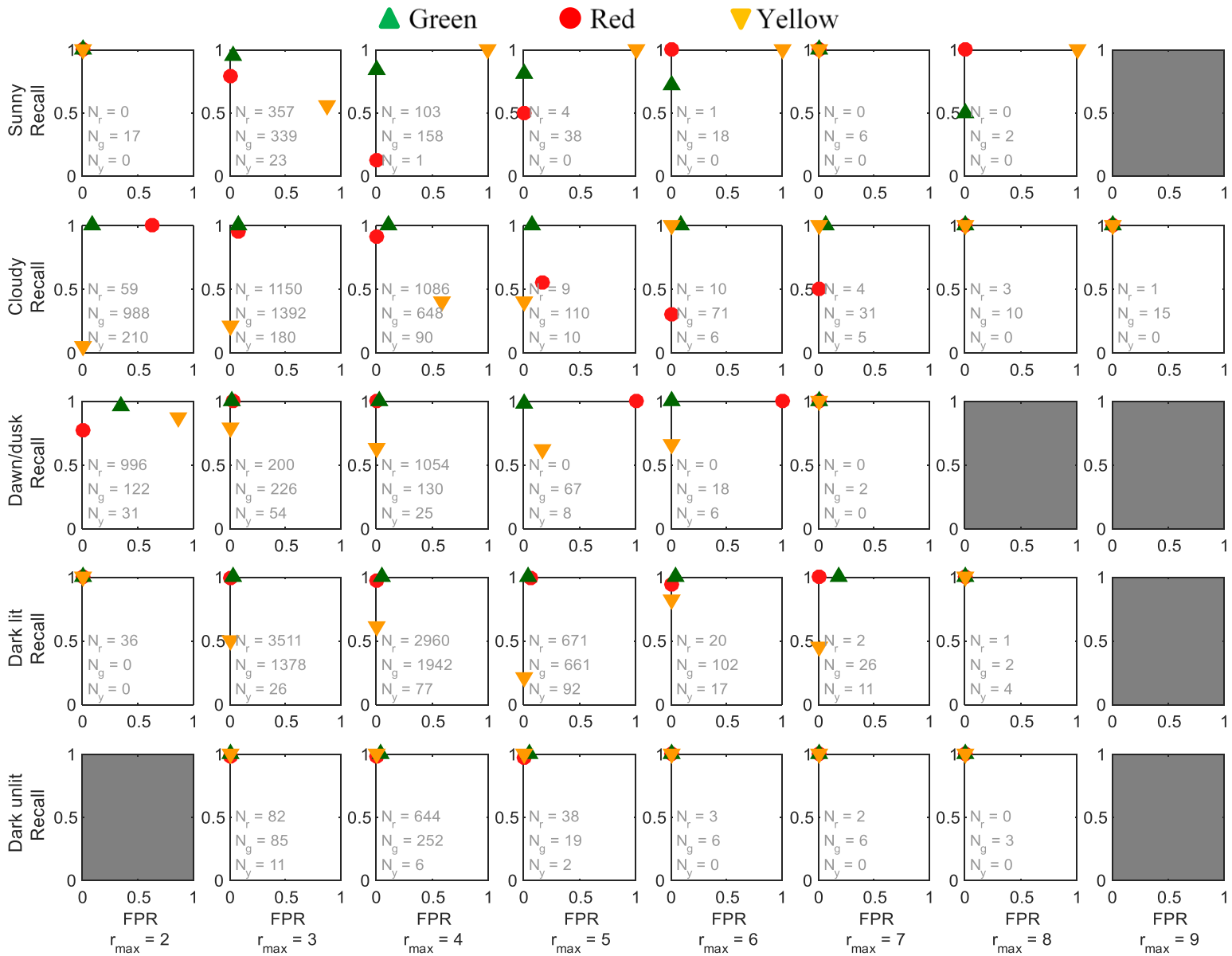


Figure 5-13 Classification using sample data from web images.

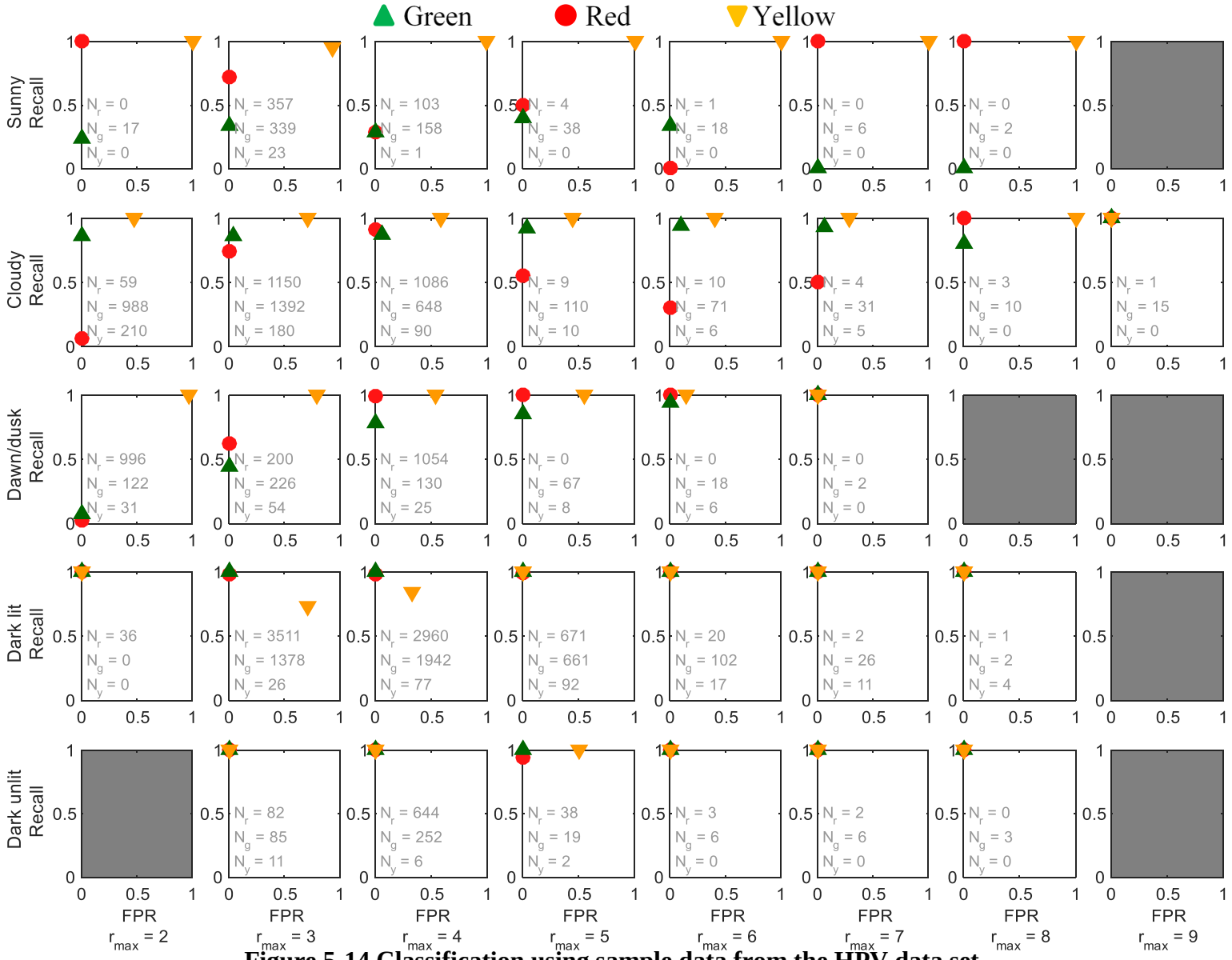


Figure 5-14 Classification using sample data from the HPV data set.

Assessment of the classification performance based on web training images were also conducted with respect to the classification confidence and the discriminative ratio. As shown in the left plot of Figure 5-15, the misclassification rate shows a general decreasing trend as the confidence increases, suggesting a positive correlation between the confidence and the classification accuracy. Though, the correlation is relatively weak, because the decision of classification is not based on the absolute value of the confidence, but based on the relative ranks of the confidences of three signal colors. Therefore, the misclassification rate is also plotted against the discriminative ratio (the right plot of Figure 8). Discriminative ratio is defined as 1 – (the minimum confidence / the maximum confidence). A small discriminative ratio implies that all signal colors have similar confidences and the chosen color only wins by a small amount. A large discriminative ratio implies that one signal color stands out. Decision made with a larger discriminative ratio is considered more reliable. In the right plot of Figure 8, this hypothesis is visually verified. When the discriminative ratio is above 0.7, the misclassification rate is constantly low.

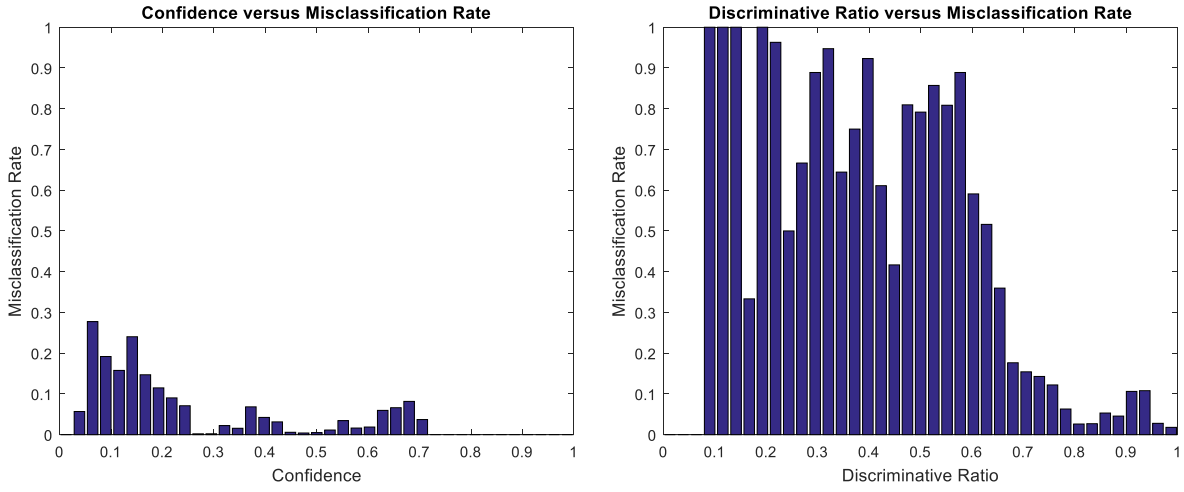


Figure 5-15 Assessments of the misclassification rate.

In summary, the classifier gave consistently good recall rates for green signals with different training datasets, except when the light condition was sunny. Sunny condition turns out to be a challenging condition for both detection and classification. This complies with the intuition that strong ambient light can significantly reduce the visibility and clarity of traffic signals even for human. In cloudy conditions, both training datasets led to confusion between red signals and yellow signals. Using sample data from the HPV dataset, the trained classifier showed better performance in dawn/dusk and dark lit conditions. In dark unlit conditions, the classifier performed ideally with either training dataset. Classification accuracy was found positively related to the absolute confidence and the relative confidence (i.e., the discriminative ratio above), with the latter showing more obvious trend. Therefore, the reliability of a classification can be effectively assessed by these two measures.

5.4 Spatiotemporal Framework Evaluation

Experiments on the spatiotemporal framework were conducted using the baseline detection setting and the sample from the HPV dataset was used for training the classifier. In addition, a general region of interests was set as the upper 60% of the frame. For the short range initialization, the upstream and downstream maximum trajectory distances were 10 m and 5 m, respectively. In the pruning pass, up to five tracks were selected. All 168 instances of signalized intersection traversal were correctly identified by the vicinity profile screening. A total number of 825 tracks were generated. Figure 5-16 shows all 825 resulting tracks aligned relative their clips' anchor frames. The blue portion is upstream of the anchor frame and the red portion is downstream of the anchor frame. The position is measured in cumulative trajectory distance (on the left) and in frame counts (equivalent to time duration). As shown, candidates could be tracked on average up to

around 500 feet or about 300 frames (about 20 seconds) upstream of the anchor frame. A few number of extremely long tracks were found false tracks.

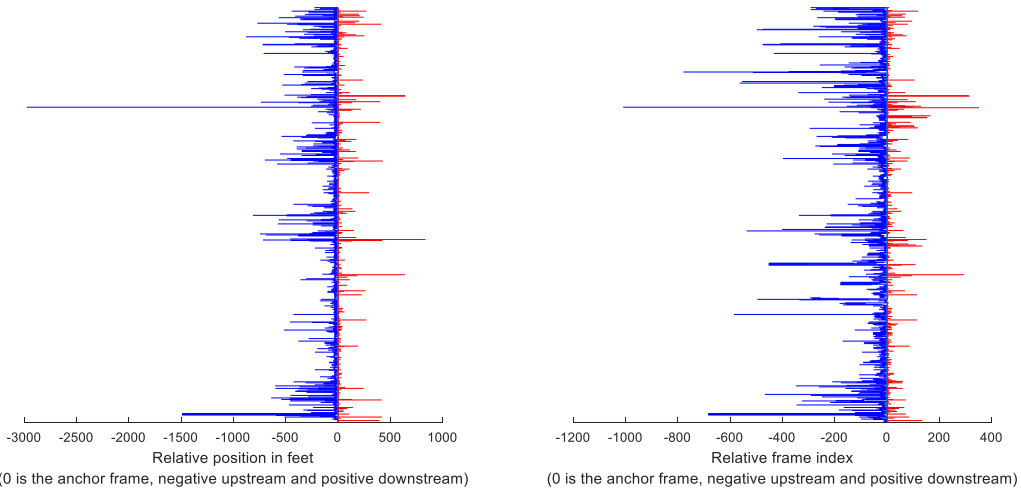


Figure 5-16 Tracks aligned relative to the anchor frame: left) position measured in feet and right) position measured in frame count (equivalent to time).

Detection results under the spatiotemporal framework are compared to those of the baseline detection in Figure 5-17. The comparison shows that in all lighting conditions except dark unlit, the spatiotemporal framework performed worse than the baseline detection. Such performance difference is counterintuitive at the first thought, because the spatiotemporal framework was designed to increase the reliability of detection. However, by carefully examining the theoretical behavior of the spatiotemporal framework, reasonable explanation can be derived. First of all, the track selection process in the pruning pass of the short range initialization stage could have excluded less stable tracks that in fact corresponded to true positive detections. Second, by selecting a track that was related to a non-traffic-signal, the program could have boosted additional false positives during the pruning and long range tracked recognition. For the sunny condition, which is challenging even for the baseline detection, both tests show similar results in general, especially in terms of recall rate. As the target size increased, the spatiotemporal framework could

have introduced more false positives along false tracks. In cloudy, dawn/dusk, and dark lit conditions, both the exclusion of true tracks and the inclusion of false tracks might play equivalent roles in degrading the detection performance, since both the recall rates and the false positive rates are lower with the spatiotemporal framework. In the dark unlit condition, the spatiotemporal framework outperformed the baseline detection. This reversion further confirms the above hypotheses because in dark unlit condition the actual traffic signals were the most conspicuous objects and were more likely to form stable tracks for long range recognition.

Color classification performance of the true positive detections under the spatiotemporal framework are illustrated in Figure 5-18. By comparing Figure 5-18 to Figure 5-14, similar patterns are observed. Because the spatiotemporal framework only provides hints for detection and tracking using classification results and does not alter the classification results, such similar performance patterns should be expected.

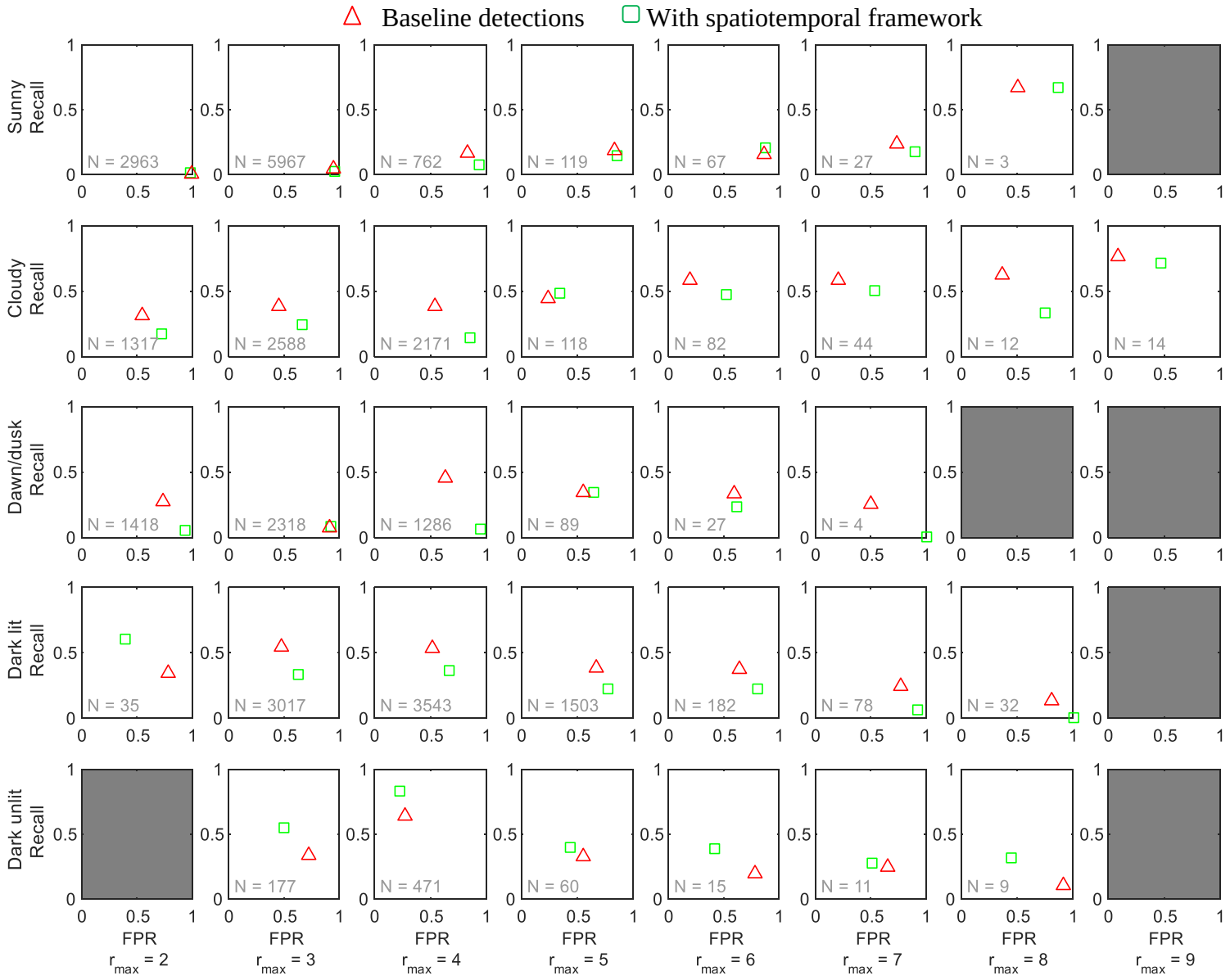


Figure 5-17 Comparison between base line detection and using spatiotemporal framework.

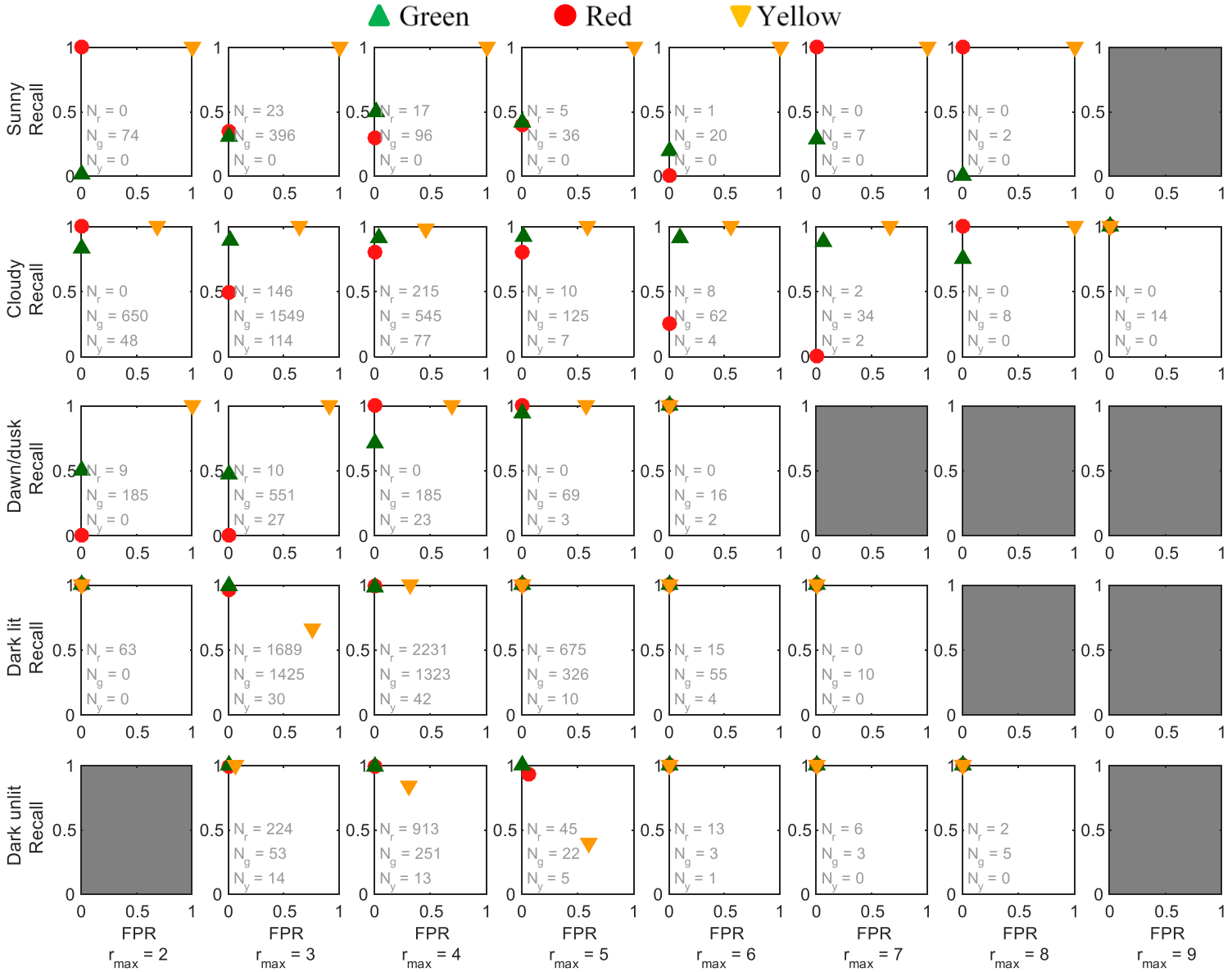


Figure 5-18 Color classification results of true positive detections under the spatiotemporal framework.

A natural question to ask is whether tracks with longer duration, in general, correspond to temporally more accurate traffic signal state. A theoretical answer is yes because the pruning stage and the long range tracked recognition stage are expected to fill in recognition gaps. In order to test this hypothesis, the temporal accuracy signal state of the tracks belonging to true detections is plotted against the duration of track (Figure 5-19). Temporal accuracy is defined as the ratio of the number of candidates that are correctly classified over the number of all candidates in a track. As shown in Figure 5-19, temporal accuracy converges towards 1 as the track length increases. Even with a short track length (below 150 frames), a descent number of tracks also show ideal temporal accuracy.

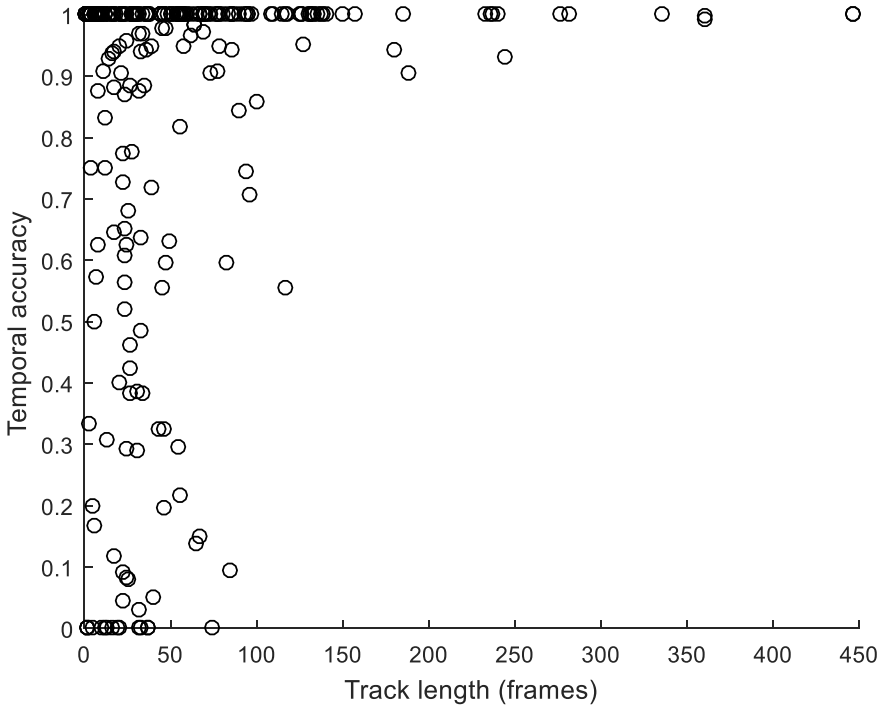


Figure 5-19 Relationship between temporal accuracy and track length.

5.5 Summary

In this chapter, experimental results are shown and analyzed. A baseline detection using default parameters yielded an initial assessment of the accuracy performance of the conspicuity based detector in different scenarios of lighting condition and target size. In general, the algorithm was able to adapt to various scenarios with at least one target correctly located in a majority of frames. Sunny condition is found to be the most challenging situation due to strong ambient light. Cloudy condition is among the most desirable scenario for detection and the detection accuracy increases as the target size increases. Reasonable performance was observed in dawn/dusk, dark lit, and dark unlit conditions, although the increase of target size did not introduce more detection benefits. In order to show the advantage of the proposed method, two other generic traffic signal detection algorithms were compared to. Results indicate that the proposed method consistently outperformed the other methods in all scenarios except sunny days. In sunny condition, all methods performed equivalently, further confirming the difficulty of traffic signal detection under strong sun lights. Visual inspections of a sample of poorly processed frames allow deeper insights into several misdetection issues. In general, properly tuning the weights of conspicuity components can effectively improve detection. Color saturation is recommended with a higher weight in dim environment and lightness contrast is more recommended for bright environment. There were issues that could not be addressed by the proposed algorithm, such as motion blur. These issues are considered limitations of the automatic TSR in general.

Tests on the classifier using different training datasets show that the histogram based approach is robust for green light classification. Sunny condition is still challenging for correct classification. Using a more representative training dataset could improve the performance in dawn/dusk and dark lit conditions. Dark unlit condition is the most desirable lighting environment

for classification with almost ideal classification accuracy. Yellow and red signals were easily confused with each other in cloudy condition. Absolute and relative confidences were both found to be reasonable predicates of the reliability of a classification, with the relative confidence being more positively related to the classification accuracy.

Experiments on the spatiotemporal framework reveal important insights into its theoretical behavior. Detection accuracy can be degraded when 1) the track of an actual traffic signal lacks stability and is excluded or when 2) the track of a non-signal gets selected due to its persistence in the detection history. The former situation could decrease the recall rate and the latter situation would increase the false positive rate. The spatiotemporal framework showed little effect on the classification results, which is expected because the framework does not alter any classification steps. Temporarily, tracks with longer durations showed more accurate temporal profile of traffic signal state.

CHAPTER 6

CONCLUSIONS AND DISCUSSION

A set of comprehensive TSR related algorithms are proposed in this thesis for road information extraction from massive video data source. The algorithms consist of two major subsets: 1) image based traffic signal detection and classification algorithms and 2) spatiotemporal information based preprocessing and coordination framework. In the development of these algorithms, minimal to no assumptions were made about the uniformity of the source cameras, the accessibility of camera exposure controls, the availability of camera-dependent sample data, the lighting condition of the road environment, or the target size in the image. Such openness of input sets a high requirement for these algorithms to be generic. Though, with the ubiquitous color cameras, the vision algorithms do assume RGB based color input. Additionally, when GPS readings of camera position are available, the spatiotemporal framework only requires a relaxed accuracy and density of the position data as long as linear interpolation can result in no more than 5 m of deviation from the actual position.

6.1 Detection and Classification Remarks

An innovative concept of conspicuity was developed to model the likelihood of a pixel being the center of a traffic signal lens. The concept is based on how traffic signal lights in the image appear distinctly to human eyes, more specifically their brightness as a result of both luminance and color saturation and the brightness contrast against their bounding boxes. According to this concept, conspicuity is modeled as a weighted geometric mean of three convolutional features: the average lightness of the lens area, the contrast of lightness between the lens area and the border area, and the maximum average saturation of the lens area and the annulus

area. The first two features are related to CIELab color space based lightness and the third related to HSV color coordinates based saturation. The convolutions take into consideration of both daytime and nighttime scenarios, especially with accommodation to the situation when light color diffuses to the halo surrounding of a traffic light in the dark. Also, multiple scale convolutions allow the final conspicuity value to adapt to any potential target size. Due to the use of geometric mean, the relative ratios of conspicuity among pixels are invariant to the choice of value range of any of its convolutional components. This is an important property in that it 1) gives a consistent shape of the resulting normalized conspicuity map, 2) allows each component to be calculated in any computational efficient numerical data types, and 3) purifies the control of weights as a control of each component's contribution rather than a mix with value scale adjustment.

An iterative localization algorithm was developed to locate the center and size of candidate traffic signal lens. The algorithm works on the normalized conspicuity map generated by the conspicuity model. It iteratively looks for peaks at different conspicuity levels. During the position search, a proposed DRMShift algorithm is used to allow a traditional peak finding mean shift algorithm to work in the case of dynamic radius.

Candidate traffic signal lens are classified using a proposed histogram similarity based color classifier. The classifier trains a 2D histogram of the "a" and "b" coordinates of the CIELab space for each traffic signal color based on sample training images. The same type of histogram is calculated for each candidate and compared to the trained histograms to give a soft classification of the candidate's signal color, i.e., with a score for the candidate to be any of the three possible signal colors. Although the requirement for training data seems to defy the design principle for generic algorithms, the histogram similarity based soft classification theoretically reduces the training data's dependency of camera. In other words, training data from one camera are expected

to work reasonably well on testing data from other cameras, because the algorithm is looking for the relative likelihoods of three possible colors rather than drawing a hard categorization using fixed thresholds.

Experiments were conducted on over 30,000 frames with various lighting conditions and target sizes with default detector parameters. For 50% or more frames in each combination of lighting condition and target size, the detection algorithm worked reasonably well in terms of correctly identifying at least one out of three to four active traffic signals presented at the same time. In a major portion of these detectable frames, the detected signals are over two and even up to three or four. False positives occurred at a considerable amount. Rather than denying the effectiveness of the detector, these false positives in fact reasonably reflect the theoretical behavior of the detector, which is based on the model of conspicuity. Most false positives are conspicuous objects in the scene, including street lights, vehicle tail lights, other non-target traffic signal lights, and even the target traffic signal lights' reflection on the camera-mounted vehicle's hood. These false positives are strong resemblance or even actual instances of traffic signals. Without other prior knowledge or sophisticated scene analyzing ability like a human has, the algorithm is reaching its limit in terms of finding the best candidates. In relation, because of these competitive false candidates, when only a specified number of top candidates are to be extracted, the true candidates may lose the competition and lead to reduced recall rates. A major portion of the non-detectable frames belong to such case.

In order to justify the detection performance, two other algorithms that were used in the literature as the state-of-the-art generic detectors were implemented, tested, and compared to. These algorithms possess similar conspicuity concept as the proposed detector, but they either underrepresent the contribution of color saturation or insufficiently reflect the lightness contrast.

As expected, the proposed algorithm outperformed the other two algorithms in all scenarios except one when one of the previous algorithms showed a better precision but an equal recall rate. The results suggest that the consideration of lightness, contrast, and saturation are a more comprehensive set of conspicuity features.

With the controllable weights of the conspicuity components, visual inspections to a sample set of non-detectable frames were conducted to investigate how weight changing would affect the detection performance. The inspected samples did reveal truly difficult frames that even human may find it imprecise to annotate the target traffic signals. There were other frames, where the targets were decent for detection but none was detected. For these frames, manual adjustment of the weights helped. Two general adjustment rules were derived. First, in dim environments, color saturation gives more distinction of traffic lights from other light source and the corresponding weight should be higher than the lightness based components to improve detection. Second, in daytime, the contrast plays an important role in differentiating the traffic signals from bright background and should be given more weight. However, these tuning rules cannot be too exact about the optimal ratios between weights for different scenarios, because the randomness of the scene can be too wild and the optimal ratio for one image may become suboptimal or even adverse for a similar image.

Sunny daytime turned out to be a challenging situation for all algorithms, because the strong ambient light can even prevent human observers to correctly locate the traffic signals. In cloudy days, a useful increasing trend of recall rate was observed as the target size increased, in other words, as the vehicle approached the signalized intersection. This observation provides a good evidence that detection is more reliable in short distance range, which complies with the assumption used in the design of the spatiotemporal framework.

Classification were tested on the true detections. Both camera-dependent and independent samples were used to train and compared. Regardless of what training samples were used, the classifier consistently showed ideal classification of green traffic signals, except in sunny days when the color information was almost undistinguishable. However, red and yellow traffic signals were easily confused, especially in cloudy or dawn/dusk conditions. Using different training data introduced subtle difference. Dark unlit condition was among the most preferable classification scenario in which all colors were ideally classified. Thinking in the application of behavior monitoring, such as red-light-running, the classifier is indeed conservative even though red and yellow signals can be confused. By reporting all recognized yellow lights indifferently with red lights, all actual red-light-runners should be captured, although with potential yellow-light-runners that may be of interests as well. Overall, the classifier could achieve a 94.4% accuracy.

6.2 Spatiotemporal Framework Remarks

A spatiotemporal framework is proposed to integrate the TSR into a production pipeline where input videos are lengthy and most of the time do not capture any traffic signals. In general, the framework uses position information to roughly extract instances of passing a signalized intersection. For each of the instances, a temporally coordinated TSR is performed to increase the efficiency and reliability of detection. There are several novel designs in this spatiotemporal framework compared to other related systems. A governing assumption leading to these design is that the data are processed offline in contrast to at real time. In an offline workflow, the position data of all frames are available all at once rather than being sequentially generated. Searching for the nearest signalized intersection can be performed at a sparser interval of frames and utilize spatial and motion constraints to fill in the interval gap without additional search. This idea was implemented as an extended kd-tree search that could speed up the vicinity calculation at a

controllable factor. Recalling the observation of potentially better detection rate at near distances, the offline workflow also allows the development of a temporal coordination that initiates the TSR on frames that are within a short range of trajectory distance from an anchor frame determined by the vicinity calculation. During this short range initialization, detected candidates are associated into tracks so their temporal persistence can be assessed and used to prune the detection results. With chosen and pruned tracks from the short range frames, long range frames, especially upstream frames, are being processed in a tracked manner. In other words, TSR is only performed in more restricted regions of interests in these frames based on track prediction. With the consideration of the presence of multiple traffic signals and the signal color change, the temporal coordination employed the Farneback dense optical flow algorithm to trace candidate footprints across frames in a robust way.

Experiments were conducted on the 21 30-minute long videos among which 168 instances of passing a signalized intersection were automatically identified and processed. Compared to TSR without the temporal coordination, lower recall rates and higher false positive rates were found. Such performance downgrade of using the spatiotemporal framework seems counterintuitive and discouraging at first, but it reveals important insights into the theoretical behavior of the temporal coordination process. The temporal coordination does not report all detection at the end, instead, only those detections associated with a top number of stable tracks are reported. When a track is less stable but contain true positive detections, these true positives are suppressed by the exclusion of the track. On the other hand, if a stable track with false positives is included and pruned, more false positives will be reported. A more reasonable way to assess the effect of temporal coordination is to see how temporal accuracy can be affected by the length (and hence stableness) of a track. By comparing the actual temporal profiles with the recognized temporal profiles of

target traffic signals, a clear trend was found that as the duration of the track increases, the average accuracy converges to 100%. Such increase of accuracy happened abruptly at around 150 frames. In terms of classification, the spatiotemporal framework introduced little impact because the algorithms used in the spatiotemporal framework do not alter the input to or the decision of the classifier.

6.3 Potential Applications

The proposed algorithms can find applications in existing and future projects. Currently, the Federal Highway Administration (FHWA) is constructing a data center to provide useful safety information that from massive SHRP2 naturalistic driving videos as mentioned in the introduction. Red-light-running events are of particular interests to the FHWA and capturing these events requires the traffic signal state information. The proposed algorithms work seamlessly with such system setting and are expected to efficiently generate instructive clues for red-light-running detection. The generic feature of the algorithms in fact gives them a wider adaptiveness to more video data sources. For example, people are becoming more prepared nowadays and many have bought a dash cam to monitor their driving environment during daily commute to collect evidence in case accidents happen. Even without a dash cam, drivers can also easily record the scene with their smartphones mounted behind the windshield. Imagine, when all these videos can be uploaded onto the internet cloud as the input to a peer-law-enforcing system, how important would it be for the system to have a robust functionality to automatically extract roadway information. With the proposed algorithms working with other computer vision technologies, such as vehicle detection, such system can automatically generate instances of potential traffic violations and identify the violator or witnesses based on the video data source. Because the proposed algorithms are generic,

they impose no requirement for the videos to be collected by the same type of camera, which would defy the concept of crowd sourcing.

6.4 Suggested Future Works

A most direct future work would be to test the algorithms on data other than the samples provided by the SHRP2 project. Data quality of the naturalistic driving data may undermine the potential of the proposed algorithms, especially by introducing complex scenarios that violate the assumptions of the proposed algorithms. By testing video data of higher quality, such as dynamic range to avoid overexposure, the capability of the proposed algorithms is possible to be fully revealed.

Although data quality presents a major challenge, the generic requirement of the algorithms has prevented them from employing advanced machine learning techniques, which rely on sizable training data that are preferably device or quality consistent with the testing data. Therefore, a natural next step is to integrate the conspicuity model into an ensemble model and/or machine learning framework (e.g., AdaBoost and convolutional neural network). Under such framework, more features can be added to the conspicuity model and weights of features can be trained. Recall that different lighting conditions have different optimal weights, the learning framework can train the weights based on information such as the whole frame lightness histogram, so the resulting conspicuity map optimally highlights the target traffic lights.

Another potential future works would be to incorporate latest advance in semantic segmentation to guide the detection.

REFERENCES

1. Yung, N., and A. Lai. An Effective Video Analysis Method for Detecting Red Light Runners. *IEEE Transactions on Vehicular Technology*, Vol. 50, No. 4, 2001, pp. 1074–1084.
2. Li, R., and Z. Tu. Automatic Recognition of Civil Infrastructure Objects in Mobile Mapping Imagery Using Markov Random Field. Presented at 2000 ISPRS Conference, Amsterdam, 2000.
3. Koukoumidis, E., M. Martonosi, and L. S. Peh. Leveraging Smartphone Cameras for Collaborative Road Advisories. *IEEE Transactions on Mobile Computing*, Vol. 11, 2012, pp. 707–723.
4. Fairfield, N., and C. Urmson. Traffic Light Mapping and Detection. Presented at IEEE International Conference on Robotics and Automation, Shanghai, 2011.
5. Levinson, J., J. Askeland, J. Dolson, and S. Thrun. Traffic Light Mapping, Localization, and State Detection for Autonomous Vehicles. Presented at IEEE International Conference on Robotics and Automation, Shanghai, 2011.
6. AASHTO. *A Policy on Geometric Design of Highways and Streets*. American Association of State Highway and Transportation Officials, Washington, DC, 2011.
7. FHWA. *Manual on Uniform Traffic Control Devices for Streets and Highways*. Federal Highway Administration, U.S. Department of Transportation, Washington, DC, 2009.
8. National Highway Traffic Safety Administration. *Traffic Safety Fact Sheets*. http://www-nrd.nhtsa.dot.gov/CATS/listpublications.aspx?Id=A&ShowBy=DocType. Accessed Nov. 2, 2016.
9. SWOV Institute for Road Safety Research. Naturalistic Driving: Observing Everyday Driving Behaviour. *SWOV Fact Sheets*. SWOV Institute for Road Safety Research, 2012, pp. 1–5.
10. Neale, V. L., S. G. Klauer, R. R. Knipling, T. A. Dingus, G. T. Holbrook, and A. Petersen. *The 100 Car Naturalistic Driving Study: Phase 1- Experimental Design*. Washington, D.C., 2002.
11. Antin, J. F., S. Lee, J. M. Hankey, and T. A. Dingus. *Design of the In-Vehicle Driving Behavior and Crash Risk Study*. Washington, D.C., 2011.
12. Dingus, T. A., S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. D. Sudweeks, M. A. Perez, J. M. Hankey, D. J. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, and R. R. Knipling. *The 100-Car Naturalistic Driving Study, Phase II - Results of the 100-Car Field Experiment*. Washington, D.C., 2006.
13. Dingus, T. A., J. M. Hankey, J. F. Antin, S. E. Lee, L. Eichelberger, K. E. Stulce, D. McGraw, M. Perez, and L. Stowe. *Naturalistic Driving Study: Technical Coordination and Quality Control*. Washington, D.C., 2015.
14. Hallmark, S. L., N. Oneyear, S. Tyner, B. Wang, C. Carney, and D. McGehee. *Analysis of Naturalistic Driving Study Data: Roadway Departures on Rural Two-Lane Curves*. Washington, D.C., 2015.
15. Hutton, J. M., K. M. Bauer, C. A. Fees, and A. Smiley. *Analysis of Naturalistic Driving Study Data: Offset Left-Turn Lanes*. Washington, D.C., 2015.
16. Victor, T., M. Dozza, J. Bärgman, C.-N. Boda, J. Engström, C. Flannagan, J. D. Lee, and G. Markkula. *Analysis of Naturalistic Driving Study Data: Safer Glances, Driver*

Inattention, and Crash Risk. Washington, D.C., 2015.
17. Backer-Grøndahl, A., R. Phillips, F. Sagberg, K. Touliou, and M. Gatscha. *Naturalistic Driving Observation: Topics and Applications of Previous and Current Naturalistic Studies. PROLOGUE Deliverable D1.1*. Oslo, Norway, 2009.
18. Neale, V. L., T. A. Dingus, S. G. Klauer, J. D. Sudweeks, and M. Goodman. An Overview of The 100-Car Naturalistic Driving Study and Findings. Presented at the 19th International Technical Conference on Enhanced Safety of Vehicles, Washington D.C., 2005.
19. Virginia Tech Transportation Institute. Event Detail Table. https://insight.shrp2nds.us/data/category/events#/table/38. Accessed Jan. 1, 2015.
20. Mu, G., Z. Xinyu, L. Deyi, Z. Tianlei, and A. Lifeng. Traffic Light Detection and Recognition for Autonomous Vehicles. *The Journal of China Universities of Posts and Telecommunications*, Vol. 22, No. 1, 2015, pp. 50–56.
21. Almagambetov, A., S. Velipasalar, and A. Baitassova. Mobile Standards-Based Traffic Light Detection in Assistive Devices for Individuals with Color-Vision Deficiency. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 3, 2015, pp. 1305 – 1320.
22. Jang, C., C. Kim, D. Kim, M. Lee, and M. Sunwoo. Multiple Exposure Images Based Traffic Light Recognition. Presented at the IEEE Intelligent Vehicles Symposium, Dearborn, 2014.
23. John, V., K. Yoneda, B. Qi, Z. Liu, and S. Mita. Traffic Light Recognition in Varying Illumination Using Deep Learning and Saliency Map. Presented at the IEEE International Conference on Intelligent Transportation Systems, Qingdao, 2014.
24. Trehard, G., E. Pollard, B. Bradai, and F. Nashashibi. Tracking Both Pose and Status of A Traffic Light via An Interacting Multiple Model Filter. Presented at the International Conference on Information Fusion, Salamanca, 2014.
25. Wang, Z., Z. Deng, and Z. Huang. Traffic Light Detection and Tracking Based on Euclidean Distance Transform and Local Contour Pattern. Presented at the Chinese Intelligent Automation Conference, Yangzhou, 2013.
26. Kim, H., Y. Shin, S. Kuk, J. Park, and H. Jung. Night-Time Traffic Light Detection Based on SVM with Geometric Moment Features. *International Scholarly and Scientific Research & Innovation*, Vol. 7, No. 4, 2013, pp. 454–457.
27. Cai, Z., M. Gu, and Y. Li. Real-time Arrow Traffic Light Recognition System for Intelligent Vehicle. Presented at the WorldComp, Athens, 2012.
28. Siogkas, G., E. Skodras, and E. Dermatas. Traffic Lights Detection in Adverse Conditions Using Color, Symmetry and Spatiotemporal Information. Presented at the International Conference on Computer Vision Theory and Applications, Rome, 2012.
29. Kim, H., J. H. Park, and H. Jung. Effective Traffic Lights Recognition Method for Real Time Driving Assistance System in the Daytime. *International Scholarly and Scientific Research & Innovation*, Vol. 5, No. 11, 2011, pp. 1425–1427.
30. Gong, J., Y. Jiang, G. Xiong, C. Guan, G. Tao, and H. Chen. The Recognition and Tracking of Traffic Lights Based on Color Segmentation and CAMSHIFT for Intelligent Vehicles. Presented at the IEEE Intelligent Vehicles Symposium, San Diego, 2010.
31. Yu, C., C. Huang, and Y. Lang. Traffic Light Detection During Day and Night Conditions by A Camera. Presented at the International Conference on Signal Processing Proceedings, Beijing, 2010.
32. Nienhüser, D., M. Drescher, and J. M. Zöllner. Visual State Estimation of Traffic Lights Using Hidden Markov Models. Presented at the IEEE Intelligent Transportation Systems,

Funchal, 2010.
33. Xu, C., T. Nai-Qiang, and L. Yan. Traffic Lights Recognition Algorithm Based on Lab Color Space and Template Match. *Journal of Computer Applications*, Vol. 30, No. 5, 2010, pp. 1251–1254.
34. de Charette, R., and F. Nashashibi. Traffic Light Recognition Using Image Processing Compared to Learning Processes. Presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, 2009.
35. de Charette, R., and F. Nashashibi. Real Time Visual Traffic Lights Recognition Based on Spot Light Detection and Adaptive Traffic Lights Templates. Presented at the IEEE Intelligent Vehicles Symposium, Xi'an, 2009.
36. Park, J.-H., and C. Jeong. Read-time Signal Light Detection. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 2, No. 2, 2009, pp. 1–10.
37. Shen, Y., U. Ozguner, K. Redmill, and L. Jilin. A Robust Video Based Traffic Light Detection Algorithm for Intelligent Vehicles. Presented at the IEEE Intelligent Vehicles Symposium, Xi'an, 2009.
38. Joo, S. K., Y. Kim, S. I. Cho, K. Choi, and K. Lee. Traffic Light Detection Using Rotated Principal Component Analysis for Video-Based Car Navigation System. *IEICE Transactions on Information and Systems*, Vol. E91-D, 2008, pp. 2884–2887.
39. Kim, Y., K. Kim, and X. Yang. Real Time Traffic Light Recognition System for Color Vision Deficiencies. Presented at the International Conference on Mechatronics and Automation, Harbin, 2007.
40. Hwang, T.-H., I.-H. Joo, and S.-I. Cho. Detection of Traffic Lights for Vision-Based Car Navigation System. *Lecture Notes in Computer Science*, No. 4319 LNCS, 2006, pp. 682–691.
41. Lindner, F., U. Kressel, and S. Kaelberer. Robust Recognition of Traffic Signals. Presented at the IEEE Intelligent Vehicles Symposium, Parma, 2004.
42. Mascetti, S., D. Ahmetovic, A. Gerino, C. Bernareggi, M. Busso, and A. Rizzi. Robust Traffic Lights Detection on Mobile Devices for Pedestrians with Visual Impairment. *Computer Vision and Image Understanding*, Vol. 13, No. 19, 2015, pp. 1–13.
43. Ying, J., C. Xiaomin, G. Pengfei, and X. Zhonglong. A New Traffic Light Detection and Recognition Algorithm for Electronic Travel Aid. Presented at the International Conference on Intelligent Control and Information Processing, Beijing, 2013.
44. Roters, J., X. Jiang, and K. Rothaus. Recognition of Traffic Lights in Live Video Streams on Mobile Devices. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 10, 2011, pp. 1497–1511.
45. Ivanchenko, V., J. Coughlan, and H. Shen. Real-Time Walk Light Detection with a Mobile Phone. *Lecture Notes in Computer Science*, Vol. 6180, 2010, pp. 229–234.
46. Angin, P., B. Bhargava, and S. Helal. A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation. Presented at the IEEE International Conference on Mobile Data Management, Kansas City, 2010.
47. Eddowes, D. M., and J. L. Krahe. Pedestrian Traffic Lights Recognition in A Scene Using A PDA. Presented at the International Conference on Visualization, Imaging, and Image Processing, Marbella, 2004.
48. Shioyama, T., H. Wu, N. Nakamura, and S. Kitawaki. Measurement of the Length of Pedestrian Crossings and Detection of Traffic Lights from Image Data. *Measurement*

Science and Technology, Vol. 13, No. 9, 2002, pp. 1450–1457.
49. Franke, U., D. Gavrila, S. Gorzig, and F. Lindner. Autonomous Driving Goes Downtown. *IEEE Intelligent systems*, Vol. 13, No. 6, 1998, pp. 40–48.
50. Chung, Y.-C., J.-M. Wang, and S.-W. Chen. A Vision-Based Traffic Light Detection System at Intersections. *Journal of Taiwan Normal University*, Vol. 47, No. 1, 2002, pp. 67–86.
51. Yelal, M. R., S. Sasi, G. R. Shaffer, and A. K. Kumar. Color-based Signal Light Tracking in Real-Time Video. Presented at the International Conference on Video and Signal Based Surveillance, Sydney, 2006.
52. Diaz-Cabrera, M., P. Cerri, and P. Medici. Robust Real-Time Traffic Light Detection and Distance Estimation Using A Single Camera. *Expert Systems with Applications*, Vol. 42, No. 8, 2015, pp. 3911–3923.
53. Cai, Z., Y. Li, and M. Gu. Real-Time Recognition System of Traffic Light in Urban Environment. Presented at the IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, 2012.
54. Diaz-Cabrera, M., P. Cerri, G. Pirlo, M. A. Ferrer, and D. Impedovo. A Survey on Traffic Light Detection. *New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops*, Vol. 9281, No. SEPTEMBER 2015, 2015.
55. Wang, C., T. Jin, M. Yang, and B. Wang. Robust and Real-Time Traffic Lights Recognition in Complex Urban Environments. *International Journal of Computational Intelligence Systems*, Vol. 4, No. 6, 2013, pp. 37–41.
56. Chiang, C., M. Ho, H. Liao, A. Pratama, and W.-C. Syu. Detecting and Recognizing Traffic Lights by Genetic Approximate Ellipse Detection and Spatial Texture Layouts. *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 12, 2011, pp. 6919–6934.
57. Lu, K.-H., C.-M. Wang, and S.-Y. Chen. Traffic Light Recognition. *Journal of the Chinese Institute of Engineers*, Vol. 31, No. 6, 2008, pp. 1069–1075.
58. Omachi, M., and S. Omachi. Detection of Traffic Light Using Structural Information. Presented at the IEEE 10th International Conference on Signal Processing, Beijing, 2010.
59. Gomez, A. E., F. A. R. Alencar, P. V. Prado, F. S. Osorio, and D. F. Wolf. Traffic Lights Detection and State Estimation Using Hidden Markov Models. Presented at the EEE Intelligent Vehicles Symposium Proceedings, Dearbon, 2014.
60. Sooksatra, S., and T. Kondo. Red Traffic Light Detection Using Fast Radial Symmetry Transform. Presented at the International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Nakhon Ratchasima, 2014.
61. Long, J., E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. Presented at *CVPR*, Boston, 2015.
62. Corney, D., J. D. Haynes, G. Rees, and R. B. Lotto. The Brightness of Colour. *PLoS ONE*, Vol. 4, No. 3, 2009.
63. Cormen, T., C. Leiserson, R. Rivest, and S. Clifford. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 2009.
64. Farneb, G. Two-Frame Motion Estimation Based on Polynomial Expansion. *Lecture Notes in Computer Science*, Vol. 2749, No. 1, 2003, pp. 363–370.
65. McLaughlin, S. B., and J. M. Hankey. *Naturalistic Driving Study: Linking the Study Data to the Roadway Information Database*. Washington, D.C., 2015.

66. U.S. Department of Transportation, and Federal Aviation Administration. Specification for the Wide Area Augmentation System (WAAS). http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/library/documents/media/waas/2892bC2a.pdf.
67. Federal Aviation Administration, National Satellite Test Bed, and Atlantic City International Airport. *Wide-Area Augmentation System Performance Analysis Report*. Atlantic City, NJ, 2006.
68. European Commission. *EGNOS Safety of Life Service Definition Document*. 2011.
69. OpenStreetMap. Tag:highway=traffic_signals. http://wiki.openstreetmap.org/wiki/Tag:highway%3Dtraffic_signals. Accessed Jan. 1, 2016.
70. Loy, G., and A. Zelinsky. A Fast Radial Symmetry Transform for Detecting Points of Interest. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, 2003, pp. 959–973.

APPENDIX A

RATIO INVARIANCE OF GEOMETRIC MEAN TO VARIABLES' SCALES

Given a weighted geometric mean,

$$y_i = \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} \left(s_k \times x_{i,k}\right)^{w_k}}$$

Proof that $\frac{y_i}{y_j}$ remains constant with respect to changes in any of the non-negative scales, s_k.

The Proof:

Assume that s_k changes to s'_k and y_i and y_j become y'_i and y'_j. The proof is equivalent to proofing $\frac{y'_i}{y'_j} = \frac{y_i}{y_j}$, as detailed below:

$$\frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} \left(s'_k \times x_{i,k}\right)^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} \left(s'_k \times x_{j,k}\right)^{w_k}}}$$

$$\rightarrow \frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{i,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {s'_k}^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{j,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {s'_k}^{w_k}}}$$

$$\rightarrow \frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{i,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {s'_k}^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{j,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {s'_k}^{w_k}}}$$

$$\rightarrow \frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{i,k}}^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{j,k}}^{w_k}}}$$

$$\rightarrow \frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{i,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} s_k^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} {x_{j,k}}^{w_k}} \times \sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} s_k^{w_k}}}$$

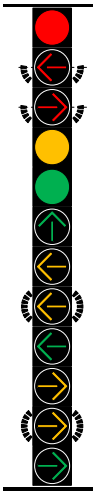
$$\rightarrow \frac{y'_i}{y'_j} = \frac{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} \left(s_k \times x_{i,k}\right)^{w_k}}}{\sqrt[\Sigma_{k=1}^{N} w_k]{\prod_{k=1}^{N} \left(s_k \times x_{j,k}\right)^{w_k}}} = \frac{y_i}{y_j}$$

APPENDIX B

A BRIEF SUMMARY OF TRAFFIC SIGNAL FACE DESIGN

Traffic signals, according to the modern standards, are illuminated lights with dedicated colors and shapes for commanding particular movements *(7)*. The term signal section is used to refer to the region of a single traffic signal lens (illuminated or not) and its bounding box. Multiple signal sections are arranged together to form a signal face to control one or more traffic movements from a single approach. More than one signal face can be given to one approach, corresponding to different movements. Generally, a signal face can only contain three, four, or five signal sections, except when a one-section signal face is used to give constant green to a conflict-free movement. Typical signal face arrangements are horizontal or vertical in a line; the relative positions of signal sections shall follow the orders shown in Table B-1 *(7)*. In a vertical arrangement, optionally, tow signal sections with the same color of indications can be placed horizontally to each other and form a cluster. Signal faces placed over the traffic lanes are called overhead signal faces. Signal faces on the roadside are called pole-mounted signal faces.

Table B-1 Orders of Signal Sections

Vertical: Top to Bottom	Horizontal: Left to Right
	
• CIRCULAR RED • Steady and/or flashing left-turn RED ARROW • Steady and/or flashing right-turn RED ARROW • CIRCULAR YELLOW • CIRCULAR GREEN • Straight-thru GREEN ARROW • Steady left-turn YELLOW ARROW • Flashing left-turn YELLOW ARROW • Left-turn GREEN ARROW • Steady right-turn YELLOW ARROW • Flashing right-turn YELLOW ARROW • Right-turn GREEN ARROW	• CIRCULAR RED • Steady and/or flashing left-turn RED ARROW • Steady and/or flashing right-turn RED ARROW • CIRCULAR YELLOW • Steady left-turn YELLOW ARROW • Flashing left-turn YELLOW ARROW • Left-turn GREEN ARROW • CIRCULAR GREEN • Straight-thru GREEN ARROW • Steady right-turn YELLOW ARROW • Flashing right-turn YELLOW ARROW • Right-turn GREEN ARROW

APPENDIX C

DETAILS OF TRAVERSED SIGNALIZED INTERSECTIONS

For the purpose of checking the accuracy of the OSM data as well as for the analysis of experimental results, detailed information of the signalized intersections being traversed in the HPV dataset were manually extracted using the satellite view of Google Maps™. Figure C-1 gives an illustration of the terminologies. For each approach, the lanes are numbered from left to right, starting at 1 and increasing by 1. The same numbering scheme also applies to the signal heads. The number of lanes does not necessarily equal the number of signal heads in the same approach, although they are commonly equivalent. Although overhead signals and roadside pole-mounted signals are both common deployments, of the 7 traversed intersections, only overhead traffic signals are used as the primary signals.

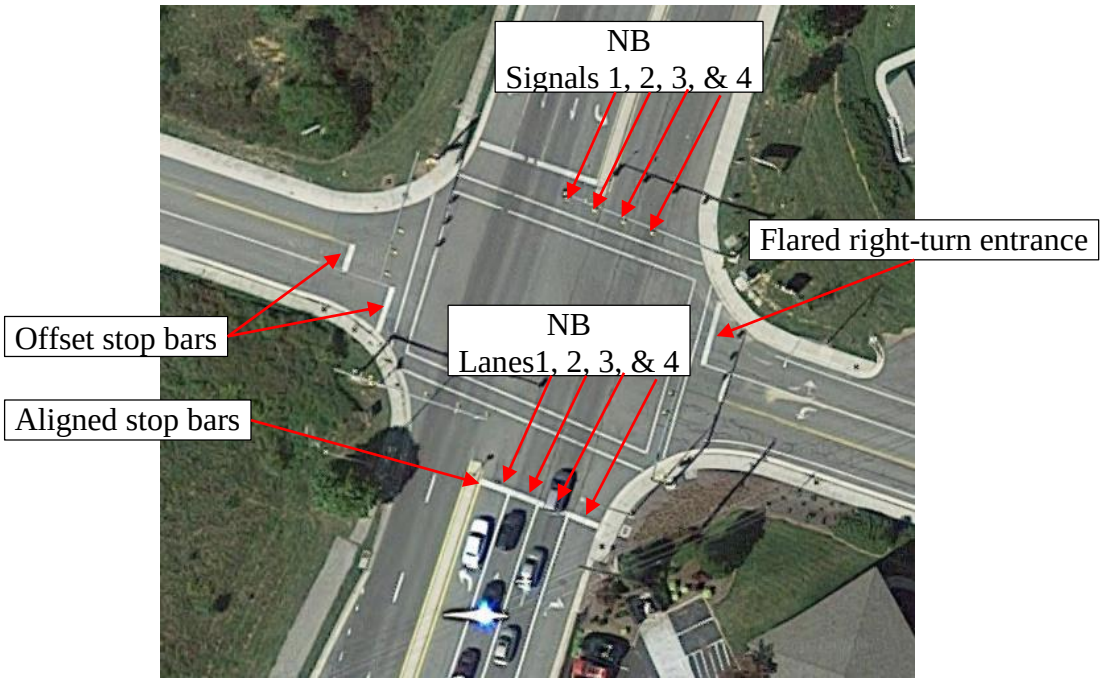


Figure C-1 Illustration of key point extraction terminologies.

For each intersection approach, the positions (in latitude and longitude) of stop bar key points and signal heads were extracted (Table C-2 and Table C-3). In addition, for each intersection, the approximate intersection center was also located as the intersection between highway centerlines (Table C-3). In Table C-2 and Table C-3, latitude and longitude coordinates are arranged in two rows. Note, stop bars are considered on a per-lane basis and the key points of each is the left end point and the right end point. Therefore, for a four-lane approach, the total number of stop bar key points is eight. In Table C-2, the left and right end points are to the left and right of the word "to", respectively. When stop bars are longitudinally aligned, some key points overlap with each other, for example, the right end point of lane 1 and the left end point of lane 2 of the NB approach of S Main Street @ Professional Park Drive. There are also cases when stop bars are offset by lane group (see Figure C-1). A stop bar normally lies within the width of its lane. An exception is when a flare right turn entry is used to provide a larger turning radius. In this case, the stop bar extends to the curb of the corner (see Figure C-1). Because of this, when calculating distance from an upstream point to the stop bar, the line between the upstream point and the midpoint of the stop bar is not a good reference. Instead, one should find the perpendicular line to the stop bar that goes through the upstream point. The distance between the upstream point and the intersection between the perpendicular line and the stop bar is the desired upstream distance.

Approaches traversed in the HPV dataset are highlighted in blue in both Table C-2 and Table C-3. The order in which these approaches were traversed in each video is identical to their row order in these two tables, except for intersection number 5, the WB approach was traversed before the EB approach.

Table C-2 Stop Bar Key Points

Intersection		Stop Bars											
ID and Name	Dir	Lane 1			Lane 2			Lane 3			Lane 4		
[1] S Main Street @ Professional Park Drive	NB	37.197575 -80.401362	to	37.197561 -80.401332	37.197561 -80.401332	to	37.197551 -80.401293	37.197551 -80.401293	to	37.197538 -80.401254	37.197538 -80.401254	to	37.197526 -80.401211
	EB	37.197820 -80.401539	to	37.197796 -80.401553	37.197776 -80.401487	to	37.197730 -80.401509						
	SB	37.197886 -80.401212	to	37.197898 -80.401261	37.197898 -80.401261	to	37.197908 -80.401292	37.197908 -80.401292	to	37.197920 -80.401331	37.197920 -80.401331	to	37.197932 -80.401374
	WB	37.197661 -80.401053	to	37.197687 -80.401042	37.209227 -80.399182	to	37.209233 -80.399139						
[2] S Main Street @ Hubbard/Ellett Road	NB	37.209154 -80.399249	to	37.209157 -80.399214	37.209157 -80.399214	to	37.209168 -80.399171	37.209227 -80.399182	to	37.209233 -80.399139			
	EB	37.209372 -80.399520	to	37.209340 -80.399515	37.209348 -80.399476	to	37.209311 -80.399469						
	SB	37.209534 -80.399306	to	37.209528 -80.399346	37.209510 -80.399344	to	37.209505 -80.399380	37.209505 -80.399380	to	37.209498 -80.399428			
	WB	37.209418 -80.399056	to	37.209446 -80.399053	37.209444 -80.399123	to	37.209485 -80.399125						
[3] Southgate Drive @ Beamer Way/Research Center Drive	NB	37.217238 -80.419116	to	37.217246 -80.419076	37.217262 -80.419079	to	37.217269 -80.419034						
	EB	37.217351 -80.419320	to	37.217317 -80.419308	37.217317 -80.419308	to	37.217277 -80.419295						
	SB	37.217522 -80.419197	to	37.217510 -80.419241	37.217496 -80.419237	to	37.217486 -80.419283						
	WB	37.217383 -80.418985	to	37.217425 -80.418988	37.217425 -80.418988	to	37.217448 -80.419007						
[4] Southgate Drive @ Duck Pond/Dairy Drive	NB	37.216226 -80.423631	to	37.216271 -80.423500									
	EB	37.216242 -80.423893	to	37.216215 -80.423879	37.216233 -80.423827	to	37.216206 -80.423811						
	SB	37.216428 -80.423741	to	37.216418 -80.423778	37.216390 -80.423766	to	37.216379 -80.423805						
	WB	37.216364 -80.423440	to	37.216400 -80.423458	37.216387 -80.423509	to	37.216422 -80.423525						
[5] Southgate Drive @ Huckleberry Trail	NB	37.213063 -80.431868	to	37.213106 -80.431805	37.213106 -80.431805	to	37.213140 -80.431695						
	SB	37.213238 -80.432174	to	37.213209 -80.432207	37.213209 -80.432207	to	37.213192 -80.432235	37.213192 -80.432235	to	37.213170 -80.432263			
	WB	37.213252 -80.431869	to	37.213282 -80.431888	37.213282 -80.431888	to	37.213303 -80.431901	37.213292 -80.431937	to	37.213338 -80.431969			
[6] US 460 5B Exit Ramp @ S Main Street	NB	37.191481 -80.403927	to	37.191467 -80.403895	37.191467 -80.403895	to	37.191453 -80.403859						
	EB	37.191791 -80.404010	to	37.191726 -80.404045									
	SB	37.191845 -80.403807	to	37.191856 -80.403842	37.191856 -80.403842	to	37.191870 -80.403876						
[7] S Main Street @ Industrial Park Road	NB	37.193838 -80.402929	to	37.193829 -80.402896	37.193829 -80.402896	to	37.193816 -80.402857	37.193816 -80.402857	to	37.193803 -80.402815	37.193822 -80.402806	to	37.193802 -80.402758
	EB	37.194102 -80.402995	to	37.194080 -80.403016	37.194068 -80.403000	to	37.194043 -80.403026						
	SB	37.194281 -80.402714	to	37.194289 -80.402751	37.194289 -80.402751	to	37.194299 -80.402789	37.194299 -80.402789	to	37.194312 -80.402828	37.194263 -80.402847	to	37.194274 -80.402889
	WB	37.193922 -80.402575	to	37.193955 -80.402562	37.193955 -80.402562	to	37.193987 -80.402551	37.194008 -80.402630	to	37.194069 -80.402602			

Table C-3 Signal Head Positions

Intersection ID and Name	Dir	Intersection Center	Signals 1	2	3	4
[1] S Main Street @ Professional Park Drive	NB	37.197734 -80.401270	37.197871 -80.401252	37.197858 -80.401214	37.197848 -80.401176	37.197836 -80.401139
	EB		37.197687 -80.401108	37.197666 -80.401112	37.197645 -80.401120	
	SB		37.197639 -80.401362	37.197650 -80.401403	37.197660 -80.401441	37.197672 -80.401481
	WB		37.197794 -80.401498	37.197816 -80.401492	37.197840 -80.401484	
[2] S Main Street @ Hubbard/Ellett Road	NB	37.209401 -80.399293	37.209527 -80.399323	37.209534 -80.399280	37.209540 -80.399246	
	EB		37.209386 -80.399103	37.209363 -80.399097		
	SB		37.209285 -80.399315	37.209280 -80.399343	37.209275 -80.399375	
	WB		37.209389 -80.399517	37.209406 -80.399519		
[3] Southgate Drive @ Beamer Way/Research Center Drive	NB	37.217366 -80.419160	37.217463 -80.419153	37.217471 -80.419126		
	EB		37.217391 -80.418986	37.217372 -80.418979	37.217355 -80.418973	
	SB		37.217260 -80.419096	37.217253 -80.419140		
	WB		37.217358 -80.419272	37.217389 -80.419284		
[4] Southgate Drive @ Duck Pond/Dairy Drive	NB	37.216298 -80.423681	37.216433 -80.423717	37.216440 -80.423691		
	EB		37.216349 -80.423536	37.216330 -80.423527	37.216311 -80.423519	
	SB		37.216179 -80.423620	37.216171 -80.423647	37.216165 -80.423669	
	WB		37.216258 -80.423831	37.216279 -80.423846	37.216294 -80.423856	
[5] Southgate Drive @ Huckleberry Trail	NB	37.213178 -80.432032	37.213303 -80.432098	37.213322 -80.432060	37.213333 -80.432022	
	SB		37.212976 -80.431924	37.212960 -80.431964	37.212944 -80.432000	
	WB		37.213090 -80.432304	37.213111 -80.432327	37.213132 -80.432346	
[6] US 460 5B Exit Ramp @ S Main Street	NB	37.191712 -80.403842	37.191615 -80.403863	37.191603 -80.403837		
	EB		37.191694 -80.403868	37.191676 -80.403883		
	SB		37.191568 -80.403998	37.191581 -80.404028		
[7] S Main Street @ Industrial Park Road	NB	37.194029 -80.402838	37.194132 -80.402799	37.194123 -80.402766	37.194111 -80.402726	37.194101 -80.402689
	EB		37.193865 -80.402649	37.193847 -80.402659	37.193828 -80.402673	
	SB		37.193943 -80.402909	37.193959 -80.402947	37.193973 -80.402979	37.193982 -80.403005
	WB		37.194103 -80.403093	37.194134 -80.403080	37.194166 -80.403069	