TEXT-BASED TRAFFIC SIGN DETECTION AND TRACKING IN VIDEO

TEXT-BASED TRAFFIC SIGN DETECTION AND TRACKING IN VIDEO

BY

JIANFENG HU, B.Eng., M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2021) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Text-based traffic sign detection and tracking in video
AUTHOR:	Jianfeng Hu M.Sc. (Multimedia Information Technology), City University of Hong Kong, Kowloon, HongKong SAR B.Eng. (Telecommunications Engineering with Management), Beijing University of Posts and Telecommunications, Beijing, China
SUPERVISOR:	Prof. T. Kirubarajan
NUMBER OF PAGES:	xix, 129

To my family

Abstract

In this thesis, the problem of multiple text-based traffic sign detection and tracking in video is explored, and the following specific problems are addressed: 1) localize the traffic sign in video by fusing the information from the front camera as well as other on-board data source such as rotation of the wheels and directional information of vehicles, 2) a CPU-based text-based traffic sign detector and directional parameter estimation for vehicles based on environmental conditions, 3) a text-based traffic signs detection and tracking framework for real-time application with a low cost data acquisition method.

As a crucial component of Advanced Driver Assistance Systems (ADAS), traffic sign detection and tracking play essential roles in the Automotive Traffic Sign Detection and Recognition (ATSDR) system. Based on their different shapes, colors, letters and symbols, traffic signs in traffic scenario can be roughly categorized to two groups, namely, graphics-based (symbol-based) traffic signs and text-based traffic signs. Graphics-based traffic signs are regulatory signs, warning signs and temporary conditions signs, and text-based traffic signs are information and direction signs. Compared to graphics-based traffic signs, only a few algorithms have focused on text-based traffic signs detection and tracking. Detecting and tracking text-based traffic signs is a challenging task, mainly due to larger variations within the category

and limited available dataset.

Starting with localizing text-based traffic sign in urban traffic scenarios, the kinematic states of vehicles and the spatial-temporal relationships between vehicles and traffic signs need to be modelled and estimated. To solve this problem, a kinematic automotive motion model is proposed. This kinematic model fuses information from the front camera as well as other on-board data source such as rotation of the wheels and directional information of vehicles. Based on the proposed kinematic model, a text-based traffic sign localization algorithm is developed. The experiential results on real world video data show that the proposed localization algorithm achieves good performance and significantly reduces the computational cost compared to previously proposed methodologies.

Next, the CPU-based text-based traffic sign detection method is studied. To relax the restriction of directional data acquisition in kinematic automotive motion model, a parameter estimation method is developed based on different environmental/weather conditions. Then, a search region definition approach for traffic signs detection is presented. This approach takes the advantages of spatial-temporal information of the previous frames and different kinematic vehicles motion models in video and largely reduces the massive and repeated detection for common Maximally Stable Extremal Regions (MSERs) detector. From the experiential results, the proposed approach achieves good performance in real-time applications.

Finally, a text-based traffic sign detection and tracking framework is proposed for video-based Traffic Sign Recognition (TSR) system. In the detection stage, a data-driven text-based traffic signs detector is trained with street view images, and a low cost data acquisition approach is presented. In the tracking stage, a multi-traffic

signs tracking algorithm is proposed based on kinematic automotive motion model. The framework is evaluated on both public Traffic Guide Panel dataset and our self-collected ETFLab Text-based Traffic Sign Video Dataset. The overall performance demonstrates the effectiveness of the proposed system, which can be better adapted to real-time applications.

Acknowledgements

I would like to take this opportunity to thank the following people, without whom I would not able to complete all my achievements.

First and foremost, I would like to acknowledge my deepest gratitude to my supervisor Prof. Thia Kirubarajan. Thank you for accepting me as a Ph.D. student, for providing me invaluable advice and insightful guidance in my research, for supporting me both financially and spiritually, and for the continuous encouragement throughout my Ph.D. journey.

I would like to express my gratitude to Prof. R. Tharmarasa for giving me generous advice and support in my research work. I would like to thank my supervisory committee members, Prof. I. Bruce and Prof. J. K. Zhang. I appreciate all your time spent on my supervisory meetings and assisted me with valuable comments and feedback on my research works.

Thanks to all my colleagues and friends in the Estimation, Tracking and Fusion research laboratory for their help and support throughout my Ph.D. career. I cannot forget to thank the administrative staff of the Electrical and Computer Engineering department, specially Ms. C. Gies and Ms. T. Coop, for the administrative support. I am also thankful to the funding from Electrical and Computer Engineering department and the International Excellence Award from the Graduate School of

Studies.

Last, but by no means least, I would like to express my special gratitude to my parents, my uncle, my aunt and my girlfriend for their unconditional love and consistent support through all these years.

Contents

List of Figures

List of Tables

Notations and Abbreviations

Notations

$A \leq B$	A is less than or equal to B
$A \geq B$	A is greater than or equal to B
$(A \mid A \in \mathbb{Z})$	A is an integer

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ADAS	Advanced Driver Assistance Systems
ATSDR	Automotive Traffic Sign Detection and Recognition
BTS Dataset	KUL Belgium Traffic Signs Dataset
BOVW	Bag Of Visual Word
CC	Connected Component

CE-MSER Contrast-Enhanced Maximally Stable Extremal Regions

CLAHE Contrast-Limited Adaptive Histogram Equalization

CLN Cascaded Localization Network

CNN Convolutional Neural Network

CPM Color Probability Model

CPU Central Processing Unit

ELM Extreme Learning Machine

ETFLab-TTSVD
ETFLab Text-based Traffic Sign Video Dataset

FFT Fast Fourier Transform

FOV Field Of View

FPS Frame Per Second

GB Gigabit

GMM Gaussian Mixture Model

GTSDB German Traffic Sign Detection Benchmark

GTSRB German Traffic Sign Recognition Benchmark

GPS Global Positioning System

GPU Graphics Processing Unit

HLS	Hue, Lightness, Saturation
HSI	Hue, Saturation, Intensity
ID	Identity
IoUT	Intersection over Union Threshold
ITS	Intelligent Transportation System
MSER	Maximally Stable Extremal Region
MTO	Ministry of Transportation of Ontario
OTM	Ontario Traffic Manual
PC	Personal Computer
RAM	Random Access Memory
RGB	Red, Green, Blue
ROI	Region Of Interest
SOC	System On a Chip
SSD	Single Shot Detector
STS Dataset	Swedish Traffic Signs Dataset
SVM	Support Vector Machine
SWT	Stroke Width Transformation
TLD	Tracking-Learning-Detection

TSR	Traffic Sign Recognition
TTSDCE	Text-based Traffic Sign Dataset in Chinese and English
YOLO	You Only Look Once
YUV	Luma and Chroma

Declaration of Academic Achievement

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

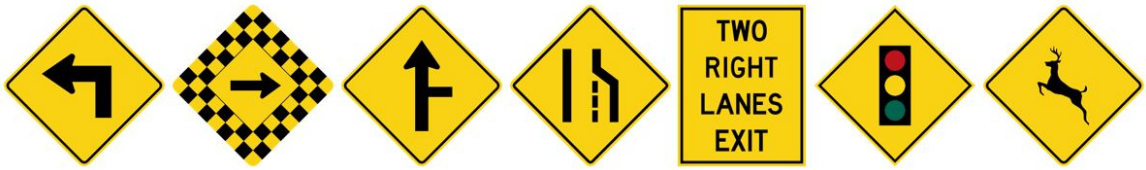
Chapter 1

Introduction

As important components of Intelligent Transportation System (ITS), Advanced Driver Assistance Systems (ADAS) are intelligent systems embedded inside the vehicle intended to get drivers better informed of traffic situations and aid safe driving. Among the techniques in the ADAS, Automotive Traffic Sign Detection and Recognition (ATSDR) system, which is also known as Traffic Sign Recognition (TSR) system, plays an essential role in autonomous driving [1] and mobile mapping [2].

Traffic signs provide important information to drivers and pedestrians, in warning about dangerous conditions and limitations of the road and in guiding to find the way. Traffic signs are designed by regulated laws in different countries. In Ontario, Canada, road signs conform to Ontario Traffic Manual (OTM) [3], which has been developed by the Ministry of Transportation of Ontario (MTO) since year of 2000. OTM covers different aspects of traffic control. For easier identification, different traffic signs use different shapes, colors, letters and symbols. As shown in Figure 1.1, there are mainly 4 kinds of traffic signs in Ontario, namely, regulatory signs, warning signs, temporary conditions signs and information and direction signs.

(a) Regulatory signs



(b) Warning signs

(c) Temporary condition signs

(d) Information and direction signs

Figure 1.1: Four major kinds of traffic signs in Ontario.

- Regulatory signs: These signs give a direction that must be obeyed. Beside stop sign (shaped in octagon, has white letters on red background) and yield sign (shaped in triangle with a red border on a white background), regulatory signs usually shaped in square or rectangular with black, white or coloured letters and a white or black background.

- Warning signs: These signs warn dangerous or unusual conditions ahead. Warning sign are usually diamond-shaped and have black letters or symbols on a

yellow background.

- Temporary condition signs: These signs warn of unusual temporary conditions such as construction zones, road work zone and lane closures. The shape of temporary condition signs are usually in diamond with black letters or symbols on an orange background.

- Information and direction signs (or Guide signs): These signs guide drivers and pedestrians about destinations and distances. Guide signs are usually rectangular with white letters on a green and blue background. Other background colours such as white and brown may guide to services, facilities and attractions.

Based on the shapes, colors, letters and symbols of traffic signs, in research studies, they are further grouped into two categories, namely, graphics-based (symbol-based) and text-based traffic signs [4]. Graphics-based traffic signs are regulatory signs, warning signs and temporary conditions signs. And text-based traffic signs are information and direction signs. Graphics-based traffic signs are then subcategorized into different classes such as stop signs, speed limit signs and construction signs. The main difference between graphics-based and text-based traffic signs is the within class variation. Sub-classes of graphics-based traffic signs often have fixed shape, uniform graphical appearance and distinct color. However, it is hard to find two identical text-based traffic signs on the road.

Figure 1.2 shows a complete TSR system containing 3 main stages, namely, traffic sign detection, traffic sign recognition and traffic sign tracking. Their detailed functionalities are:

- Traffic sign detection: The first stage for either an image-based or video-based

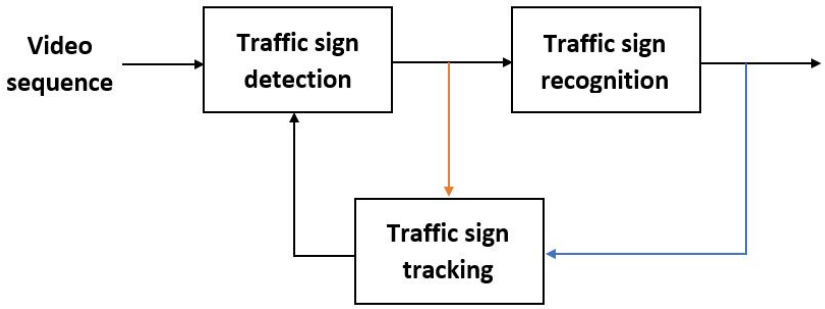


Figure 1.2: Flowchart of video-based TSR system. Tracking functionality can be applied either before (orange line) or after (blue line) the recognition stage.

TSR system. The outputs of traffic sign detection stage yield bounding box level results, which enclose the traffic sign region of interests (ROIs).

- Traffic sign recognition: Recognition stage is the post stage after detection stage, the traffic sign ROIs will be classified and the characters on the traffic signs will be recognized.

- Traffic sign tracking: This stage can be either applied before or after recognition stage, and the input of tracking functionality is the localization information or the recognition results of traffic sign ROIs, respectively. Traffic sign tracking is vital especially for video-based TSR system, the detected traffic signs are associated over the following frames.

1.1 Traffic Sign Detection

As a prerequisite task of traffic sign recognition, a timely and accurate traffic signs detection functionality is vital and has significant impacts on the performance of subsequent recognition functionality. Poor performance of detected text-based traffic

sign bounding box areas will result in missing characters in the post recognition stage.

Traffic sign detection is a challenging task mainly due to the following aspects [5], [6], [7]:

- Within- and across-category variation : Similar signs, different standardization of signs.
- Complexity of environment: Similar colored objects, uneven lighting, color fading, weather conditions, damaged or occluded signs and background complexity.
- Interference factors during the image acquisition: perspective distortion, defocused or motion-blur imagery, multi-orientation due to motion and partial or complete occlusion.
- Variations in text content (text-based traffic signs): Font, color, size, stroke width and multilingual environment.

In recent years, many existing algorithms have focused on graphics-based traffic sign detection and achieved promising results on many public datasets, such as the MASTIF Dataset [8], [9], [10], German Traffic Sign Recognition Benchmark (GTSRB) [11], German Traffic Sign Detection Benchmark (GTSDB) [12], KUL Belgium Traffic Signs Dataset (BTS Dataset) [2] and Swedish Traffic Signs Dataset (STS Dataset) [13]. However, only few research studies have focused on text-based traffic sign detection. As described in [4], this maybe partially caused by the difficulties of the task itself [14] and insufficient public datasets. Currently, the Traffic Guide Panel Dataset [15] is the only (partially) public text-based traffic sign dataset benchmark, which is an image dataset, not a video. Some examples of different datasets are shown in Figure 1.3.

(a)

(b)

(c)

(d)

(e)

(f)

Figure 1.3: Example of sample images for public graphics-based and text-based traffic sign datasets. MASTIF Dataset (a), GTSDB (b), BTS Dataset (c), STS Dataset (d), Traffic Sign Video Dataset ((e), (f))

A brief review of traffic sign detection methodologies on both categories of traffic sign are given in the following.

1.1.1 Graphics-based Traffic Sign Detection

The methods of graphics-based traffic sign detection are divided into four groups: color-based, shape-based, sliding window-based and ROIs-based methods. The color-based method usually first converts images to a new color space, such as normalized RGB [2], YUV [16], HSI [17], improved HLS [18], then relies on a designated color thresholding or color enhancement of the converted images to extract the ROIs. In [19], a Color Probability Model (CPM) is proposed to deal with the color information of graphics-based traffic signs by enhancing the red, blue and yellow colors of traffic signs and suppressing background colors.

Beside color, shape is another major distinct characteristic for traffic signs, which is relatively constant, such as, triangle, rectangle, square, octagon and circle. The shape-based methods commonly use Hough Transform [20], [21], [22] and its variants [23], [24], [25], radial symmetry voting [26] or corner detection to localize the traffic signs. Though these two kinds of methods are widely used for traffic sign detection, they are not robust to complex environment, such as uneven lighting and poor weather conditions.

Similar to the general object detection, sliding window-based methods use a region classifier such as AdaBoost [27], [28], [29] to determine whether the current window is a traffic sign. This kind of method is time-consuming and difficult to find the window size and aspect ratio. ROIs based method usually contains two steps: the traffic sign ROIs are extracted first, then filters out non-traffic sign regions with a classifier.

Compared to sliding window based method, ROIs based method is computationally faster. To effectively extract the traffic sign ROIs in the first step, methods such as Maximally Stable Extremal Regions (MSERs) detector [30] and template matching are used in [14], [31], [32]. The second step can be treated as a classification task, methods like Support Vector Machine (SVM) clasifier [33], Extreme Learning Machine (ELM) [34] and Convolutional Neural Network (CNN) [35] are commonly used in this step. For ROIs based method, it is important to balance the recall rate and the number of ROIs in the first step, since the missed traffic signs can not be recovered in the later filtering step and the computational time is related to the number of extracted ROIs.

1.1.2 Text-based Traffic Sign Detection

Compared to graphics-based traffic sign detection, research explicitly focused on the detection of text-based traffic signs is limited. Informed by road sign detection in color video sequence [36], a method to detect texts in text-based traffic signs from videos was proposed in [37]. The method first defines the geometric relationship between a moving car and traffic signs by assuming that the traffic sign is on a planar surface perpendicular to the horizontal ground and that the camera moves along its optical axis that is roughly horizontal. Then, the orientation of the plane is estimated using three or more points in two consecutive frames. A multiscale text detection algorithm is proposed on each candidate traffic panel area using edge detection, adaptive search, Gaussian Mixture Model (GMM) and geometric linear analysis to obtain the position of a text line and to track it with a feature-based tracking algorithm. In [38], the blue and white rectangular regions of interests (ROIs) were extracted using

a color-based segmentation algorithm and Fast Fourier Transform (FFT), then the four corner points of the rectangular regions were reoriented horizontally to align text characters. After analyzing the chrominance and luminance, an adaptive segmentation is carried out, and connected components labeling and position clustering are done for the arrangement of the different characters on the panel. In [39], the traffic signs that are located above ground and on the right side of the road are separated into two independent regions of interest. Then, the blue and white traffic panel regions were extracted for every single image based on color segmentation and Bag of Visual Words (BOVW) approach [40], and then classified the regions using classifiers, which was trained by support vector machines [33] or Naïve Bayes [41]. In [14], the pinhole camera model is used to restrict the search areas of the traffic sign in the detection stage. Then the potential text-based traffic sign candidates are detected in the defined search region based on the combination of MSERs detector and HSV color thresholding. Finally, the false positive regions are eliminated with the temporal and structural information.

Recently, with the intense power of deep learning, some text-based traffic sign detection methods have achieved promising results. In [15], a Cascaded Localization Network (CLN) is proposed to detect text-traffic sign candidates in the first stage, and then to locate text regions and eliminate the false alarm in the second stage. In [42], the traffic sign ROIs are extracted using MSERs algorithms in gray and normalized RGB channels, and then the regions are classified into different classes, including both graphics-based and text-based traffic signs by their proposed multi-task convolutional neural network. In [4], a text-based traffic signs detection algorithm was proposed

based on cascaded segmentation detection networks, which can achieve the state-of-the-art 0.9 precision with a computational speed of 0.15 seconds per frame.

1.1.3 Scene Text Detection

Many recent works on text-based traffic sign detection originated from scene text detection [43], [44], [45], [46], [47], [48], [49]. Some comprehensive surveys on scene text detection can be found at [7], [50]. The methods for scene text detection can be roughly divided into three groups, namely, sliding window-based method, connected component-based (CC-based) method, and deep learning-based method. The sliding window-based method uses multi-scale windows to move across the image to localize the high confidence text regions [51], [52], [53]. The main challenges of this method are training a powerful classifier by discriminative features and the heavy computation time caused by a large number of scanning window. Unlike the sliding window-based method, the connect component-based (CC-based) method is more efficient. CC-based method assumes that the characters in the image generate connected components. And the pixels in the same CCs have the same properties, such as stroke width, pixel intensity and grayscale level. The total number of the connected components is much less than the scanning windows so that it will take shorter computational time. Two representatives in this category are Stroke Width Transformation (SWT)-based method [46] and Maximally Stable Extremal Regions (MSERs)-based method [30]. SWT-based method utilizes the property that local characters have uniform stroke width to filter out false alarms [54]. The MSERs-based method uses the uniformity of the pixel intensity of the text stroke. The advantage of using the MSERs-based method is that it is fast and able to handle images even in low resolutions and contrast.

Many works [55], [56], [57] were inspired by SWT and MSERs methods. Recently, taking advantage of both deep learning algorithms and strong computational power of GPUs, originating from SSD [58] and TextBoxes [59], deep learning-based method treats the text detection problem as a regression problem by assuming text region to be a common category of objects such as cars and have achieved the most promising result [53], [59], [60], [61], [62].

1.2 Traffic Sign Tracking

Traffic sign tracking stage is vital especially for video-based TSR systems, which provide more valuable information than detecting signs in single image. The tracking functionality can be either applied before or after the recognition module. When recognition stage is applied before tracking stage, the recognition results are used for data association in later tracking stage. Otherwise, the detection results are used in the tracking stage. Same traffic sign in the video sequence will be assigned a unique id, and hence the number of repeated recognition for same traffic sign is reduced. Compared to traffic sign detection and recognition, only a few approaches have been studied in traffic sign tracking. In [63], a road sign tracking method is proposed by using continuous adaptive mean shift (cam-shift) method. In [64], a detected traffic sign is tracked using a simple motion model and temporal information propagation. Then, the results of the individual frame are fused for more robust detection. The works in [36], [65], [66] track the detected traffic signs by using a kalman filter [67], a credible result is achieved by deleting the detections that cannot be identified for consecutive frames. The kalman filter based tracker is used to reduce the computational load in detection stage and fuse the results from consecutive frames

to get better classification performance in [36], and improve a pre-trained off-line trained detector with an on-line updated detector in [68]. In [69], a Tracking-Learning-Detection (TLD) framework is adopted to track the recognized signs in real time to provide enough information for driver assistance function.

A good traffic sign tracking algorithm should have the following aspects. First, the tracker can track the detected traffic sign with only one detection (i.e. when the object is initially detected). Second, the running time of tracking algorithm should be fast enough to achieve real time application. Third, the algorithm should be able to handle miss detection in between frames and robust to occlusion. Forth, the tracker should be able to delete the traffic sign when it moves outside the field of view (FOV).

1.3 Theme and Objectives of Dissertation

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

The articles in the dissertation focuses on building a framework for multiple text-based traffic signs detection and tracking in video. The objectives of the thesis are the following:

- To propose a kinematic automotive motion model, which mathematically details the spatial relationship between traffic signs and different kinematic vehicles motion models. (Paper I)
- To fuse the information from the front camera as well as other on-board data

source such as rotation of the wheels and vehicles directional information or environmental/weather conditions. (Paper I, II, III)

- To propose an efficient CPU-based non-deep learning text-based traffic signs detector. (Paper II)
- To provide a fast and accurate data-driven text-based traffic signs detection approach and a low cost data acquisition method. (Paper III)
- To develop a text-based traffic signs detection and tracking framework for real-time application, which yields highly accurate detection results and at a significantly reduced computational cost. (Paper III)

1.4 Related Publications

- J. Hu, R. Tharmarasa, R. Lee, and T. Kirubarajan, "Localization of Text-based Traffic Signs Using a Kinematic Automotive Model", Submitted to *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- J. Hu, R. Tharmarasa, R. Lee, and T. Kirubarajan, "Text-based Traffic Sign Detection in Video using Kinematic Automotive Model", To be submitted to *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- J. Hu, R. Tharmarasa, R. Lee, and T. Kirubarajan, "A Framework for Text-based Traffic Sign Detection and Tracking in Video", To be submitted to *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Chapter 2

Localization of Text-based Traffic Signs Using a Kinematic Automotive Model

2.1 Abstract

As a crucial component of Advanced Driver Assistance Systems (ADAS), traffic sign detection and tracking play essential roles in the Traffic Sign Recognition (TSR) module. Many algorithms have proposed for the detection of graphics-based traffic signs, but only a few have focused on text-based traffic signs. Existing state-of-the-art algorithms, which are usually based on deep learning methods, have achieved good detection results with near real-time performance on high-end graphical processing units. However, these approaches are not yet ready for real consumer-level TSR systems mainly because of their high cost. In urban traffic scenarios, detecting text-based traffic signs are challenging due to the complex environment, where the

kinematic states of vehicles and the spatial relationships between vehicles and traffic signs need to be modelled and estimated. In this Chapter, we propose a kinematic automotive model, which details the spatial relationship between traffic signs and a moving vehicle. In addition, we propose a new text-based traffic sign localization algorithm, which can predict the position of traffic signs after detecting them in two frames and obviate the need for the computationally expensive detection process in every frame. The proposed algorithm yields an Intersection over Union Threshold (IoUT) of at least 0.8 and 0.5 in the following 10 and 20 frames, respectively. The computational time of our approach is only 0.012s per frame for a 1080p video in Python 3.7 on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. The prediction results from our localization algorithm can be used as detection results by applying machine learning-based detection on some key frames, which reduces the computational complexity even further. Furthermore, the proposed algorithm is robust to the changes in the external environment such as uneven lighting and occlusion.

2.2 Introduction

As essential components of intelligent transportation systems (ITS) [70], advanced driver assistance systems (ADAS) [71] are intended to aid safe driving and to increase general road safety. In ADAS, the traffic sign recognition (TSR) module is an important feature since traffic signs contain valuable information that can be used for guidance. Generally, traffic signs can be categorized into two groups [4], namely, graphics-based and text-based traffic signs. Graphic-based traffic signs usually provide environmental information, such as school or construction zone, speed limits [72],

icy or windy road signs, etc. Text-based traffic signs often provide destination details, which are crucial pieces of information when the GPS or mobile signal is poor.

A typical TSR system has three different functionalities, namely, traffic sign detection, tracking, and recognition [68], [69]. As the first step in a TSR module, the timely and accurate detection of traffic signs has a significant impact on the subsequent recognition and tracking functionalities. Traffic signs are not amenable for perfect detection due to many challenging interior and exterior conditions [7]. Compared with the interior conditions such as variations in character font, color, size and multilingual nature, exterior conditions such as complexity of the background, uneven lighting [21], perspective distortion, defocused or motion-blur imagery, multi-orientation [2] due to motion and partial or complete occlusion are more challenging [73]. Many algorithms for traffic sign detection have been proposed in the literature with promising results on many public datasets, such as the MASTIF dataset [8] and the German Traffic Sign Detection Benchmark (GTSDB) [12]. However, only a few works have focused on text-based traffic sign detection and tracking. Comprehensive surveys on text-based sign detection can be found at [7], [50]. Compared to the graphics-based traffic signs [29], [69], [74], [75], [76], text-based traffic signs often lack attributes such as color, shape, size and uniformity in visual appearance [39]. Thus, techniques that have been shown to work well on graphics-based traffic signs [19] may not be effective in text-based traffic sign detection and tracking. Most recent works on text-based traffic sign detection originated from scene text detection [43], [44], [45], [46], [47], [48], [49]. Among the many techniques proposed for scene text detection, machine/deep learning-based methods are the most promising [53], [59], [60], [61], [62].

The main drawback of machine learning-based methods is the computational cost. In real-time applications, there is often a trade-off between detection accuracy and processing time. In [4], a detection algorithm based on cascaded segmentation detection networks, which can achieve 0.9 Precision but requires 0.15 seconds per frame is proposed. The major limitations of this algorithm are the high computational cost and the inability to detect traffic signs in every frame of the video. Down-sampling of frames to improve detection speed is a possibility. Also, with traffic sign tracking in videos, there is a correlation between frames, which can be exploited to make sign detection more efficient. The kinematic status of moving cars and the spatial relationship between them and the traffic signs can be modelled and used for text-based traffic signs detection.

In [36], a kinematic visual model for recognizing circular, triangular and octagonal road signs such as stop signs is introduced. An approximate visual model is used to predict the size and the position of a sign in the video. In [37], the geometric relationship between a moving car and traffic signs is defined by assuming that the traffic sign is on a planar surface perpendicular to the horizontal ground and that the camera moves along its optical axis that is roughly horizontal. Then, the orientation of the plane is estimated using three or more points in two consecutive frames. A multiscale text detection algorithm is proposed on each candidate traffic panel area using edge detection, adaptive search, Gaussian Mixture Model (GMM) and geometric linear analysis to obtain the position of a text line and to track it with a feature-based tracking algorithm. In [39], the traffic signs that are located above ground and on the right-side of the road are separated into two independent regions of interest. In [14], the pinhole camera model is used to restrict the search areas of the traffic sign

in detection, and then text-based traffic sign candidates are detected and matched to those previous frame.

To sum up, accurate modelling of the kinematic relationship between moving vehicles and traffic signs can help address many challenging problems such as occlusion and motion blurriness and reduce false alarms in detection. Existing methods have four main issues: 1) Most methods were developed specifically for graphics-based traffic signs. Text-based traffic signs pose different challenges that have not been addressed in the literature. 2) Most methods focus on detecting traffic signs in single images. That is, they do not take advantage of the spatio-temporal correlation in videos. 3) Detection of traffic signs in every frame is computationally burdensome in real-time applications, and often unnecessary. 4) Existing TSR algorithms only use the camera sensor. The information from on-board data sources such as wheel rotation and motion direction are not used. These limitation provide the motivation for our work in this Chapter.

To better adapt to real-time application and to changes in the environments such as uneven lighting and occlusion, in this Chapter, we propose a text-based traffic sign localization algorithm. Our work proposes a more detailed and accurate kinematic automotive model with better approximation and fewer assumptions than in the previous work in [37]. The main contributions of our work are as follows: 1) We develop a traffic sign localization algorithm, which details the spatial relationship between traffic signs and different kinematic vehicles motion models. 2) The proposed kinematic model can be further integrated into the current TSR modules and traffic sign tracking algorithms. 3) The proposed algorithm uses the information from the front camera as well as other on-board data source such as rotation of the wheels

and vehicles directional information. 4) Our work yields highly accurate localization results and at a significantly reduced computational cost. From the experiential results on real video data, the prediction results of our algorithm have a Precision of 1.0 with Intersection over Union Threshold (IoUT) greater than 0.8 and 0.5 for the following 10 and 20 frames, respectively. The computational time of our approach is only 0.012s per frame for a 1080p video in Python 3.7 on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. Thus, the proposed algorithm can be used to substitute existing algorithms and integrated into tracking algorithms at a significantly reduced computational cost.

The structure of the Chapter is as follows. In Section 2.3, a brief review of the camera model is given first. Then, the details of the kinematic automotive model and the proposed traffic sign localization algorithm are presented. Experimental and evaluation results are presented and compared with the state-of-the-art text-based traffic sign detection algorithms in Section 2.4. Finally, conclusions are discussed in Section 2.5.

2.3 Traffic Sign Localization Problem Formulation

The computational cost of detecting and recognizing text-based traffic signs in every frame in real-time is too high. Unpredictable changes such as uneven light in the environment makes detection even harder and less accurate. In this section, we propose a traffic sign localization algorithm to predict text-based traffic signs in video sequence, which obviates the need for detection in every frame and reduces the computational cost.

To facilitate text-based traffic sign tracking, the kinematic states of a vehicle as

well as the spatial relationship between the vehicle state and a stationary traffic sign need to be modelled properly. A camera model is used to establish the mapping from a 3D scene to a 2D camera image plane.

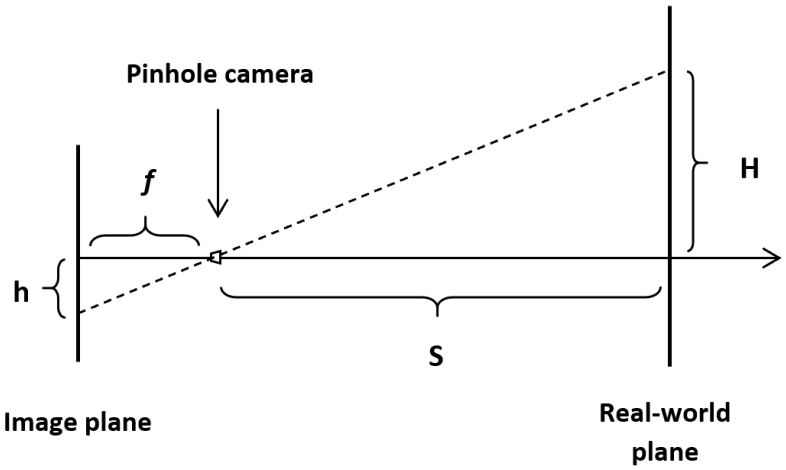


Figure 2.1: Projection of 3D information to 2D with a pinhole camera model.

A pinhole camera model [14], [37] is commonly used to mathematically formulate the projection relationship between the 3D world and the 2D camera image. As illustrated in Figure 2.1, denoted by H, h, f, S are the real-world object height, projected object height on the camera image, camera focal length and the distance between the camera and the object in real-world, respectively. Under the pinhole camera model [77], we have

$$\frac{h}{f} = \frac{H}{S} \tag{2.3.1}$$

where f is a predefined parameter of the camera, h is measured in pixels, and H and S are unknown values.

2.3.1 Proposed System

In a 2-dimensional coordinate system, the localization of traffic signs in images can be achieved by applying a deep learning-based traffic sign detection algorithm. However, detecting traffic signs in every frame is burdensome and sometimes unnecessary. By taking advantage of the spatio-temporal correlation in videos, the kinematic states of a vehicle and the spatial relationship between the vehicle state and a stationary traffic sign over time can be used in a 3-dimensional coordinate system.

The architecture of the proposed text-based traffic sign localization algorithm is shown in Figure 2.2. The detection begins at t_0, and stops when the same sign (target) is detected the second time at t_k $(k|k \in \mathbb{Z}, k \geq 1)$. We first consider the possibility of miss detection during the interval t_0 to t_k. The probability of detecting the second time in frame k depends on the nominal probability of detection P_d. Then,

$$Pr(k = N) = P_d \cdot (1 - P_d)^{N-1} \tag{2.3.2}$$

For example, if $P_d = 90\%$, the probability of the second detection when $k = 1$ is 90%, $k = 2$ is 9% and $k > 2$ is the remaining 1%. The proposed localization algorithm consists of two stages, namely, estimation and prediction. After detection, during the estimation stage, the improved kinematic automotive model of these two detected frames t_0 and t_k is used to estimate the distance between camera and the traffic sign. This is followed by the prediction stage to calculate the position of this traffic sign in the next frame at time t_{k+1}. From time t_{k+1}, the position information from the current and the previous frame is used, and the estimation and prediction steps are called recursively. In our proposed algorithm, we assume that we know the moving

distance (or speed) and the turning direction in every frame from on-board sensors. This assumption can be relaxed by estimating these values in real-time, which is the focus of our upcoming Chapters.

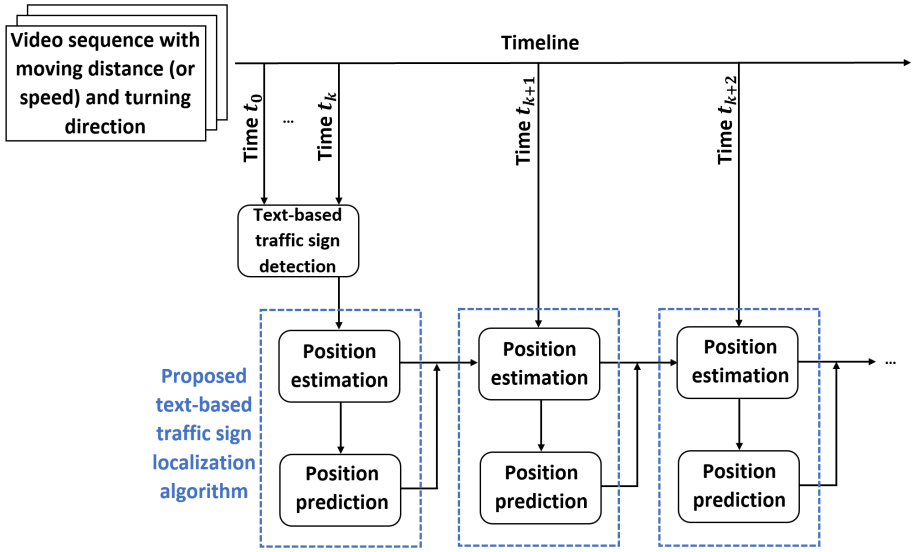


Figure 2.2: Architecture of the proposed text-based traffic sign localization algorithm.

2.3.2 Assumptions

With the knowledge of the traffic sign position from the two detected frames and the on-board sensor information of moving distance and direction in each frame, our goal is to fuse these information and predict the sign position in the subsequent frames. The kinematic state of a moving vehicle will be used to fuse this information. The motion of a forward-moving car can be described using different motion models [37], including straight line motion, lane changing and turning case. In all these cases, we make the following two assumptions:

1) The camera is moving alongside its optical axis, which is parallel to the ground

or road.

2) The text-based street sign lies on a planar surface that is perpendicular to the horizontal ground.

The first assumption is very close to the real-world scenario when we neglect the vibration of a moving camera. The second assumption is not limiting, as we demonstrate with real data. When moving close to an object in the 3D world, the projection area in the 2D video sequence is focused. Figure 2.3 shows the different positions and sizes of the same object in two consecutive frames t_0 and t_1. The simple (ideal) case is when the center of the object lies on the camera optical axis as shown in Figure 2.3 (a). In this situation, when the camera moves along its optical axis, the position of the object center in the image stays the same. However, this case is not practical in real traffic. Figure 2.3 (b) illustrates a more common scenario where the text-based road sign changes between the two frames in a forward-moving scenario.

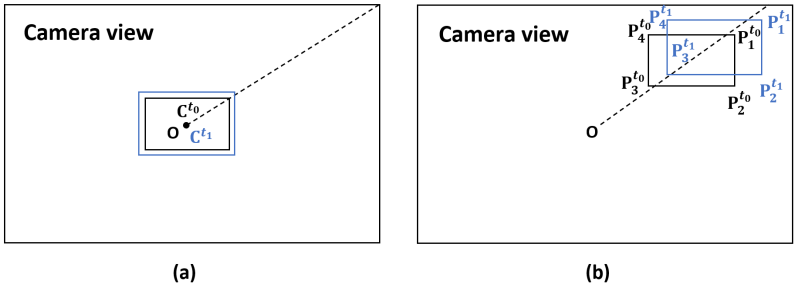


Figure 2.3: Different object positions and sizes in two consecutive frames t_0 (black) and t_1 (blue) in camera view. (a) central case (b) non-central case

2.3.3 Kinematic Automotive Model Formulation

The physical relationship between a forward moving car and a traffic sign is shown in Figure 2.4. Here we have two coordinate systems for each time step t_n, namely,

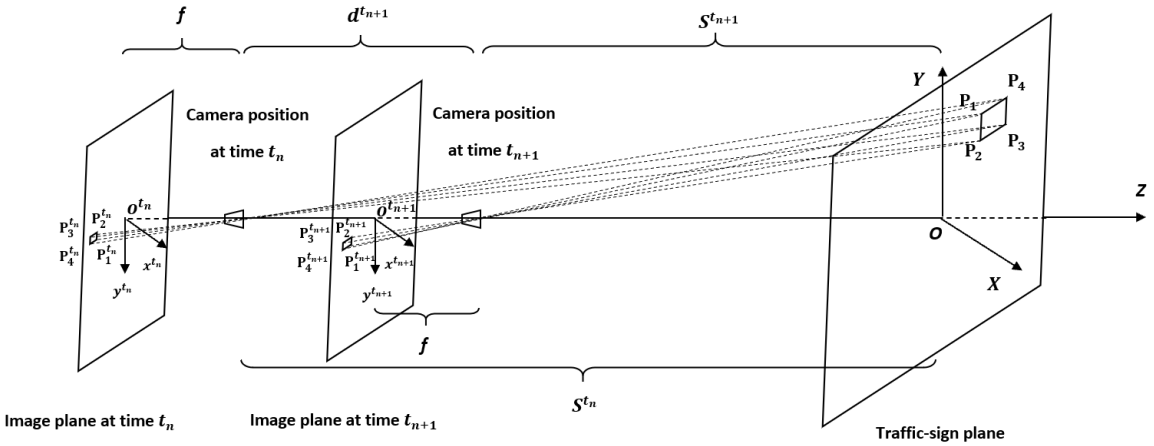


Figure 2.4: The geometric relationship across two consecutive frames at times t_n and t_{n+1} and the traffic-sign and its camera image.

a real-world traffic sign 3-dimensional coordinate system $O^{t_n}X^{t_n}Y^{t_n}Z^{t_n}$ and a projected image plane 2-dimensional coordinate systems $o^{t_n}x^{t_n}y^{t_n}$. The 3-D traffic sign coordinate system is actually a 2-D plane because of the planar surface assumption. Theoretically, the size of the traffic sign plane is infinite, and the size of image plane is defined by the camera with width w and height h. The X-axis is horizontal and parallel to the horizon, Y-axis is vertical and the Z-axis is the camera optical axis. Also, o^{t_n} is the origin of the image plane coordinate system and O^{t_n} is the intersection of the optical axis and the traffic sign plane at time t_n. Thus, for any n $(n|n \in \mathbb{Z}, n \geq 0)$, points o^{t_n}, $o^{t_{n+1}}$, O^{t_n} and $O^{t_{n+1}}$ are on the same line in the forward moving case. In Figure 2.4, P_1, P_2, P_3 and P_4 are the four corner points (key points) of a text-based road sign on the real-world traffic sign plane, while $P_i^{t_n} : (X_i^{t_n}, Y_i^{t_n})$ and $p_i^{t_n} : (x_i^{t_n}, y_i^{t_n})$ $(i = 1, 2, 3, 4)$ are their coordinates on the traffic-sign plane and the coordinates of their corresponding projection points on the image plane at time t_n, respectively. The camera's focal length f is the a known parameter, and its value is not needed in the algorithm in the forward-moving case. Denote by S^{t_n} the distance

between the traffic-sign plane and the image-plane along optical axis Z, and by d^{t_n} the moving distance from time t_{n-1} to t_n, where d^{t_n} is known, usually calculated from rotation times of the wheels. However, in implementing the proposed algorithms in this Chapter, the estimated distance $\widehat{d^{t_n}}$ is obtained by multiplying the velocity value v^{t_n} read from the speedometer in the time interval $(t_n - t_{n-1})$. Here, the velocity is assumed constant between two consecutive frames in view of high frame rates. (e.g. 0.0417 seconds for a 24 fps video). In this Chapter, without loss of generality, the constant velocity dynamic model is used to formulate the kinematic automotive model, which assumes the velocity is constant in sampling given time interval. Note that other motion models such as nearly constant acceleration or coordinated turn [67] can be handled by our algorithm.

Straight Line Motion

In the straight line motion case, the optical axis Z is perpendicular to both the real-world sign plane $O^{t_n}X^{t_n}Y^{t_n}$ and the image plane $o^{t_n}x^{t_n}y^{t_n}$ at every time t_n. Then, the traffic-sign plane does not change with time (i.e. the turning angle θ is 0°). Figure 2.5 (a) illustrates the moving direction of points on the traffic sign in camera's view, and the derivation of the moving direction of points is given in the Section 2.6. The knowledge of the moving directions of objects in the camera's view can help reduce the search region for detection and filter out false alarms. Note that no traffic sign plane can appear in the shaded area shown in Figure 2.5 (a) [14].

For convenience, here we only consider a single point $P : (X, Y)$ on the traffic sign. With the spatial relationship shown in Figure 2.5 (b), the same traffic sign is detected at time t_0 and t_k $(k \geq 1)$, and the sign may not be detected in any frame

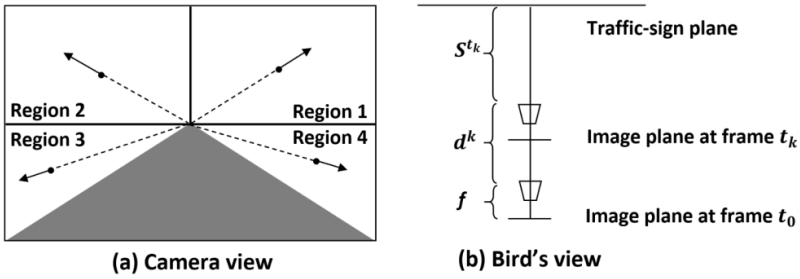


Figure 2.5: Straight line motion case. (a) Moving directions of points on the traffic-sign in camera's view (b) Kinematic model of straight line motion case in bird's view

between time t_0 and t_k. Here, $p^{t_0} : (x^{t_0}, y^{t_0})$ and $p^{t_k} : (x^{t_k}, y^{t_k})$ are projection points of P on the image plane $o^{t_0}x^{t_0}y^{t_0}$ and $o^{t_k}x^{t_k}y^{t_k}$, respectively, which are known. Our goal is to locate the projection point of P from time t_{k+1}.

From the pinhole camera model in (2.3.1), we get

$$\frac{X}{x^{t_0}} = \frac{Y}{y^{t_0}} = \frac{S^{t_0}}{f} \tag{2.3.3}$$

$$\frac{X}{x^{t_k}} = \frac{Y}{y^{t_k}} = \frac{S^{t_k}}{f} \tag{2.3.4}$$

where x^{t_0}, y^{t_0}, x^{t_k} and x^{t_k} are known and f is the camera's focal length parameter. Later, we will show that it is not necessary to know f in the straight line motion case. Since the moving distance d^k from t_0 to t_k is also known, we have

$$S^{t_0} = S^{t_k} + d^k \tag{2.3.5}$$

where $d^k = \sum_{i=1}^{k} d^{t_i} = \sum_{i=1}^{k} (v^{t_i} \cdot (t_i - t_{i-1}))$

Substituting (2.3.3) and (2.3.4) into (2.3.5), we get our plane distance at time t_k as

$$S_X^{t_k} = \frac{-d^k}{1 - \frac{x^{t_k}}{x^{t_0}}} \tag{2.3.6}$$

$$S_Y^{t_k} = \frac{-d^k}{1 - \frac{y^{t_k}}{y^{t_0}}} \tag{2.3.7}$$

where both $1 - \frac{x^{t_k}}{x^{t_0}}$ and $1 - \frac{y^{t_k}}{y^{t_0}}$ are both than zero, since $|x^{t_k}| > |x^{t_0}|$ and $|y^{t_k}| > |y^{t_0}|$, and the proof is given in the Section 2.6.

Equation (2.3.6) is obtained by eliminating X and (2.3.7) is obtained by eliminating Y from (2.3.3), (2.3.4) and (2.3.5). Ideally, these two values should be equal. However, considering sensor noise and truncation errors that may affect the accuracy of estimation, we take the average of these two to get our estimated planar distance as

$$\widehat{S^{t_k}} = \left(\frac{-d^k}{1 - \frac{x^{t_k}}{x^{t_0}}} + \frac{-d^k}{1 - \frac{y^{t_k}}{y^{t_0}}} \right) /2 \tag{2.3.8}$$

Substituting (2.3.8) back into (2.3.4), we have the estimated coordinates of point P on the traffic-sign plane as

$$\widehat{X} = \left(\frac{-d^k}{1 - \frac{x^{t_k}}{x^{t_0}}} + \frac{-d^k}{1 - \frac{y^{t_k}}{y^{t_0}}} \right) \frac{x^{t_k}}{2f} \tag{2.3.9}$$

$$\widehat{Y} = \left(\frac{-d^k}{1 - \frac{x^{t_k}}{x^{t_0}}} + \frac{-d^k}{1 - \frac{y^{t_k}}{y^{t_0}}} \right) \frac{y^{t_k}}{2f} \tag{2.3.10}$$

From the camera model, at time t_{k+1} we have

$$\frac{X}{x^{t_{k+1}}} = \frac{Y}{y^{t_{k+1}}} = \frac{S^{t_{k+1}}}{f} \tag{2.3.11}$$

with moving distance $d^{t_{k+1}}$ from t_k to t_{k+1}, we have

$$S^{t_k} = S^{t_{k+1}} + d^{t_{k+1}} \tag{2.3.12}$$

From (2.3.9), (2.3.10), (2.3.11) and (2.3.12), we get our predicted coordinates of projection point $p^{t_{k+1}}$ on the image plane as

$$\widehat{x^{t_{k+1}}} = \frac{\widehat{S^{t_k}} x^{t_k}}{\widehat{S^{t_k}} - d^{t_{k+1}}} = \frac{\left(\frac{-d^k}{2-\frac{2x^{t_k}}{x^{t_0}}} + \frac{-d^k}{2-\frac{2y^{t_k}}{y^{t_0}}} \right) x^{t_k}}{\left(\frac{-d^k}{2-\frac{2x^{t_k}}{x^{t_0}}} + \frac{-d^k}{2-\frac{2y^{t_k}}{y^{t_0}}} \right) - d^{t_{k+1}}} \tag{2.3.13}$$

$$\widehat{y^{t_{k+1}}} = \frac{\widehat{S^{t_k}} y^{t_k}}{\widehat{S^{t_k}} - d^{t_{k+1}}} = \frac{\left(\frac{-d^k}{2-\frac{2x^{t_k}}{x^{t_0}}} + \frac{-d^k}{2-\frac{2y^{t_k}}{y^{t_0}}} \right) y^{t_k}}{\left(\frac{-d^k}{2-\frac{2x^{t_k}}{x^{t_0}}} + \frac{-d^k}{2-\frac{2y^{t_k}}{y^{t_0}}} \right) - d^{t_{k+1}}} \tag{2.3.14}$$

We can further use (2.3.13) and (2.3.14) to recursively predict the coordinates of the projection point after t_{k+2}. If $|\widehat{x^{t_{k+1}}}|$ is larger than $w/2$ or $|\widehat{y^{t_{k+1}}}|$ larger than $h/2$, we know that the projection point $p^{t_{k+1}}$ is no longer on the image plane.

Lane Changing and Turning

The moving trajectory of the vehicles in the lane changing and turning cases are shown in Figure 2.6. The main difference between moving straight and lane changing or turning is in the turning angle θ. In the latter cases, with a non-zero angle parameter θ, the optical axis is no longer perpendicular to the traffic-sign plane, and the image planes in two consecutive frames are not parallel. Furthermore, all points on the traffic sign may not be on the same planar plane. These lead to the changes in the

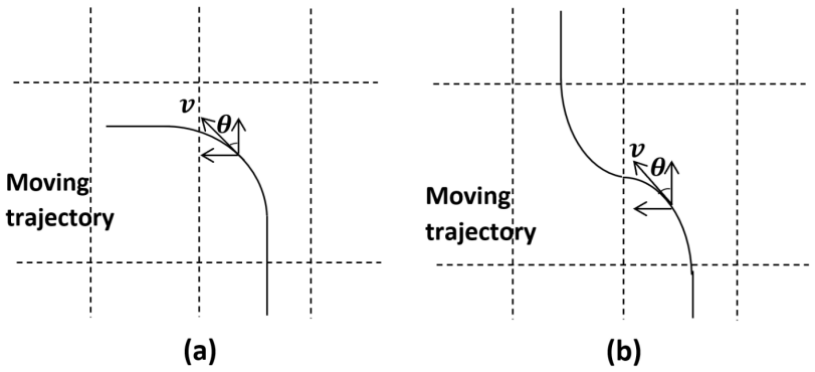


Figure 2.6: Moving trajectories in the lane changing and turning cases. (a) Turning (b) Lane changing

traffic sign plane at every time step. In contrast to the situation with the straight line motion, the direction of the optical axis Z keeps changing with time, so the origin of the traffic-sign plane is changing as well. Then, any two points with different horizontal coordinates will not share the same traffic-sign plane. Here, we define θ^{t_n} as the direction change between t_{n-1} and t_n. Since the time interval between two consecutive frames is tiny, we neglect the variation in θ within each time interval. Also we assume that the direction of the vehicle's velocity is tangential to the moving trajectory.

The spatial kinematic model in the line changing or turning case is given in Figure 2.7. From this geometric relationship, one can see that, although all projection points share the same image plane at same time step, they are not projected from the same traffic-sign plane. The ground truth of moving distance d is a curve, and we approximate it here with a straight segment at every time step.

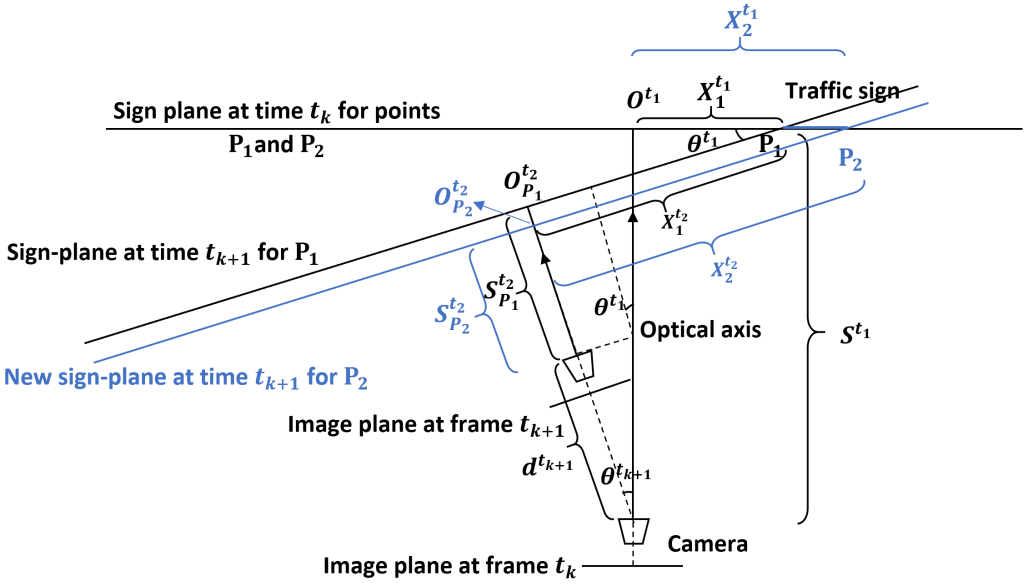


Figure 2.7: Kinematic model in the lane changing and turning cases.

2.3.4 Algorithm Description

We divide the localization process into two steps: 1) Estimating the distance between the image plane and the traffic-sign plane and estimating the coordinates of the traffic sign on the traffic-sign plane by using two consecutive frames with detections. 2) Predicting the coordinates of the traffic sign on the image plane recursively to the subsequent frames .

Estimation Stage

Consider a single point $P : (X, Y)$ on the traffic sign. Assume that the same traffic sign is detected at times t_0 and t_k $(k \geq 1)$. Thus, the values of $p^{t_0} : (x^{t_0}, y^{t_0})$ and

$p^{t_k} : (x^{t_k}, y^{t_k})$ are known. From the camera model in (2.3.1), we get

$$\frac{X^{t_0}}{x^{t_0}} = \frac{Y^{t_0}}{y^{t_0}} = \frac{S^{t_0}}{f} \tag{2.3.15}$$

$$\frac{X^{t_k}}{x^{t_k}} = \frac{Y^{t_k}}{y^{t_k}} = \frac{S^{t_k}}{f} \tag{2.3.16}$$

where (X^{t_n}, Y^{t_n}) are the coordinates of the traffic sign on the traffic-sign plane and S^{t_n} is the distance between image plane $o^{t_n}x^{t_n}y^{t_n}$ and traffic sign plane $O^{t_n}X^{t_n}Y^{t_n}$ at time t_n.

Since the turning angle θ affects only the horizontal axis, we have

$$Y^{t_0} = Y^{t_k} \tag{2.3.17}$$

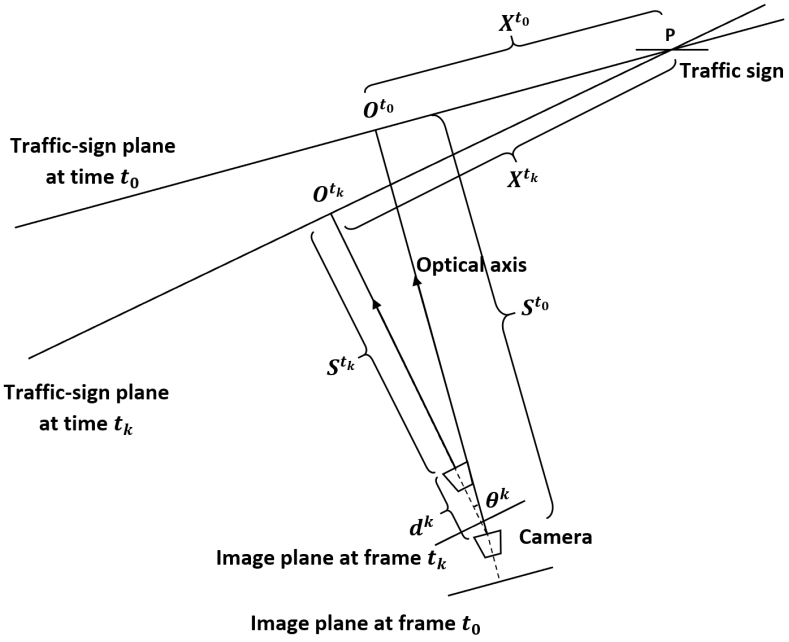


Figure 2.8: Illustration of the kinematic geometric relationships in the estimation step.

From the kinematic geometric relationship shown in Figure 2.8, we have

$$S^{t_k} = (S^{t_0} - X^{t_0} \tan\theta^k) \cos\theta^k - d^k \tag{2.3.18}$$

$$X^{t_k} = (S^{t_k} + d) \tan\theta^k + \frac{X^{t_0}}{\cos\theta^k} \tag{2.3.19}$$

where $d^k = \sum_{i=1}^{k} d^{t_i} = \sum_{i=1}^{k} (v^{t_i} \cdot (t_i - t_{i-1}))$ is the approximate moving distance from time t_0 to the t_k, and $\theta^k = \sum_{i=1}^{k} \theta^{t_i}$ is the directional difference between times t_0 and t_k. Thus, the smaller the value of k, the better the estimation results will be.

To predict the traffic sign position on the image plane to time instants beyond t_{k+1}, we need to know (X^{t_k}, Y^{t_k}) and S^{t_k}. Substituting (2.3.15), (2.3.16) and (2.3.19) into (2.3.18), we get

$$\widehat{S^{t_k}} = \frac{d\ (-1 - \frac{f \sin\theta^k \cos\theta^{t_k}}{x^{t_0}} + \sin^2\theta^k)}{1 - (\frac{x^{t_k} \cos^2\theta^k}{x^{t_0}} - \frac{f \sin\theta^k \cos\theta^k}{x^{t_0}} - \frac{x^{t_k} \sin\theta^k \cos\theta^k}{f} + \sin^2\theta^k)} \tag{2.3.20}$$

Substituting (3.3.8) back into (2.3.16) we get

$$\widehat{X^{t_k}} = \frac{d(-1 - \frac{f \sin\theta^k \cos\theta^k}{x^{t_0}} + \sin^2\theta^k)}{1 - (\frac{x^{t_k} \cos^2\theta^k}{x^{t_0}} - \frac{f \sin\theta^k \cos\theta^k}{x^{t_0}} - \frac{x^{t_k} \sin\theta^k \cos\theta^k}{f} + \sin^2\theta^k)} \cdot \frac{x^{t_k}}{f} \tag{2.3.21}$$

$$\widehat{Y^{t_k}} = \frac{d(-1 - \frac{f_x \sin\theta^k \cos\theta^k}{x^{t_0}} + \sin^2\theta^k)}{1 - (\frac{x^{t_k} \cos^2\theta^k}{x^{t_0}} - \frac{f \sin\theta^k \cos\theta^k}{x^{t_0}} - \frac{x^{t_k} \sin\theta^k \cos\theta^k}{f} + \sin^2\theta^k)} \cdot \frac{y^{t_k}}{f} \tag{2.3.22}$$

Prediction Stage

From the camera model equation in (2.3.1), we get

$$\frac{X^{t_k}}{x^{t_k}} = \frac{Y^{t_k}}{y^{t_k}} = \frac{S^{t_k}}{f} \tag{2.3.23}$$

$$\frac{X^{t_{k+1}}}{x^{t_{k+1}}} = \frac{Y^{t_{k+1}}}{y^{t_{k+1}}} = \frac{S^{t_{k+1}}}{f} \tag{2.3.24}$$

Since the turning angle θ affects only horizontal axis, we have

$$Y^{t_{k+1}} = Y^{t_k} \tag{2.3.25}$$

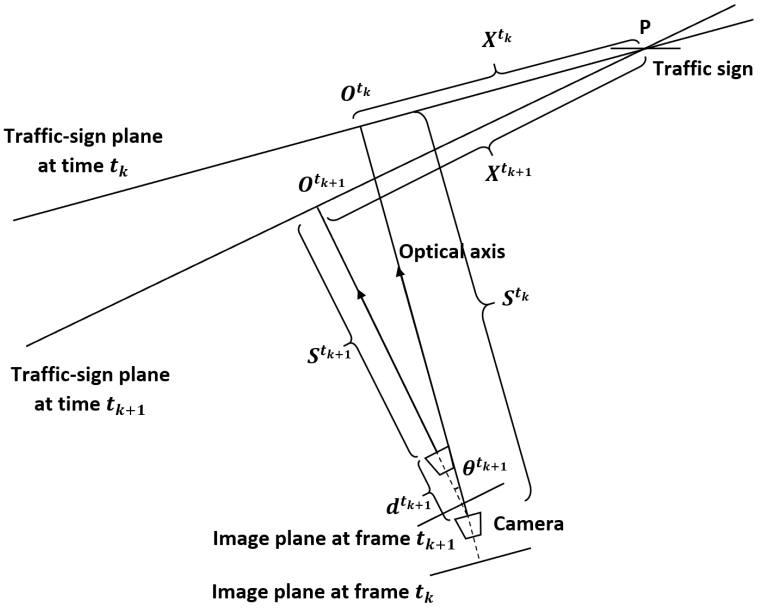


Figure 2.9: Illustration of the kinematic geometric relationship in the prediction step.

From the kinematic geometric relationship shown in Figure 2.9, we have

$$S^{t_{k+1}} = S^{t_k} \cos\theta^{t_{k+1}} - X^{t_k} \sin\theta^{t_{k+1}} - d^{t_{k+1}} \tag{2.3.26}$$

$$X^{t_{k+1}} = (S^{t_{k+1}} + d^{t_{k+1}}) \tan\theta^{t_{k+1}} + \frac{X^{t_k}}{\cos\theta^{t_{k+1}}} \tag{2.3.27}$$

where $d^{t_{k+1}} = v^{t_k+1}(t_{k+1} - t_k)$, which is the approximate moving distance from time

t_k to t_{k+1}, and $\theta^{t_{k+1}}$ is the directional difference between time t_k and t_{k+1}.

Substituting (3.3.13), (3.3.14) (3.3.15) into the (3.3.12) we get

$$\widehat{x^{t_{k+1}}} = \frac{(S^{t_k}\sin\theta^{t_{k+1}} - X^{t_k}\sin\theta^{t_{k+1}}\tan\theta^{t_{k+1}} + \frac{X^{t_k}}{\cos\theta^{t_{k+1}}})f}{S^{t_k}\cos\theta^{t_{k+1}} - X^{t_k}\sin\theta^{t_{k+1}} - d^{t_{k+1}}} \tag{2.3.28}$$

$$\widehat{y^{t_{k+1}}} = \frac{Y^{t_k}f}{S^{t_k}\cos\theta^{t_{k+1}} - X^{t_k}\sin\theta^{t_{k+1}} - d^{t_{k+1}}} \tag{2.3.29}$$

Details of the proposed text-based traffic sign localization algorithm are given in Algorithm 1.

Algorithm 1: Text-based traffic sign localization algorithm description

Data: v^{t_k} and θ^{t_k}: velocity and directional information for every frame k at time t_k, $k \geq 0$
M: the number of corner/key points for the text-based traffic sign,
initial detection: same traffic sign is detected at time t_0 and t_k, $p_m^{t_0} : (x_m^{t_0}, y_m^{t_0})$, and $p_m^{t_k} : (x_m^{t_k}, y_m^{t_k})$, $m = 1, 2, ..., M$

```
i = k;
while not out of plane do
    if k ≥ 2 then
        d^{t_i} = d^k = Σ_{n=1}^{k} d^{t_n}, θ^{t_i} = θ^k = Σ_{n=1}^{k} θ^{t_n}, p_j^{t_{i-1}} = p_j^{t_0};
    end
    for j = 1 to M do
        calculate P_j^{t_i} : (X_j^{t_i}, Y_j^{t_i}) and S_{P_j}^{t_i} by calling function
          estimation(p_j^{t_{i-1}}, p_j^{t_i}, θ^{t_i}, d^{t_i}) ← using estimation model equation in
          (3.3.8), (3.3.9) and (3.3.10);
        calculate p_j^{t_{i+1}} : (x_j^{t_{i+1}}, y_j^{t_{i+1}}) by calling function
          prediction(p_j^{t_i}, P_j^{t_i}, S_{P_j}^{t_i}, θ^{t_{i+1}}, d^{t_{i+1}}) ← using prediction model equation
          in (3.3.16) and (3.3.17);
    end
    Draw polygon (corner points p^{t_{i+1}}) on frame i + 1 at time t_{i+1};
end
```

2.4 Performance Evaluation and Discussions

2.4.1 Dataset

In the literature, the two recent text-based detection algorithms in [4] and [15] are evaluated based on the Traffic Guide Panel dataset [15]. The Traffic Guide Panel dataset is a benchmark dataset containing 3,841 images in total (2,315 images contain highway guide panels and 1,526 contains no traffic signs). However, this public text-based traffic sign dataset is only for detection benchmarking, not localization.

There is no public video dataset available for text-based traffic signs with known moving distance (d) and direction (θ) for every frame. To evaluate the effectiveness of our proposed traffic sign localization algorithms, we collected a representative dataset called the ETFLab Text-based Traffic Sign Video Dataset (ETFLab-TTSVD). Our video dataset is collected using a fixed camera mounted on the interior of the windshield. The videos are captured with 1080p (1920x1080) resolution at 24 frame per second rate (fps) for a duration of one second each. Thus, there are 24 frames in one video. There are 30 videos in total and each video contains at least 1 text-based traffic sign on the highway. The speed and the turning angle are roughly constant in each video. The focal length of the camera is fixed at 24 mm. Our video dataset is available from [78].

2.4.2 Evaluation Metrics

The evaluation metric we use to measure the performance of our proposed algorithm is Intersection over Union (IoU) [4], which is commonly used in the area of object detection and segmentation. IoU measures whether the predicted bounding box is

true or false. It is defined as

$$\text{IoU} = \frac{\text{Intersection}(G, P)}{\text{Union}(G, P)} \tag{2.4.1}$$

where G is the ground truth bounding box and P is the predicted bounding box. The predicted result is normally considered good, if IoU is larger than 0.5 [7], [73]. Then, we calculate the Precision, Recall and $\text{F}_{\text{measure}}$, which are defined as [4]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.4.2}$$

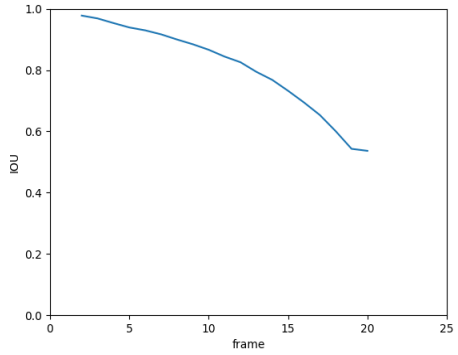
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.4.3}$$

$$\text{F}_{\text{measure}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4.4}$$

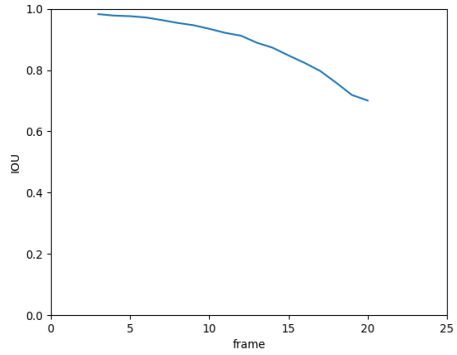
where TP, FP and FN are the true positive, false positive and false negative (miss) rates respectively . Since our method is a one-to-one prediction algorithm, there is no false negative, the value of Recall is constant at 1.0. To further test if the predicted results are accurate enough for post recognition functionality, we set the IoUT to 0.8.

2.4.3 Experimental Details and Overall Results

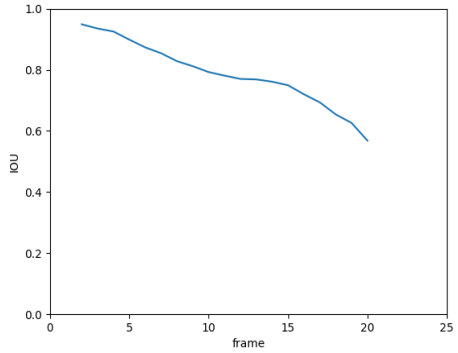
We implemented our proposed algorithm, which is described in Algorithm 1 in the Python 3.7 language on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. We tested both the straight line motion and turning cases when the value of k is 1 and 2. The resulting IoU values are shown in Figure 2.10. The IoU of prediction results with straight line motion are greater then those with turning for both $k = 1$ and 2. Their corresponding Precision, Recall and $\text{F}_{\text{measure}}$ results for the following 10



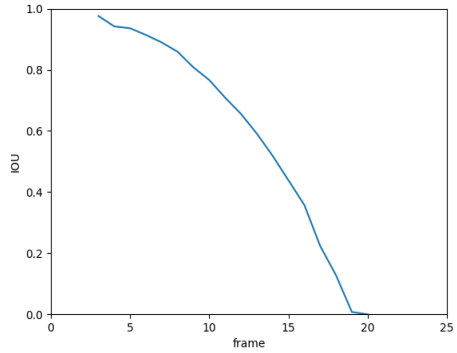
(a) Straight line motion, $k = 1$



(b) Straight line motion, $k = 2$



(c) Turning, $k = 1$



(d) Turning, $k = 2$

Figure 2.10: IoU performance with straight line motion and turning for $k = 1$ and $k = 2$.

and 20 frames are given in Tables 2.1 and 2.2, respectively, with IoUT = 0.5.

Figure 2.10 (a), (b) and Table 2.1 show that, with straight line motion, the proposed text-based traffic sign localization algorithm can achieve 100% in Precision, Recall and $F_{measure}$ for the next 20 frames when k is 1 or 2. With turning, as shown in Figure 2.10 (c), (d) and Table 2.2, the proposed localization algorithm can achieve 100% in Precision, Recall and $F_{measure}$ for the following 20 frames when $k = 1$ and for the following 10 frames when $k = 2$. However, for the following 20 frames when $k = 2$, Precision, Recall and $F_{measure}$ values are only 60%, 100%, 75%, respectively.

Table 2.1: Precision, Recall and $F_{measure}$ with straight line motion (IoUT = 0.5)

Straight line motion	Precision	Recall	$F_{measure}$
10 frames, $k = 1$	1.0	1.0	1.0
10 frames, $k = 2$	1.0	1.0	1.0
20 frames, $k = 1$	1.0	1.0	1.0
20 frames, $k = 2$	1.0	1.0	1.0

Table 2.2: Precision, Recall and $F_{measure}$ with turning (IoUT = 0.5)

Turning	Precision	Recall	$F_{measure}$
10 frames, $k = 1$	1.0	1.0	1.0
10 frames, $k = 2$	1.0	1.0	1.0
20 frames, $k = 1$	0.9	1.0	0.95
20 frames, $k = 2$	0.6	1.0	0.75

Over time, the drop rate of IoU is much faster in the turning case when $k = 2$ than the other three cases.

If a detection is missed during the initial detection stage, the prediction performance is greatly affected in the turning case. After initial detection, since the proposed localization algorithm is independent of the camera parameter, it can successfully handle many challenging situations such as uneven lighting and occlusion. Examples of the predicted and ground truth bounding boxes for both cases when $k = 1$ are given in Figures 2.11 and 2.12.

We compare our prediction results with those of existing text-based traffic sign methods in Table 2.3. In [4], a cascaded segmentation-detection framework for text-based traffic sign detection was presented with significant improvement in Precision, Recall and $F_{measure}$ compared to the other deep-learning based methodologies proposed in [15], [53]. Consider the first 10 following frames and the following 20 frames when $k = 1$, the Precision, Recall and $F_{measure}$ values of our prediction are better than those of the state-of-the-art detection results in [4]. The computational time with

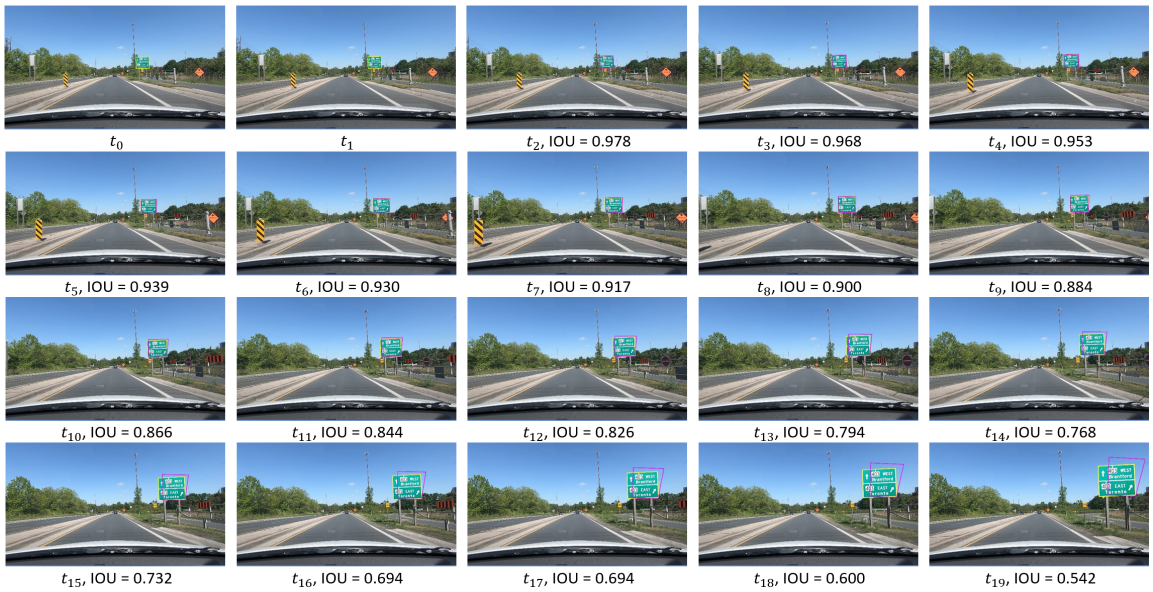


Figure 2.11: Results of traffic-sign position prediction with straight line motion when $k = 1$. Ground truth bounding box in yellow and predicted bounding box in magenta.

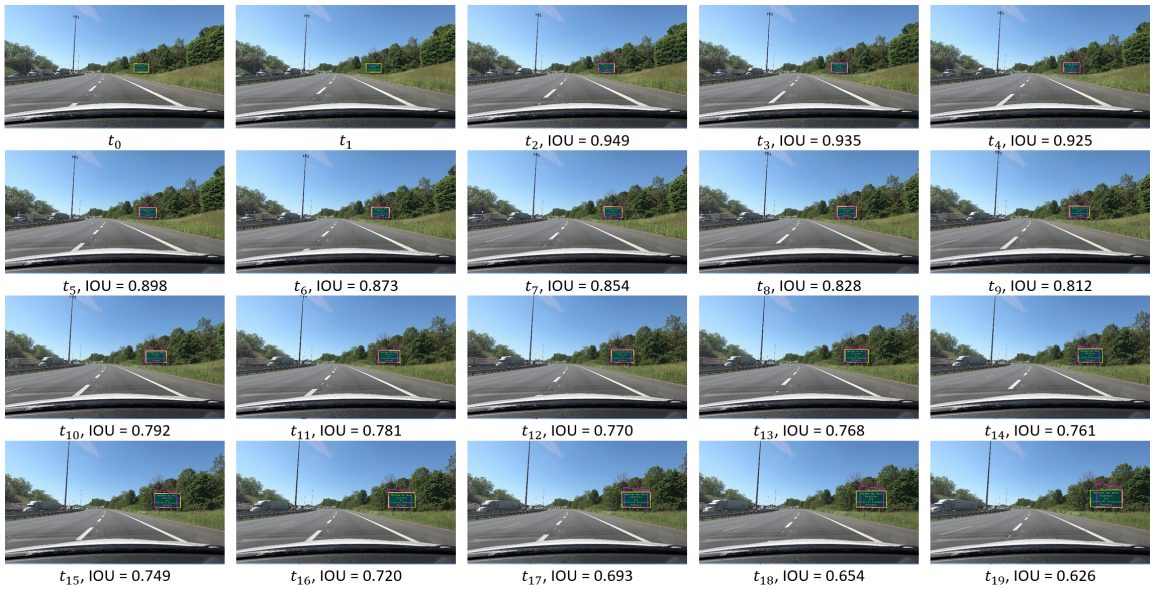


Figure 2.12: Results of traffic-sign position prediction with turning when $k = 1$. Ground truth bounding box in yellow and predicted bounding box in magenta.

CPU only is 0.012s for a 1080p resolution image, which is one tenth of the fastest existing GPU-based implementation in [4]. This means that in real application it is possible to have 4 detections and more than 30 localizations in one second. Our proposed algorithm is implemented using the Python language, and faster runtimes can be achieved with C++ and GPU implementations.

Table 2.3: Precision, Recall, $F_{measure}$, and running times of different text-based traffic sign methods. (IoUT = 0.5)

Method	Precision	Recall	$F_{measure}$	Time(s)	Device
Epshtain et al. [46]	0.35	0.41	0.38	2.51	CPU
Gómez and karatzas [45]	0.46	0.53	0.49	1.32	CPU
Jaderberg et al. [53]	0.59	0.71	0.64	4.53	GPU&CPU
Rong et al. [15]	0.73	0.64	0.68	0.16	GPU
Zhu et al. [4]	0.90	0.87	0.88	0.15	GPU&CPU
Ours (10 frames)	**1.00**	**1.00**	**1.00**	**0.012**	CPU
Ours (20 frames, $k = 1$)	**0.95**	**1.00**	**0.97**	**0.012**	CPU
Ours (20 frames, turn, $k = 2$)	0.6	1.00	0.75	**0.012**	CPU

Furthermore, as shown in Table 2.4, when IoUT=0.8, our prediction results for the first 10 frames are comparable results in terms of Precision, Recall and $F_{measure}$ values to those in the turning case, which means that the prediction results can be used in the subsequent tracking and recognition stages. However, when missed detections occur (i.e., $k > 1$), the prediction results over 20 frames in the turning case are not accurate. The performance of the proposed localization algorithm degrades fast because of the less accurate estimates with poor initial detection.

Table 2.4: Performance of Precision, Recall and $F_{measure}$ in both cases (IoUT = 0.8)

First 10 frames	Precision	Recall	$F_{measure}$
Straight, $k = 1$	1.0	1.0	1.0
Straight, $k = 2$	1.0	1.0	1.0
Turning, $k = 1$	0.8	1.0	0.89
Turning, $k = 2$	0.7	1.0	0.82

2.4.4 Robustness Analysis and Limitations

With prediction being a post-detection step, the performance of the proposed traffic-sign localization algorithm will be affected by the performance of text-based traffic sign detection. To analyze the robustness of our proposed algorithm, we take the poor initial estimates due to the probability of detection into consideration.

The state-of-the-art text-based traffic sign detection method in [4] achieves a precision of 0.9. As defined in (2.3.2), a probability of detection of 0.9 means that the second detection in frame $k = 1$ is 90% and for $k = 2$ it is 9%. The IoU indicates how accurate a detected region is. In general, a predicted region is considered good or correct if its IoU greater than 0.5 [7], [73]. However, for text-based traffic sign algorithm, from our experiments, detected regions may not be accurate enough for the post-text recognition stage if the IoU less than 0.8. This is because some characters may be lost with lower IoU values. The IoU results for various values of different motion models are shown in Figure 2.10. The predicted IoU results do not change with k in the straight line motion case. However, with $k = 2$, the IoU results degrade faster in the turning case.

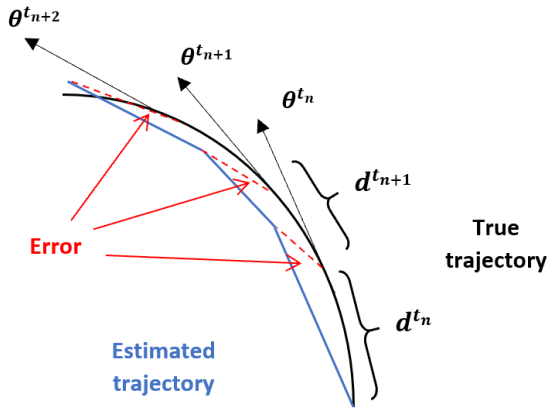


Figure 2.13: Accumulating prediction errors in the vehicle turning case.

The reason for this faster degradation is the poor approximation of angle θ. As illustrated in Figure 2.13, the actual vehicle trajectory is not always a straight line. In the proposed model, we assume that the movement across two consecutive frame is straight. Since prediction errors accumulate over time, the poorer initial estimates with $k = 2$ result in faster degradation of IoU over time. In addition, faster vehicle motion can degrade the IoU results as well. Under the condition of a fast moving vehicle negotiating a curve on highway, our proposed algorithm still gets correct prediction results for at least the first 10 frames even when miss detection happens.

The limitation of our work is that it needs to know the moving distance (or speed) and the turning direction θ from on-board sensors. Errors in these on-board sensor data can further degrade the performance of sign detection. A potential topic for future research is the on-line egomotion estimation with consideration for egomotion estimation errors in sign localization.

2.5 Conclusions

In this Chapter, an improved kinematic automotive model and a text-based traffic sign localization algorithm were presented. The proposed method fuses the information from the front camera and other on-board data source based on the spatial relationship between traffic signs and different kinematic vehicles motion models. The localization algorithm can handle real-world problems such as uneven lighting and occlusion since the prediction accuracy is independent of camera parameters given accurate on-board sensor data. The experimental results shows that, with accurate initial detections, our work can be used to significantly reduce the computational cost of sign localization and prediction, while maintaining high values of IoU of predicted

text-based traffic sign bounding boxes in real-time applications. The proposed algorithm yields acceptable prediction results even with poor initial estimates. Moreover, the prediction results can be also used as an improved predicted state of a Kalman filter based tracking algorithm.

2.6 Appendix: Apparent moving direction of a stationary object in a moving camera's view

Here, we consider a single point on a stationary text-based traffic sign and a pinhole camera model. As illustrated in Figure 2.14, the point $P : (X, Y)$ is on the road sign plane $OXYZ$. Denoted by $o^{t_n}x^{t_n}y^{t_n}$ and $o^{t_{n+k}}x^{t_{n+k}}y^{t_{n+k}}$, the two frames (image planes) at time t_n and t_{n+k} $(k = 1, 2, ...)$, the projection point of P on them are $p^{t_n} : (x^{t_n}, y^{t_n})$ and $p^{t_{n+k}} : (x^{t_{n+k}}, y^{t_{n+k}})$, respectively. Here, f is the focal length of the camera, d^k is the moving distance from frame t_n to t_{n+k} along the optical axis Z, and S^{t_n} is the actual distance between the camera plane and the traffic sign plane at t_n. The optical axis Z is perpendicular to the traffic sign plane and the image planes, and the intersection points of the traffic sign plane and the two image planes with the optical axis are O, o^{t_n} and $o^{t_{n+k}}$, respectively. For convenience, we define O, o^{t_n} and $o^{t_{n+k}}$ as the origin of their respective plane.

According to our assumptions, Z is approximately parallel to the ground. From

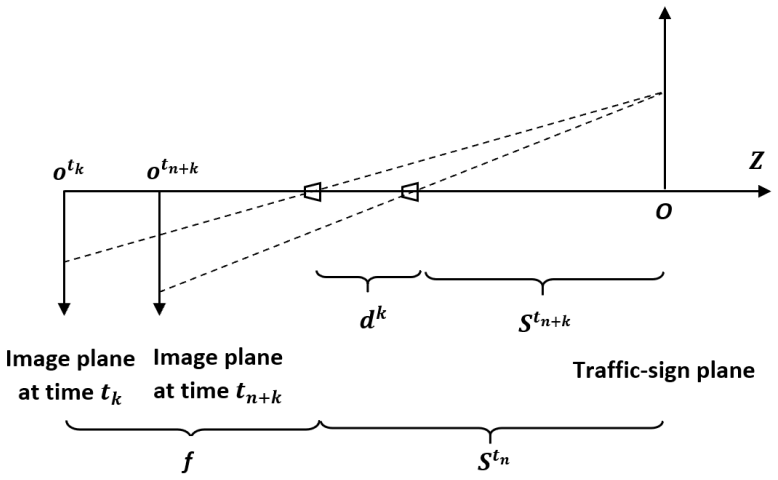


Figure 2.14: Geometric relationship between two frames t_n and t_{n+k} $(k = 1, 2, ...)$

the camera model in (2.3.1), we get

$$p^{t_n} : (x^{t_n}, y^{t_n}) = \left(\frac{f}{S^{t_n}} X, \frac{f}{S^{t_n}} Y \right) \tag{2.6.1}$$

$$p^{t_{n+k}} : (x^{t_{n+k}}, y^{t_{n+k}}) = \left(\frac{f}{S^{t_n} - d^k} X, \frac{f}{S^{t_n} - d^k} Y \right) \tag{2.6.2}$$

Here, p^{t_n} is known from the detection in frame t_n, but the real-world P and S^{t_n} are unknown parameters. Also, f and d^k are known from the camera parameters and on-board sensors, respectively. We show that the values of f and d^k have no impact on predicting the motion direction. The ratios of $x_{t_n}/x_{t_{n+k}}$ and $y_{t_n}/y_{t_{n+k}}$ are given by

$$\frac{x^{t_n}}{x^{t_{n+k}}} = \frac{\frac{f}{S^{t_n} X}}{\frac{f}{S^{t_n} - d^k} X} = \frac{S^{t_n} - d^k}{S^{t_n}} \tag{2.6.3}$$

$$\frac{y^{t_n}}{y^{t_{n+k}}} = \frac{\frac{f}{S^{t_n}} Y}{\frac{f}{S^{t_n} - d^k} Y} = \frac{S^{t_n} - d^k}{S^{t_n}} \tag{2.6.4}$$

From (2.6.3) and (2.6.4), we have

$$\frac{x_{t_n}}{x_{t_{n+k}}} = \frac{y_{t_n}}{y_{t_{n+k}}} \tag{2.6.5}$$

Thus, points o, p^{t_n} and $p^{t_{n+k}}$ are on the same line.

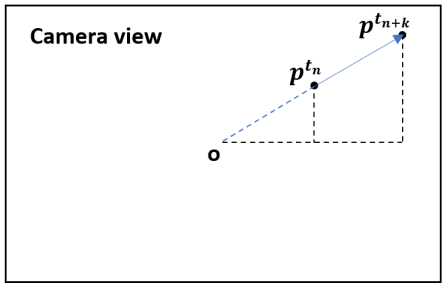


Figure 2.15: Apparent moving direction of a stationary object as seen from a straight-moving car

Thus, for a straight-moving car, the moving direction of a non-center point p in the subsequent following frames is $\vec{op}$, as shown in Figure 2.15.

Chapter 3

Text-based Traffic Sign Detection in Video using Kinematic Automotive Model

3.1 Abstract

Traffic sign detection plays an essential role in the modern Traffic Sign Recognition (TSR) module. Traffic signs can be categorized into two groups, graphics-based and text-based. Compared to graphics-based traffic signs, only a few algorithms have focused on text-based traffic signs. Due to the complex nature of the urban environment, in urban traffic scenarios, text-based traffic signs detection is a challenging problem. In this Chapter, we propose a text-based traffic sign detection algorithm for video, including a more detailed and accurate search region definition algorithm based on the kinematic automotive model. The search regions of text-based traffic signs can be modelled and estimated mathematically based on the kinematic states

of vehicles and the spatial-temporal relationships between motionless traffic signs and moving vehicles. The potential candidate regions will be extracted within the defined search regions using the Contrast-Enhanced Maximally Stable Extremal Regions (CE-MSERs) detector and then finally selected based on the structural-temporal information. The proposed text-based traffic sign detection algorithm has achieved overall precision and recall of 0.95 and 0.97, respectively, with a high Intersection over Union Threshold (IoUT). The computational time of the proposed approach is 0.035s per frame for a 1080p video in Python 3.7 on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. Thus, the proposed algorithm can be used to substitute existing algorithms and integrated into tracking algorithms at a significantly reduced computational cost.

3.2 Introduction

Advanced Driver Assistance Systems (ADAS) are intelligent systems embedded inside the vehicle intended to aid safe driving and get better informed. Among techniques in the ADAS, traffic sign detection and recognition system plays an essential role in autonomous driving [1] and mobile mapping [2]. Traffic signs can be generally categorized into two groups, namely, graphics-based and text-based traffic signs. Graphic-based traffic signs usually provide environmental information such as stop sign, icy road sign, and construction sign, while text-based traffic signs contain semantic information to guide drivers towards their destination, which is useful, especially in poor GPS and mobile internet signal areas.

As a prerequisite task of traffic sign recognition, a timely and accurate traffic signs detection functionality is vital and has significant impacts on the performance of

subsequent recognition functionality. Poor performance of detected text-based traffic sign bounding box areas will result in missing characters in the post recognition stage. Many existing works have focused on graphic-based traffic sign detection in recent years and achieved promising results on many public datasets, such as the MASTIF dataset [8] and German Traffic Sign Detection Benchmark (GTSDB) [12]. While only few research studies have focused on text-based traffic sign detection. This maybe partially caused by the difficulties of the task itself [14] and insufficient public datasets as mentioned in [4], the Traffic Guide Panel dataset which was proposed in [15] is the only (partially) public text-based traffic sign dataset benchmark, which is an image dataset, not a video.

Text-based traffic sign detection is a challenging task mainly due to three aspects, namely, variations in text content (i.e. font, color, size, stroke width and multilingual nature), complexity of the background (i.e. tree leaves and building bricks) and interference factors during the image acquisition (i.e. uneven lighting, perspective distortion, defocused or motion-blur imagery, multi-orientation due to motion and partial or complete occlusion). The research explicitly focused on the detection of text-based traffic signs is limited. Informed by road sign detection in color video sequence [36], a method to detect texts in text-based traffic signs from videos was proposed in [37]. The method first defines the geometric relationship between a moving car and traffic signs by assuming that the traffic sign is on a planar surface perpendicular to the horizontal ground and that the camera moves along its optical axis that is roughly horizontal. Then, the orientation of the plane is estimated using three or more points in two consecutive frames. A multiscale text detection algorithm is proposed on each candidate traffic panel area using edge detection, adaptive search, Gaussian Mixture

Model (GMM) and geometric linear analysis to obtain the position of a text line and to track it with a feature-based tracking algorithm. In [38], the blue and white rectangular regions of interests (ROIs) were extracted using a color-based segmentation algorithm and Fast Fourier Transform (FFT), then the four corner points of the rectangular regions were reoriented horizontally to align text characters. After analyzing the chrominance and luminance, an adaptive segmentation is carried out, and connected components labeling and position clustering are done for the arrangement of the different characters on the panel. In [39], the traffic signs that are located above ground and on the right side of the road are separated into two independent regions of interest. Then, the blue and white traffic panel regions were extracted for every single image based on color segmentation and Bag of Visual Words (BOVW) approach [40], and then classified the regions using classifiers, which was trained by support vector machines [33] or Naïve Bayes [41]. In [14], the pinhole camera model is used to restrict the search areas of the traffic sign in the detection stage. Then the potential text-based traffic sign candidates are detected in the defined search region based on the combination of MSERs detector and HSV color thresholding. Finally, the false positive regions are eliminated with the temporal and structural information.

Recently, with the intense power of deep learning, some text-based traffic sign detection methods have achieved promising results. In [15], a Cascaded Localization Network (CLN) is proposed to detect text-traffic sign candidates in the first stage, and then to locate text regions and eliminate the false alarm in the second stage. In [42], the traffic sign ROIs are extracted using MSERs algorithms in gray and normalized RGB channels, and then the regions are classified into different classes, including both graphics-based and text-based traffic signs by their proposed multi-task convolutional

neural network. In [4], a text-based traffic signs detection algorithm was proposed based on cascaded segmentation detection networks, which can achieve the state-of-the-art 0.9 Precision with a computational speed of 0.15 seconds per frame.

Many recent works on text-based traffic sign detection originated from scene text detection [43], [44], [45], [46], [47], [48], [49]. Comprehensive surveys on scene text detection can be found at [7], [50]. The methods for scene text detection can be roughly divided into three groups, namely, sliding window-based method, connected component-based (CC-based) method, and deep learning-based method. The sliding window-based method uses multi-scale windows to move across the image to localize the high confidence text regions [51], [52], [53]. The main challenges of this method are training a powerful classifier by discriminative features and the heavy computation time caused by a large number of scanning window. Unlike the sliding window based method, the connect component-based (CC-based) method is more efficient. CC-based method assumes that the characters in the image generate connected components. And the pixels in the same CCs have the same properties, such as stroke width, pixel intensity and grayscale level. The total number of the connected components is much less than the scanning windows so that it will take shorter computational time. Two representatives in this category are Stroke Width Transformation (SWT)-based method [46] and Maximally Stable Extremal Regions (MSERs)-based method [30]. SWT-based method utilizes the property that local characters have uniform stroke width to filter out false alarms [54]. The MSERs-based method uses the uniformity of the pixel intensity of the text stroke. The advantage of using the MSERs-based method is that it is fast and able to handle images even in low resolutions and contrast. Many works [55], [56], [57] were inspired by SWT and MSERs

methods. Recently, by taking the advantages of both deep learning algorithms and strong computational power of GPUs, originating from SSD [58] and TextBoxes [59], deep learning-based method treats the text detection problem as a regression problem by assuming text region to be a common category of objects such as cars and have achieved the most promising result [53], [59], [60], [61], [62].

Existing text-based traffic sign detection methods have four main issues: 1) Most existing methods were explicitly developed for graphics-based traffic signs. Text-based traffic signs pose different challenges that have not been addressed in the literature. 2) Most text-based traffic signs detection algorithms focus on a single image, they do not take advantage of the temporal-spatial information in videos. 3) The main pitfall of the MSERs detector is massive and repeated detection [47], on average, it could have more than 3 thousand extracted regions in a 1080p resolution grayscaled image of the traffic scene. The number of regions will be even higher if combined with more image space such as HSV, RGB. 4) Deep learning-based algorithms are computationally burdensome, these kinds of detection methods may not be suitable for every frame in a complete traffic sign detection, recognition and tracking framework in real-time applications. These limitations provide the motivation for our work in this Chapter.

In this Chapter, in order to better adapt to real-time application and to address the limitations listed above, we propose a text-based traffic sign detection algorithm for video, including a more detailed and accurate search region definition algorithm based on the kinematic automotive model, which was proposed in our Chapter 2. The main contributions of our work are as follows: 1) We develop a fast and accurate search region definition algorithm based on the kinematic automotive model, which was proposed in Chapter 2, which takes the advantages of spatial-temporal information of

the previous frames and different kinematic vehicles motion models. 2) The proposed search region definition method includes a turning angle estimation algorithm, which details the relationship between the speed of the vehicle and turning angel based on the environmental conditions. 3) The proposed text-based traffic sign detection algorithm can be further integrated into the current TSR modules and traffic sign tracking algorithms. 4) Our work yields highly accurate detection results and at a significantly reduced computational cost. From the experiential detection results on real video data, the proposed algorithm has achieved overall a Precision and Recall of 0.95 and 0.97, respectively, with a high Intersection over Union Threshold (IoUT). The computational time of our approach is 0.035s per frame for a 1080p video in Python 3.7 on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. Thus, the proposed algorithm can be used to substitute existing algorithms and integrated into tracking algorithms at a significantly reduced computational cost.

The structure of the Chapter is as follows. In Section 3.3, a review of the kinematic automotive model is given first. In Section 3.4, the details of the proposed text-based traffic sign detection algorithm are presented. Experimental and evaluation results are presented and compared with the state-of-the-art text-based traffic sign detection algorithms in Section 4. Finally, conclusions are discussed in Section 3.5.

3.3 Review of the kinematic automotive model

This Chapter applies the model proposed in Chapter 2, which is summarized in this section, to detail the spatial-temporal relationship between traffic signs and different kinematic vehicles motion models.

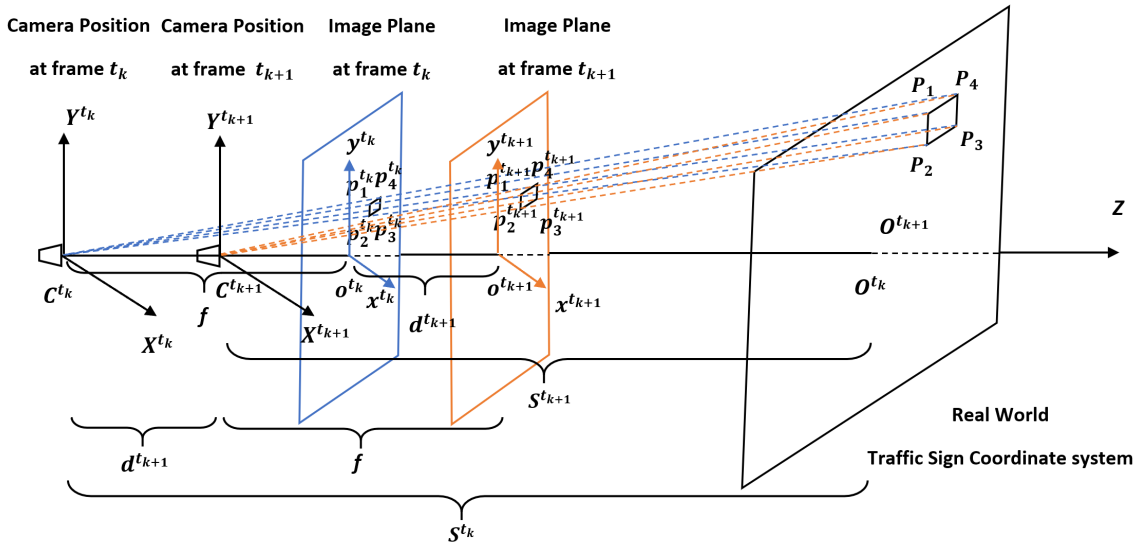


Figure 3.1: The geometric relationship across two consecutive frames at times t_k and t_{k+1} and the traffic-sign and its camera image.

3.3.1 Localize the traffic sign

The physical relationship between a forward moving car and a traffic sign at time t_k and t_{k+1}, $(k = 0, 1, 2, ..., n-1)$ is shown in Figure 3.1, the notations are defined as follows:

- $T = \{t_0, t_1, ..., t_n\}$ - sampling period,
  - t_0 - First time a target is detected.
  - t_n - Last time the target is detected on the image plane.
- $\boldsymbol{Z}$ - Optical axis which is assumed parallel to the ground. The camera moves along side its optical axis. The direction of $\boldsymbol{Z}$ may different at each time.
- $\boldsymbol{O}^{t_k}\boldsymbol{X}^{t_k}\boldsymbol{Y}^{t_k}\boldsymbol{Z}^{t_k}$ - 3-Dimension traffic sign coordinate system at time t_k.

  - $\boldsymbol{X}$ - Horizontal axis which is parallel to the ground and perpendicular to optical axis $\boldsymbol{Z}$.
  - $\boldsymbol{Y}$ - Vertical axis which is perpendicular to the ground.
  - $\boldsymbol{O}$ - Intersection of $\boldsymbol{Z}$ and X-Y plane.

- $\boldsymbol{o}^{t_k}\boldsymbol{x}^{t_k}\boldsymbol{y}^{t_k}$ - 2D camera image plane which is projected by real world coordinate system $\boldsymbol{O}^{t_k}\boldsymbol{X}^{t_k}\boldsymbol{Y}^{t_k}\boldsymbol{Z}^{t_k}$ at each time step t_k.
  - $\boldsymbol{x}, \boldsymbol{y}$ - Horizontal and vertical axis of image plane.
  - $\boldsymbol{o}$ - Intersection of $\boldsymbol{Z}$ and image plane.

- P^{t_k} - Point of the traffic sign on the real world coordinate system $\boldsymbol{O}^{t_k}\boldsymbol{X}^{t_k}\boldsymbol{Y}^{t_k}\boldsymbol{Z}^{t_k}$ at time t_k.

- p^{t_k} - Projected point of P^{t_k} on image plane $\boldsymbol{o}^{t_k}\boldsymbol{x}^{t_k}\boldsymbol{y}^{t_k}$ at time t_k.

- f - Camera predetermined parameter.

- $S_P^{t_k}$ - Distance between traffic sign and camera along optical axis $\boldsymbol{Z}$ of point P at time t_k.

- d^{t_k} - Moving distance during the time interval t_{k-1} to t_k.

- $\boldsymbol{v}_k^t$ - Velocity of car at time t_k, we assume it is same unchanged during the time interval t_k to t_{k+1}.

- θ^{t_k} - Difference of turning angle between time interval t_{k-1} to t_k. We define positive value for turning left and negative for right.

At each time step t_k, we define that the camera is the origin of the 3D real world coordinate system. P_1, P_2, P_3, P_4 are the four corner points (key points) of a text-based rectangular road sign in the real world traffic sign system. These corner points are often fixed in real world. However, since the camera moves at every time step, the coordinates of these corner points are changing with the time. In [37], they assumed that scene text lies on planar surfaces. We do not require this assumption in our work. Though these corner points stay on same traffic sign, they do not always share the same vertical traffic sign plane, which parallels 2D image plane. In addition, the image plane of time t_k and t_{k+1} may not always parallel to each other. In real world system, corner points $P_i^{t_k}$ $(i = 1, 2, 3, 4)$'s coordinates are represented by $(X_i^{t_k}, Y_i^{t_k}, Z_i^{t_k})$ at time t_k. Their corresponding projected points $p_i^{t_k}$ $(i = 1, 2, 3, 4)$ on image plane are represented by $(x_i^{t_k}, y_i^{t_k})$. The relationship between these two coordinate systems by using a pinhole camera model are defined by equation (3.3.1)

$$\frac{X_i^{t_k}}{x_i^{t_k}} = \frac{Y_i^{t_k}}{y_i^{t_k}} = \frac{Z_i^{t_k}}{f}, \qquad i = 1, 2, 3, 4 \tag{3.3.1}$$

where $Z_i^{t_k} = S_{P_i}^{t_k}$ by definition.

In this Chapter, without loss of generality, the constant velocity dynamic model is used to formulate the kinematic automotive model, which assumes the velocity is constant in sampling given time interval. Note that other motion models such as nearly constant acceleration or coordinated turn [67] can be handled by our algorithm as well.

3.3.2 Kinematic automotive model

For a forward-moving car, its kinematic states can be generally categorized into three cases, which are straight line motion, lane changing and turning. From Chapter 2, we know that straight line motion can be treated as a special case of lane changing and turning, where the value of turning θ is 0. In this Chapter, we only focus on the general turning scenario.

In general turning case, unlike our previous work in Chapter 2, in this Chapter, we do not have the on-board sensor information of turning direction θ at each time step. We will use the rules of physic to limit the range and predict the turning θ, which will be described in Section 3.4.1. To better address this problem, we further divide this localization process into two stages, which are the estimation and prediction stages.

During the estimation stage, by knowing the detection result of corner points $p_i \ \ (i = 1, 2, 3, 4)$ at time t_0 and t_m, i.e. $p_i^{t_0} : (x_i^{t_0}, y_i^{t_0})$ and $p_i^{t_m} : (x_i^{t_m}, y_i^{t_m})$, and car velocity v^t, $(t = t_0, t_1, ..., t_m)$. m is an integer and larger or equal to 1, and since currently there is no detection algorithm with 100% reliability, we need to take miss detection situation into consideration. The miss detection may occur during the time t_0 to t_m, the value of m depends on the probability of detection. Denoted by P_d the probability of detection, we have

$$Pr(m = N) = P_d \cdot (1 - P_d)^{N-1}, \quad m = 1, 2, 3, ..., n \tag{3.3.2}$$

For example, if $P_d = 90\%$, the probability of $m = 1$ is 90%, $m = 2$ is 9% and $m > 2$ is the rest 1%. And we will be able to estimate the coordinates of corner points in real world system, i.e. $P_i : (X_i^{t_m}, Y_i^{t_m}, S_{P_i}^{t_m}) \quad (i = 1, 2, 3, 4)$ with the knowledge of

turning direction θ, which will be used in the latter prediction stage to predict the coordinates of traffic sign on the image plane for the following frames recursively, i.e. $p_i^{t_k} : (x_i^{t_k}, y_i^{t_k}) \quad (k \in \mathbb{Z} \mid m \leq k \leq n)$.

Estimation stage

For convenience, here we only consider a single point $P : (X^{t_0}, Y^{t_0}, Z^{t_0})$ on the traffic sign in real world system and its corresponding projected point on image plane is denoted by p. Same traffic sign is detected consecutively at time t_0 and t_m ($m \in \mathbb{Z}|m \geq 1$), miss detection may occur between the time t_0 to t_m. $p^{t_0} : (x^{t_0}, y^{t_0})$ and $p^{t_m} : (x^{t_m}, y^{t_m})$ are projection points of P on image plane $o^{t_0}x^{t_0}y^{t_0}$ and $o^{t_k}x^{t_m}y^{t_m}$ respectively, and the value of them are known. From (3.3.1), we have

$$\frac{X^{t_0}}{x^{t_0}} = \frac{Y^{t_0}}{y^{t_0}} = \frac{Z^{t_0}}{f}, \tag{3.3.3}$$

$$\frac{X^{t_m}}{x^{t_m}} = \frac{Y^{t_m}}{y^{t_m}} = \frac{Z^{t_m}}{f}, \tag{3.3.4}$$

where $Z^{t_0} = S_P^{t_0}, Z^{t_m} = S_P^{t_m}$.

Since the turning angle θ affects only the horizontal axis, we have

$$Y^{t_0} = Y^{t_m} \tag{3.3.5}$$

As shown in Figure 3.2, the kinematic geometric relationship of t^0 and t^m, we have

$$S_P^{t_m} = S_P^{t_0} cos\theta^m - X^{t_0} sin\theta^m - d^m \tag{3.3.6}$$

$$X^{t_0} = X^{t_m} cos\theta^m - (S_P^{t_m} + d^m) sin\theta^m \tag{3.3.7}$$

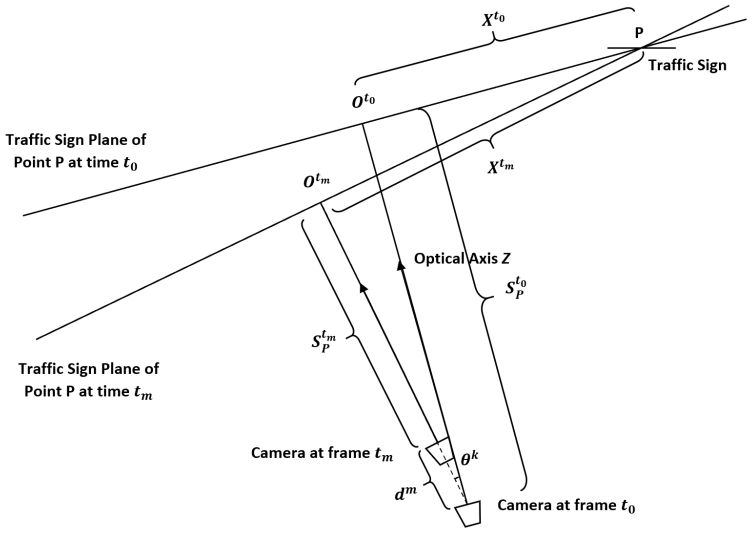


Figure 3.2: Illustration of the kinematic geometric relationships in the estimation step.

where $d^m = \sum_{i=1}^{m} d^{t_i} = \sum_{i=1}^{m} (v^{t_i} \cdot (t_i - t_{i-1}))$ is the approximate moving distance from time t_0 to t_m, $\theta^m = \sum_{i=1}^{m} \theta^{t_i}$ is the directional difference between time t_0 and t_m. In this model, we assume d^m is a straight line. However, it should be a curve line. Thus, the smaller m we have, the better estimation we will get.

To predict the traffic sign position on the image plane for the time after t_{m+1}, we need to know X^{t_m}, Y^{t_m} and S^{t_m}, substitute (3.3.3), (3.3.4) and (3.3.7) into (3.3.6), we get

$$S_P^{t_m} = \frac{d^m \cdot (-1 - \frac{f \sin\theta^m \cos\theta^{t_m}}{x^{t_0}} + \sin^2\theta^m)}{1 - (\frac{x^{t_m}\cos^2\theta^m}{x^{t_0}} - \frac{f\sin\theta^m\cos\theta^m}{x^{t_0}} - \frac{x^{t_m}\sin\theta^m\cos\theta^m}{f} + \sin^2\theta^m)} \tag{3.3.8}$$

Substitute back into the (3.3.4) we get the estimate coordinate of P at time t_m

$$X^{t_m} = \frac{d^m \cdot (-1 - \frac{f \sin\theta^m \cos\theta^m}{x^{t_0}} + \sin^2\theta^m)}{1 - (\frac{x^{t_m}\cos^2\theta^m}{x^{t_0}} - \frac{f\sin\theta^m\cos\theta^m}{x^{t_0}} - \frac{x^{t_m}\sin\theta^m\cos\theta^m}{f} + \sin^2\theta^m)} \cdot \frac{x^{t_m}}{f} \tag{3.3.9}$$

$$Y^{t_m} = \frac{d^m \cdot (-1 - \frac{f \sin\theta^m \cos\theta^m}{x^{t_0}} + \sin^2\theta^m)}{1 - (\frac{x^{t_m}\cos^2\theta^m}{x^{t_0}} - \frac{f\sin\theta^m\cos\theta^m}{x^{t_0}} - \frac{x^{t_m}\sin\theta^m\cos\theta^m}{f} + \sin^2\theta^m)} \cdot \frac{y^{t_m}}{f} \tag{3.3.10}$$

Prediction stage $(k \in \mathbb{Z} \mid m \leq k \leq n)$

From equation (3.3.1), we get that

$$\frac{X^{t_k}}{x^{t_k}} = \frac{Y^{t_k}}{y^{t_k}} = \frac{S_P^{t_k}}{f} \tag{3.3.11}$$

$$\frac{X^{t_{k+1}}}{x^{t_{k+1}}} = \frac{Y^{t_{k+1}}}{y^{t_{k+1}}} = \frac{S_P^{t_{k+1}}}{f} \tag{3.3.12}$$

Since the turning angle θ only affect horizontal axis, we have

$$Y^{t_{k+1}} = Y^{t_k} \tag{3.3.13}$$

From the kinematic geometric relationship as shown in Figure 3.3, we have

$$S_P^{t_{k+1}} = S_P^{t_k}\cos\theta^{t_{k+1}} - X^{t_k}\sin\theta^{t_{k+1}} - d^{t_{k+1}} \tag{3.3.14}$$

$$X^{t_{k+1}} = (S_P^{t_{k+1}} + d^{t_{k+1}})\tan\theta^{t_{k+1}} + \frac{X^{t_k}}{\cos\theta^{t_{k+1}}} \tag{3.3.15}$$

where $d^{t_{k+1}} = v^{t_k+1}(t_{k+1} - t_k)$, which is the approximate moving distance from time t_k to t_{k+1}, $\theta^{t_{k+1}}$ is the directional difference between time t_k and t_{k+1}.

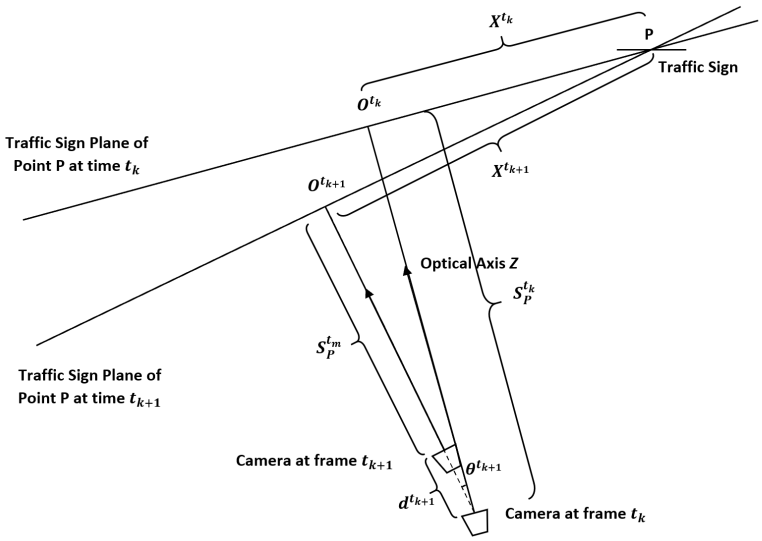


Figure 3.3: Illustration of the kinematic geometric relationship in the prediction step.

Substitute (3.3.13), (3.3.14), (3.3.15) into the (3.3.12) we get

$$x^{t_{k+1}} = \frac{(S_P^{t_k} \sin\theta^{t_{k+1}} - X^{t_k} \sin\theta^{t_{k+1}} \tan\theta^{t_{k+1}} + \frac{X^{t_k}}{\cos\theta^{t_{k+1}}})f}{S_P^{t_k} \cos\theta^{t_{k+1}} - X^{t_k} \sin\theta^{t_{k+1}} - d^{t_{k+1}}} \tag{3.3.16}$$

$$y^{t_{k+1}} = \frac{Y^{t_k} f}{S_P^{t_k} \cos\theta^{t_{k+1}} - X^{t_k} \sin\theta^{t_{k+1}} - d^{t_{k+1}}} \tag{3.3.17}$$

3.4 Proposed algorithm

By using the spatial-temporal information based on the kinematic automotive model, a tracking-based text-based traffic signs detection algorithm is proposed. The flowchart of the proposed algorithm and the result after each stage are shown in Figure 3.4. There are three major stages of the proposed tracking-based text-based traffic signs detection method, namely, define the text-based traffic sign search regions, text-based traffic sign candidates extraction and filtering, and text-based traffic sign candidate

selection.

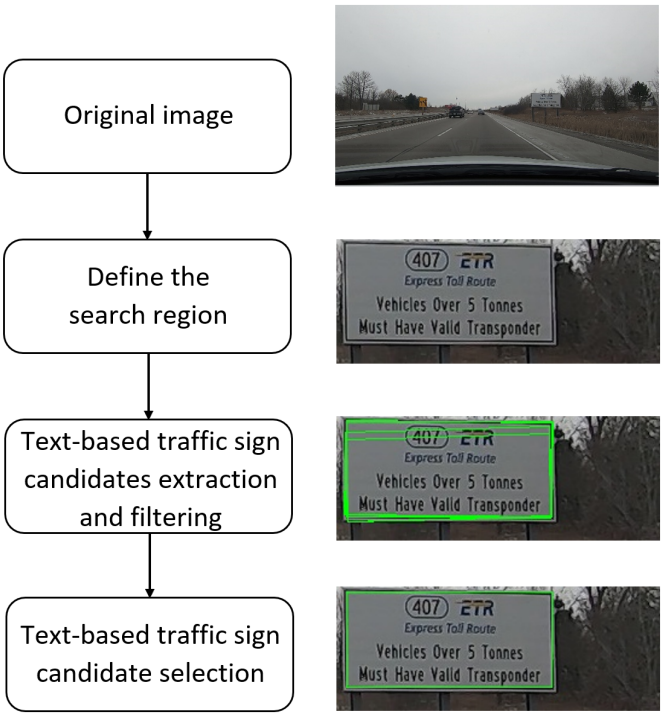


Figure 3.4: Flowchart of the proposed text-based traffic sign detection algorithm and results of the each step.

In define the text-based traffic sign search regions stage, the range of turning angle is first estimated based on the speed of the vehicles and the conditions of the environment. Next, by taking the advantages of the spatial-temporal correlation in videos, the kinematic automotive model is used to define the search regions of the text-based traffic sign.

In the text-based traffic sign candidates extraction and filtering stage, a contrast-enhanced MSERs algorithm is used within the defined search region, and most of the detected regions are filtered by the proper size of the traffic sign to reduce the computational time in post-stage.

In the text-based traffic sign candidate selection stage, since most of the non-text-based traffic sign regions are eliminated in the former stage, a nearest neighbour algorithm is used to choose the final predicted traffic sign region, which will also be used in the next frame.

More details of these three stages will be explained in the following sections.

3.4.1 Define the text-based traffic sign search regions

Unlike our previous work in Chapter 2, in this Chapter, we do not have the information of turning angle θ from on-board sensors at each time step. When negotiating a curve safely, the maximum turning angle is in proportion to the vehicle's speed. By using the range of the turning angle, we can localize all the possible positions of the traffic sign and thus define the search regions of current frames based on previous detection by using the kinematic automotive model. Our goal is to have a minimal search region, which contains all the possible locations of the traffic sign.

Some physical rules are used to estimate the range of the ideal turning angle. Assume no skidding during negotiating a curve, according to Newtons second law of motion [79], net force is equal to mass times acceleration. Thus, the centripetal force F_c, as shown in Figure 3.5, is

$$F_c = ma_c = m\frac{v^2}{r} \tag{3.4.1}$$

where r stands for the turning radius.

The kinetic frictional force F_k is defined by (3.4.2).

$$F_k = \mu_k mg \tag{3.4.2}$$

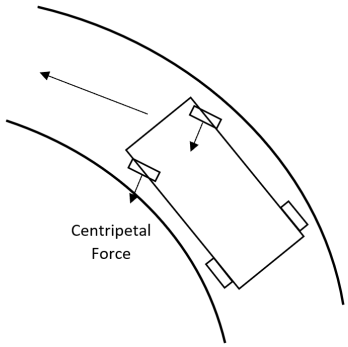


Figure 3.5: Illustration of the centripetal force in the turning scenario.

where μ_k is the coefficient of kinetic friction, g is the gravitational constant.

The value of μ_k only depends on the two contacted surface and varies between two surfaces. From the coefficient of friction table [80], we learned that the kinetic frictional coefficient of rubber on concrete is $0.6-0.85$ on a sunny day and $0.45-0.75$ on a rainy day.

To negotiate an unbanked curve, the force of kinetic friction is the only force to act in the direction of centripetal force and help a car to turn. Thus, we have

$$F_c \leq F_k \tag{3.4.3}$$

$$m\frac{v^2}{r} \leq \mu_k mg \tag{3.4.4}$$

$$r \geq \frac{v^2}{\mu_k g} \tag{3.4.5}$$

Also, the turning radius r should be greater than its manufactured minimum turning radius, and this value is variant based on different vehicle's brand. For example, the turning circle of a year 2020 Toyota Corolla is 5.3 meter [81].

From the equation of angular velocity, we have

$$\omega = \frac{\Delta\theta}{\Delta t} = \frac{v}{r} \tag{3.4.6}$$

$\Delta\theta$ is the directional difference of the turning angle in given sampling time interval Δt. Substitute (3.4.5) into (3.4.6), we get our estimated ideal turning angle, which is

$$|\theta| = \frac{vt}{r} \leq \frac{\mu_k g t}{v}\ radians \tag{3.4.7}$$

Similar to our previous work in Chapter 2, we use corner points p_i $(i = 1, 2, 3, 4)$ to locate the traffic signs. The range of turning angle θ is calculated by using the vehicle's speed in the corresponding estimation and prediction stages. As it is shown in Figure 3.6 (a) and (b), the localization results of p_i in the following frames are a group of points. A search region will be defined by connecting these points accordingly.

The computational cost of finding the search region depends on the step value of the turning angle θ. For example, let the speed of vehicle be 100 km/h, $\mu_k = 0.7$, $g = 9.8\ ms^{-2}$ in both estimation and prediction stages, by using (4.3.30) we get $|\theta| \leq 0.59\ radians$. If the step value equals 0.01, we will have approximately 120 estimated P_i after estimation stage, and more than 10 thousands predicted p_i in prediction stage. This hugely burdens the computational speed, and most of the prediction results are repeated. To reduce the computational load, we use a larger step value (eg. 0.3) and find the minimal horizontal rectangle region, which include all these points, as it is shown in Figure 3.6 (c).

Figure 3.6: The result after the defined search region step. (a) and (b) Predicted corner points in red (c) Final defined search region.

3.4.2 Text-based traffic sign candidates extraction and filtering

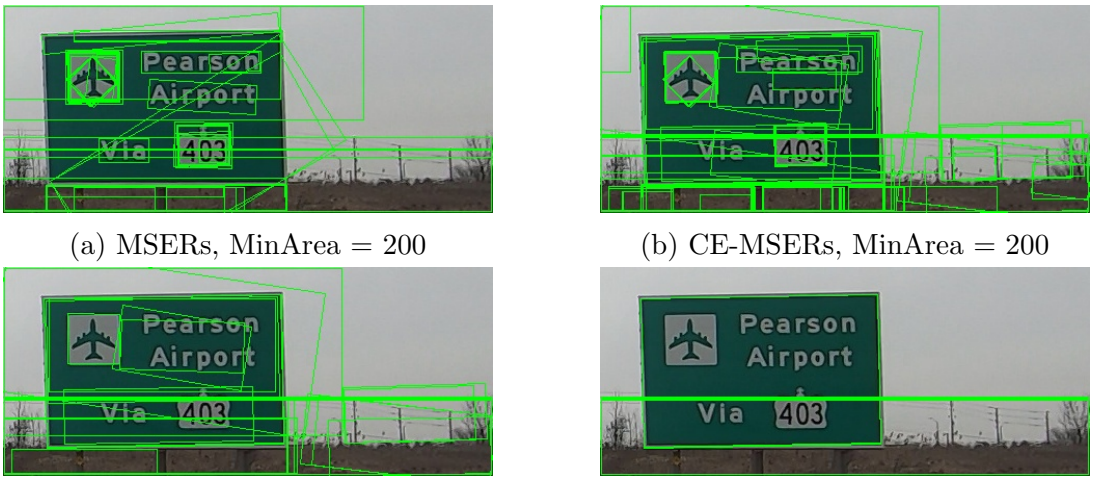
The next step of the proposed algorithm is to detect the text-based traffic sign within the defined search region. We divide the detection into two steps, namely, text-based traffic sign candidates extraction and filtering. Similar to many existing works in scene text detection [47], [60], we use the MSERs algorithm to extract text-based traffic sign candidates. As discussed in [14], MSERs detector is robust to lighting and contrast variations and can detect high-contrast regions, which makes it a powerful and effective method to find the text-based traffic sign regions even in many challenging situations.

However, in the traffic scenario, the performance of the original MSERs detector is reduced due to the following reasons. First, text components extraction by using

MSERs are easily distorted by complex background such as bricks, leaves, which will result in more false detections and increase computational cost in post-processing steps. Second, interference factors such as defocused and motion blurring which common occurrence in traffic circumstance, will destabilize MSERs and make it hard to extract the correct text traffic sign regions. Third, miss detections in this extraction stage are impossible to recover in later steps, which will reduce the performance of defining the search regions in the following frames in the video.

To improve the performance of text-based traffic sign candidate extraction and detect all possible text-based traffic sign regions, a fast and effective contrast-enhanced MSERs detector (CE-MSERs) [82] is used. A contrast-limited adaptive histogram equalization (CLAHE) is first applied to the defined search regions before detecting MSERs. The threshold of MSERs is set to 1 for gaining a higher Recall rate and minimal area size is set for reducing the tiny detected regions, which are false alarms. In general, the CE-MSERs algorithm could handle more conditions compared to the original MSERs algorithm. As shown in Figure 3.7 (a) and (b), with the same minArea parameter setup, CE-MSERs detector is able to detect the true text-based traffic sign region but with more false alarms.

Many other kinds of enhanced MSERs algorithms, such as edge-enhanced MSERs [55] and other variants of contrast-enhanced MSERs [60] could also be an alternative methods in this step. The main reason for using a simple contrast-enhanced MSERs detector is to reduce the computational cost. Merging the detection result of MSERs in other color spaces such as HSV may increase the possibility of detecting the text-based traffic sign regions, but this will also slow down the computational speed, making it hard to be used in real-time applications.



(a) MSERs, MinArea = 200

(b) CE-MSERs, MinArea = 200

(c) CE-MSERs, MinArea = 2000

(d) CE-MSERs, MinArea = 2000, sign size thresholding based on previous frame

Figure 3.7: Illustration of the result of the extracted candidates stage with different algorithm and parameters setup (delta = 1).

Before proceeding to the traffic sign candidate selection stage, we further reduce the total number of detected text-based traffic sign candidates by filtering the non-traffic sign regions using the area size of the detected region. For a forward-moving vehicle, the size of the traffic sign in frame t^{k+1} is in proportional to its previous frame t^k [36]. Using this propriety, we set the threshold of the size of the text-based traffic sign to its possible range depending on the detection results of the previous frame. As shown in Figure 3.7 (c) and (d), the false alarms are decreased with the proper traffic sign size threshold setup. In this Chapter, we set the min and max threshold of the size to 0.8 and 1.2 times the size of the traffic sign in the previous frame. On average, the number of text-based traffic sign candidates is reduced to less than 20 after the filtering.

3.4.3 Text-based traffic sign candidate selection

The targeted text-based traffic sign is within its defined search region. The next stage is selecting one region to be our final predicted traffic sign out of all the extracted candidates and as inputs for the next estimation stage in the next frame. Since the time interval of two consecutive frames is small, the same traffic sign should not be too far apart. The nearest neighbour algorithm is used to select the detected region. Let $p_i^{t_k}$ $(i = 1, 2, 3, 4)$ to be the corner points of the predicted region of frame t_k, and $S = \{\hat{p}_{1_i}^{t_{k+1}}, \hat{p}_{2_i}^{t_{k+1}}, ..., \hat{p}_{n_i}^{t_{k+1}}\}$ (n is number of the candidates) is the set of corner points of the detected candidates region of frame t_{k+1}. For each pair of the corresponding corner points, we calculate the euclidean distance

$$d(p^{t_k}, p_n^{t_{k+1}}) = \left(\sum_{i=1}^{4} (\hat{p}_{n_i}^{t_{k+1}} - p_i^{t_k})^2 \right)^{1/2} \tag{3.4.8}$$

The region which has the minimum euclidean distance is selected to be our final detected text-based traffic sign, as shown in Figure 3.8.

(a) Before selection (b) Final result

Figure 3.8: Illustration of the result of the candidates selection stage.

3.5 Experiments and results

3.5.1 Evaluation Metrics and Datasets

The recent text-based detection algorithm [4] is tested on the Traffic Guide Panel dataset and Text-based Traffic Sign Dataset in Chinese and English (TTSDCE), which are proposed in [4] and [15], respectively. The Traffic Guide Panel dataset is a benchmark dataset containing 3,841 images in total (2,315 images contain highway guide panels and 1,526 contains no traffic signs). Moreover, the TTSDCE is a bilingual text-based traffic sign dataset, which is composed of 1,800 images from the Internet and car camera with resolution ranging from approximately 300 by 300 to 1280 by 720. However, these two datasets only contain images, not video datasets, which make them inappropriate for testing in our proposed method.

To evaluate the effectiveness of the proposed algorithm in real-life scenario, the proposed method is tested on our self-collected the ETFLab Text-based Traffic Sign Video Dataset (ETFLab-TTSVD) [78], which is proposed in Chapter 2. The ETFLab-TTSVD is collected using a fixed camera mounted on the interior of the windshield, captured with 1080p (1920 by 1080) resolution at 24 frames per second rate (fps) for a duration of one second each. There are 30 videos in total and each video contains at least one text-based traffic sign on the highway. The focal length of the camera is fixed at 24 mm. Moreover, the speed of the vehicle is roughly constant in each video.

We measure the accuracy of our proposed algorithm by using the Intersection over Union (IoU) [4], which is commonly used in the area of object detection and

segmentation. The IoU is defined as

$$\text{IoU} = \frac{\text{Intersection}(G, P)}{\text{Union}(G, P)} \tag{3.5.1}$$

where G is the ground truth bounding box and P is the predicted bounding box. If the value of IoU greater than 0.5, it means the predicted bounding box can be considered a true detection, otherwise, the detection is false. Setting the threshold of IoU equals to 0.5, we calculate the Precision, Recall and $\text{F}_{\text{measure}}$, which are defined as [4]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.5.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.5.3}$$

$$\text{F}_{\text{measure}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.5.4}$$

where TP, FP and FN are the true positive, false positive and false negative (miss) rates respectively . Since our method is a one-to-one prediction algorithm, there is no false negative, the value of Recall is constant at 1.0.

3.5.2 Experimental Details and Comparative Analysis

We implemented our proposed method in Python 3.7 language on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM. We compared our proposed algorithm with the localization algorithm, which was proposed in Chapter 2 with an example of a turning case; the IoU of predicted bounding boxes are shown in Figure 3.9. With the knowledge of the speed of the vehicle and roughly turning angle information from

the on-board sensor, when $k = 1$, the localization algorithm can predict the position of the targeted traffic sign in the next 20 frames with the threshold of IoU greater than 0.5, the predicted results are shown in Figure 3.10 (a). However, if a miss detection occurs during the initial detection stage, i.e., $k = 2$, the IoU of predicted bounding boxes degrades fast.

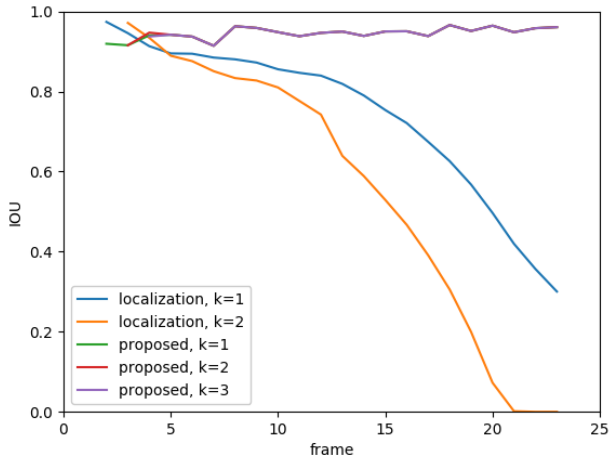
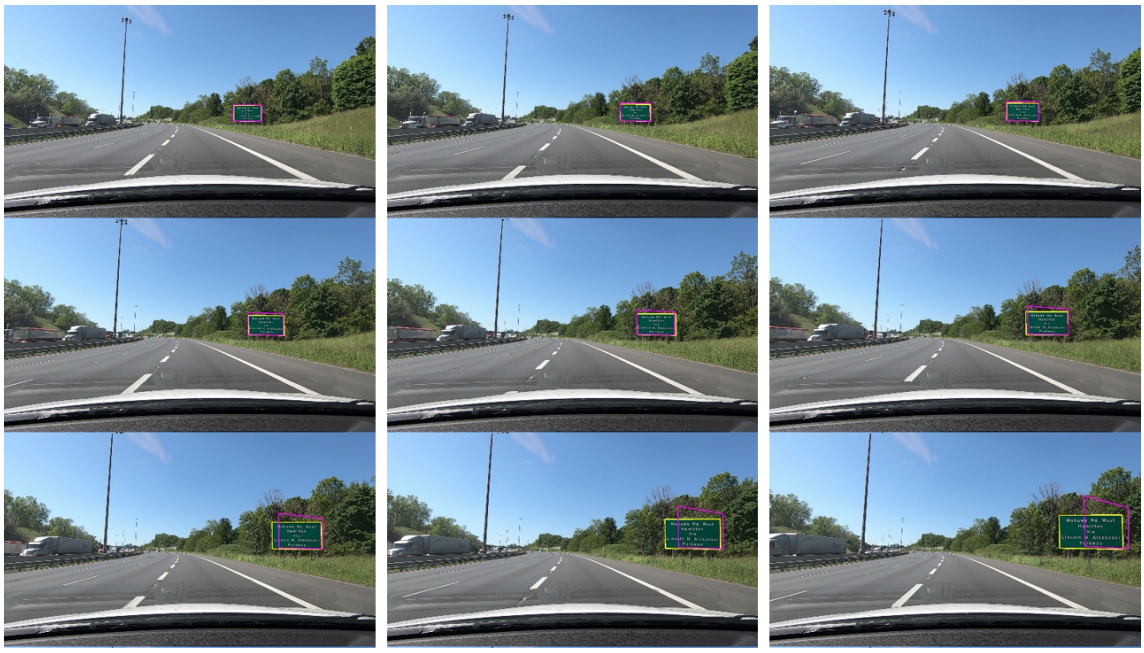


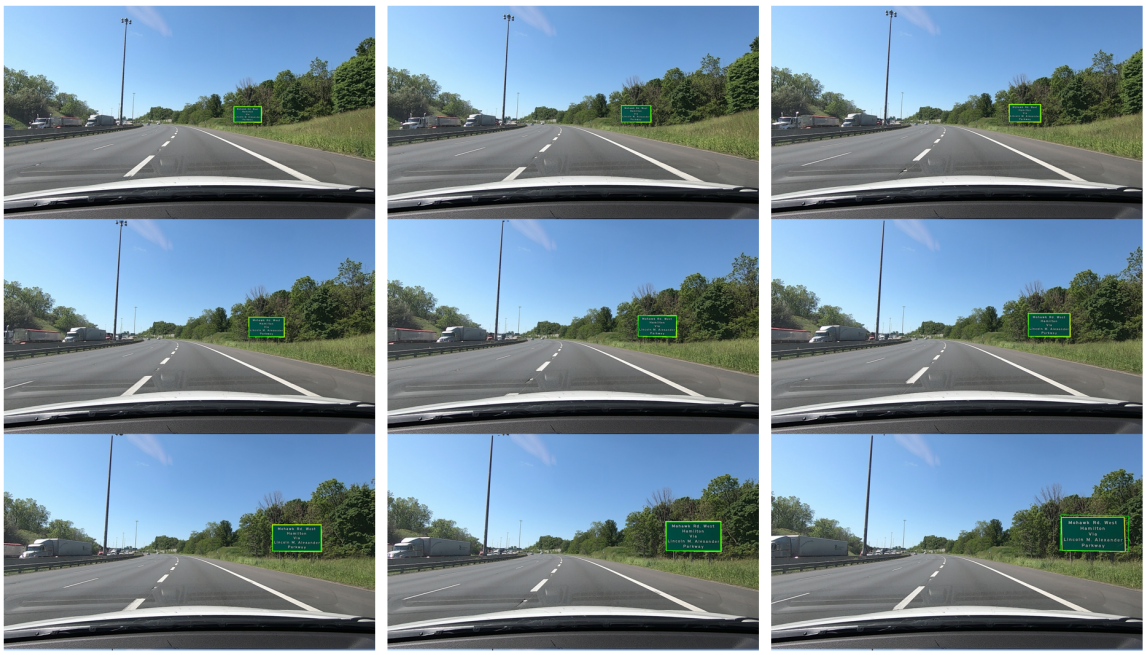
Figure 3.9: The IoU performance of different algorithm in video

Unlike the localization algorithm, the proposed algorithm does not need the turning angle information from the on-board sensor. The range of ideal turning angle is estimated through the speed of the vehicle, with the coefficient of kinetic friction μ_k and the gravitational constant g. In this example, we set $\mu_k = 0.7$ and $g = 9.8\ ms^{-2}$. As shown in Figure 3.9, after relaxing the data acquisition of turning angel and with the help of the text-based traffic region extraction algorithm, the IoU of detected regions have a significant improvement compared to the localization algorithm. Though both of the methods have a good performance initially, the predicted results of the proposed method are more stable later. The miss detection is not a major impact on



(a)

(b)



(c)

Figure 3.10: Comparison of predicted bounding box results with different algorithms when $t = 3, 5, 7, 10, 12, 14, 17, 19, 21$, ground truth, localization and detection bounding boxes are in yellow, magenta and green, respectively. (a) Localization algorithm, $k = 1$ (b) Our proposed, $k = 1$ (c) Our proposed, $k = 2$

the fast degradation of the IoU, the predicted results when $k = 1$ and 2 are shown in Figure 3.10 (b) and (c). In addition, the average of IoU is greater than 0.9.

The running time of the proposed algorithm is increased compared to the localization algorithm. From the experiment, the computational time of defining the search regions only takes less than 0.0005 seconds. The primary computational cost is from the text-based traffic candidates extraction stage. The candidates extraction stage needs 0.033 seconds on average depending on the size of the search regions. Our proposed algorithm is implemented using the Python language and CPU, and faster runtimes can be achieved with C++ and GPU implementations.

We further compared our proposed algorithm with those of existing text-based

Table 3.1: Precision, Recall, $F_{measure}$, and running times of different text-based traffic sign methods. (IoUT = 0.5)

Method	Precision	Recall	$F_{measure}$	Time(s)	Device
Epshtain et al. [46]	0.35	0.41	0.38	2.51	CPU
Gómez and karatzas [45]	0.46	0.53	0.49	1.32	CPU
Jaderberg et al. [53]	0.59	0.71	0.64	4.53	GPU&CPU
Rong et al. [15]	0.73	0.64	0.68	0.16	GPU
Zhu et al. [4]	0.90	0.87	0.88	0.15	GPU&CPU
Our method	**0.95**	**0.97**	**0.96**	**0.033**	CPU

traffic sign methods in Table 3.1. In [4], a cascaded segmentation detection framework for text-based traffic sign detection was presented with significant improvement in Precision, Recall and $F_{measure}$ compared to the other deep learning-based methodologies proposed in [15], [53]. For 24 fps videos, one-second duration, consider the remaining frames after initial detection, the Precision, Recall and $F_{measure}$ values of our prediction are better than those of the state-of-the-art detection results. The computational time with CPU is 0.033s for a 1080p resolution image, which is one-fifth of the fastest existing GPU-based implementation in [4]. This means, with our proposed algorithm, it is possible to implement in a real-time application without

(a)

(b)

Figure 3.11: More text-based traffic sign detection results.

drop the frame rate. Some additional examples are shown in Figure 3.11.

3.5.3 Robustness Analysis and Failure cases

If the defined search region falls to include the true text-based traffic sign region, the post-stage cannot recover it. Thus, we require a high Recall rate in the defined search region. From our experiment, with the high IoU value of detected traffic sign region from the previous frames, the defined search region always contains the true targeted traffic sign, even with the less accuracy of the speed of the vehicle. Figure 3.12 illustrates the difference of results in the defined search region stage with different speeds of the vehicle. The defined search region is able to include the true text-based traffic sign region with an error of the estimated speed of the vehicle within 10%.

The failure cases are mainly caused by the miss detection of the MSERs algorithm and the selection stage based on the nearest neighbour algorithm. As shown in Figure 3.13, the MSERs algorithm detected the true text-based traffic sign region, the region may not be chosen in the candidate selection stage. Though the final selected region

(a) 70km/h

(b) 80km/h

(c) 90km/h

(d) 100km/h

Figure 3.12: Illustrate the defined search regions with different the speed of the vehicle.

is satisfied with the IOU threshold in the current frame, the search region in the following frame may fail to include the true text-based traffic sign region. This failure is due to the nearest neighbour algorithm. To balance the computational time, we use the nearest neighbour algorithm in the selection stage, a fast classifier could be an alternative solution, which we will improve in our future work.

(a)

(b)

Figure 3.13: Failure case sample caused by candidates selection stage. (a) Extracted candidates (b) Final selected region

3.6 Conclusions

In this Chapter, a text-based traffic sign detection algorithm for the video, including a more detailed and accurate search region definition algorithm based on the kinematic automotive model, was presented. The proposed search regions definition algorithm modelled and estimated the search regions of text-based traffic signs mathematically based on the kinematic states of vehicles and the spatial-temporal relationships between the motionless traffic signs and moving vehicles. The potential candidate regions are then extracted within the defined search regions by using a fast and effective CE-MSERs detector. To balance the computational time, a nearest neighbour algorithm is used in the final selection stage. From the experimental results, our proposed method can be used to significantly reduce the computational cost of traffic sign detection while maintaining the high performance of detection results in real-time applications. In the future work, a fast and accurate classifier will be trained and used in the selection stage, and the proposed algorithm will be integrated into a traffic sign detection, recognition and tracking framework.

Chapter 4

A Framework for Text-based Traffic Sign Detection and Tracking in Video

4.1 Abstract

Video based Traffic Sign Recognition (TSR) system is one of the essential module in Advanced Driver Assistance Systems (ADAS). A complete TSR system contains three functionalities, namely, traffic sign detection, tracking and recognition. There are two categories of traffic signs in traffic scenario, namely, graphics-based (symbol-based) traffic signs and text-based traffic signs. Although graphics-based traffic signs have been studies for many years, only a few existing algorithms have focused on text-based traffic signs detection and tracking. Detecting text-based traffic signs is a challenging task, mainly due to the reason of larger variations and limited available dataset. In this Chapter, a text-based traffic signs detection and tracking framework

is proposed. A text-based traffic signs detector is trained during the detection stage on street view images, and a multi-traffic signs tracking algorithm is proposed based on kinematic automotive motion model for video. The framework is evaluated on both public Traffic Guide Panel dataset and our self-collected the ETFLab Text-based Traffic Sign Video Dataset. The overall performance demonstrates the effectiveness of the proposed system, which can be better adapted to real-time applications.

4.2 Introduction

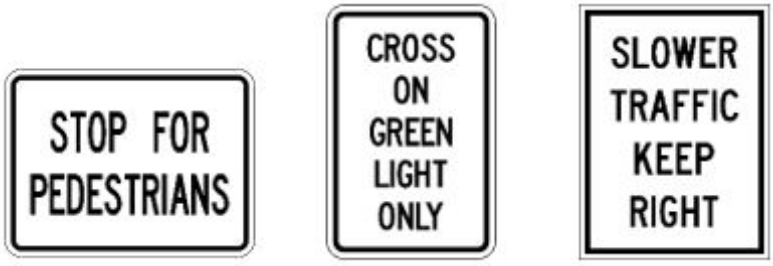
As a crucial component of Intelligent Transportation System (ITS), Advanced Driver Assistance Systems (ADAS) aim to better inform various traffic users and make safer use of transport networks. Among techniques in ADAS, traffic sign recognition (TSR) systems play an important role in autonomous driving [1] and mobile mapping [2]. A complete TSR system contains three stages, namely, traffic sign detection, recognition and tracking. The outputs of traffic sign detection stage yield bounding box level results, which enclose the traffic sign region of interests (ROIs). The functionality of the followed recognition stage is to classify the type of graphics-based traffic signs or read the characters on text-based traffic signs ROIs. Traffic sign tracking is vital especially in video-based TSR system, because the detected traffic signs in the previous frame are tracked over the following image frames and the cost of repetitive recognition on the same traffic sign are reduced.

Traffic signs can be generally categorized into two groups, namely, graphics-based (symbol-based) and text-based traffic signs. As shown in Figure 4.1 (a) and (b), graphics-based traffic signs usually provide environmental information such as stop

sign, yield sign, and construction sign, text-based traffic signs contain semantic information to guide drivers towards their destination. Though traffic signs shown in Figure 4.1 (c) have text on them, they are still categorized into the graphics-based traffic signs for the reason that the text on each of them are same, which can be thought of as a typical graphic. The main difference between graphics-based and text-based traffic signs is the variation in appearance. Graphics-based traffic signs often have fixed shape, uniform graphical appearance and distinct color. However, it is difficult to find two identical text-based traffic signs on the road.

(a) Graphics-based traffic signs

(b) Text-based traffic signs



(c) Traffic signs with text

Figure 4.1: Different categories of traffic signs.

4.2.1 Text-based traffic sign detection

Text-based traffic sign detection is a challenging task mainly due to three aspects, namely, variations in text content (i.e. font, color, size, stroke width and multilingual nature), complexity of the background (i.e. tree leaves and building bricks) and interference factors during the image acquisition (i.e. uneven lighting, perspective distortion, defocused or motion-blur imagery, multi-orientation due to motion and partial or complete occlusion). As the first step of TSR system, a effective and accurate traffic signs detection functionality is essential and has significant impacts on the performance of subsequent recognition and tracking functionality. Poor performance of detected text-based traffic sign bounding box areas and slow detection functionality running time will result in missing characters in the post recognition stage and low frame per second (fps) in real time application. In recent years, many existing algorithms have focused on graphics-based traffic sign detection and achieved promising results on many public datasets, such as the MASTIF dataset [8] and the German Traffic Sign Detection Benchmark (GTSDB) [12]. However, only few research studies have focused on text-based traffic sign detection. As described in [4], this maybe partially caused by the difficulties of the task itself [14] and insufficient public datasets. Currently, the Traffic Guide Panel dataset proposed in [15] is the only (partially) public text-based traffic sign dataset benchmark, which is an image dataset, not a video.

The studies have focused on the detection of text-based traffic signs is limited. Inspired by road sign detection in color video sequence [36], a algorithm was proposed in [37], to detect text-based traffic signs in videos. The algorithm assumes that the traffic sign is on a planar surface perpendicular to the horizontal ground and that the

camera moves along its optical axis that is roughly horizontal. Then, the geometric relationship between a moving car and traffic signs is defined based on a pinhole camera model. After that, the orientation of the plane is estimated using three or more points in two consecutive frames. A multiscale text detection algorithm is proposed on each candidate traffic panel area using edge detection, adaptive search, Gaussian Mixture Model (GMM) and geometric linear analysis to obtain the position of a text line and to track it with a feature-based tracking algorithm.

Focusing on the distinct background color of text-based traffic signs, in [38], the blue and white rectangular regions of interests (ROIs) were extracted using a color-based segmentation algorithm and Fast Fourier Transform (FFT), then the four corner points of the rectangular regions were reoriented horizontally to align text characters. After analyzing the chrominance and luminance, an adaptive segmentation is carried out, and connected components labeling and position clustering are done for the arrangement of the different characters on the panel.

By utilizing spatial information, the search regions of text-based traffic sign can be limited in urban scenario. In [39], the traffic signs that are located above ground and on the right side of the road are separated into two independent ROIs. Then, the blue and white traffic panel regions were extracted for every single image based on color segmentation and Bag of Visual Words (BOVW) approach [40]. After that, the ROIs are classified by using support vector machines [33] or Naïve Bayes [41] classifiers. In [14], the search areas of the traffic signs are restricted first by using a pinhole camera model in the detection stage. Then, the potential text-based traffic sign candidates are detected based on the combination of Maximally Stable Extremal Regions (MSERs) [30] detector and HSV color thresholding in the defined search region. Finally, the

false positive regions are eliminated with the temporal and structural information.

increasingly, deep learning-based object detection methods are becoming main stream in research [58], [83]. Many deep learning-based text-based traffic sign detection algorithms have been proposed and achieved promising results. Inspired by You Only Look Once (YOLOv1) [84] object detector, in [15], a Cascaded Localization Network (CLN) is proposed to find all text-traffic sign candidates with a high Recall rate in the first stage. Then, locate text regions and eliminate the false alarms, including the non-panel and redundant detections in the second stage. In [42], the traffic sign ROIs, including both graphics-based and text-based traffic signs, are first extracted using MSERs algorithms in gray and normalized RGB channels. And then the ROIs are classified into different classes by their proposed multi-task convolutional neural network. In [4], a text-based traffic signs detection algorithm is proposed based on cascaded segmentation detection networks, which can achieve the state-of-the-art 0.9 Precision with a computational speed of 0.15 seconds per frame on Traffic Guide Panel dataset.

4.2.2 Traffic sign tracking

Traffic sign tracking stage is vital especially for video-based TSR systems, which provide more valuable information than detecting signs in single image. Compare to traffic sign detection and recognition, only a few approaches have been studied in traffic sign tracking. In [63], a road sign tracking method is proposed by using continuous adaptive mean shift (cam-shift) method. In [64], a detected traffic sign is tracked using a simple motion model and temporal information propagation. Then, the results of the individual frame are fused for more robust detection. The works in

[36], [65], [66] track the detected traffic signs by using a kalman filter [67], a credible result is obtained by deleting the detections that cannot be identified for consecutive frames. The kalman filter based tracker is used to reduce the computational load in detection stage and fuse the results from consecutive frames to get better classification performance in [36], and improve a pre-trained off-line trained detector with an on-line updated detector in [68]. In [69], a Tracking-Learning-Detection (TLD) framework is adopted to track the recognized signs in real time to provide enough information for driver assistance function.

A good traffic sign tracking algorithm should have the following aspects. First, the tracker can track the detected traffic sign with only one detection (i.e. when the object is initially detected). Second, the running time of tracking algorithm should be fast enough to achieve real time application. Third, the algorithm should be able to handle miss detection in between frames and robust to occlusion. Forth, the tracker should be able to delete the traffic sign when it moves outside the field of view (FOV).

In this Chapter, in order to better adapt to real-time application, a text-based traffic sign detection and tracking framework is proposed for video-based TSR system. The main contribution of our work are summarized as below: 1) A fast and accurate text-based traffic sign detector is presented. 2) A new text-based traffic sign tracking algorithm is proposed based on kinematic automotive motion model, the tracker is robust to occlusion and miss detection in between the frames and can handle the traffic sign when moves out of FOV. 3) The proposed framework yields highly accurate detection results and fast computational speed on public Traffic Guide Panel dataset and our self-collected the ETFLab Text-based Traffic Sign Video Dataset. 4) A new dataset of text-based traffic signs in urban scenario is collected. From the experiential

detection results on real life video data, the proposed algorithm has achieved overall Precision, Recall and $F_{measure}$ of 0.99, 0.78 and 0.88 on the Traffic Guide Panel dataset, respectively. The computational time of our approach is 0.015 second per frame for a 1080p video on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM and a NVIDIA GeForce GTX 1080 Ti GPU with 11 GB RAM. The proposed can be further extended to multilingual environment and applied on system on chips (SoCs).

The structure of the Chapter is as follows. The proposed text-based traffic sign detection and tracking framework is detailed in Section 4.3. Experimental and evaluation results are presented in Section 4.4. Finally, conclusions are discussed in Section 4.5.

4.3 Proposed framework

In this Chapter, a text-based traffic sign detection and tracking framework is proposed for video-based TSR systems. The flowchart of the proposed framework is shown in Figure 4.2. A fast and accurate text-based traffic sign detector will be applied to the incoming frame first. Then, the proposed tracking algorithm will be used. There are five phases during the tracking functionality, namely, update, delete, predict, add and estimate. The newly detected traffic signs will be associated to the traffic signs in the known list based on the estimated position of center and size of the traffic signs from previous frame. Their new position and size will be updated. For the rest unassociated traffic signs in the known list, their unique ID will be deleted if they move out of FOV or max disappear time is reached, otherwise their position and size will be predicted. For the rest unassociated newly detected traffic signs, they will be added to known list and a unique ID will be assigned. After that, for

all traffic signs in the known list, their estimated position and size will be calculated for next frame based on kinematic automotive motion model. Due to the limitation of available text-based traffic sign dataset, in this Chapter, we collect a text-based traffic sign dataset for training process from the internet. The details of training set will be introduced in Section 4.4.1.

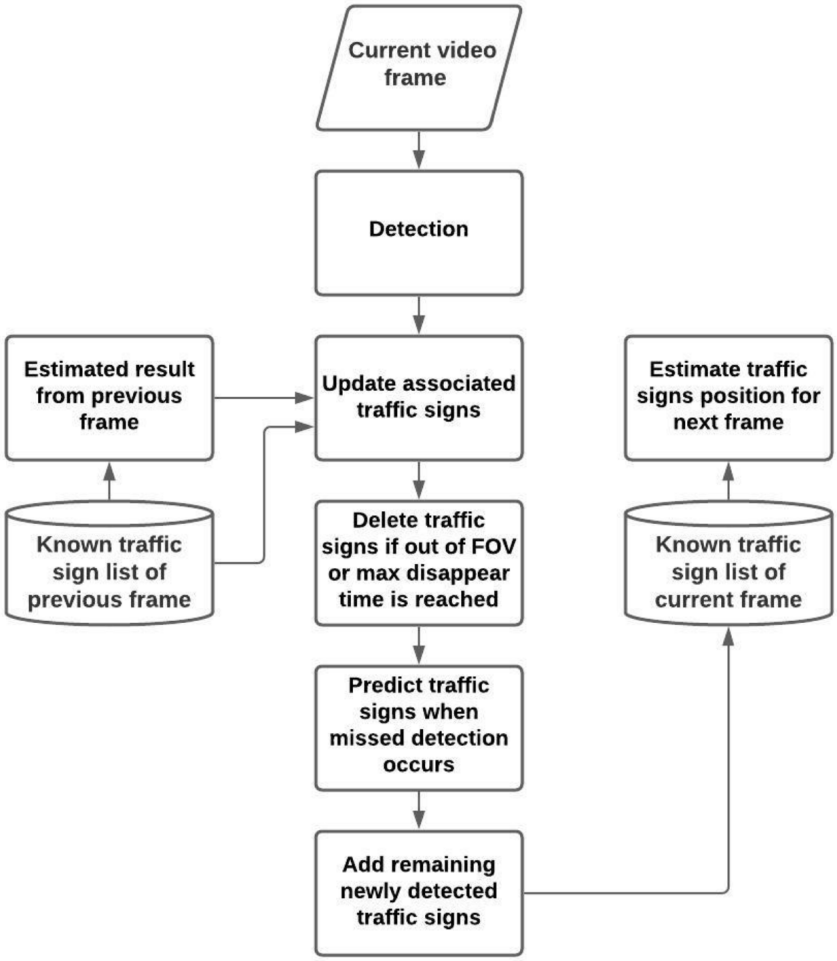


Figure 4.2: Flowchart of the proposed text-based traffic sign detection and tracking framework

4.3.1 Text-based traffic signs detection

In this section, the detection stage of the text-based traffic signs is presented. To satisfy the requirement of real time application as well as to maintain the high detection accuracy, a text-based traffic sign detector is trained based on You Only Look Once version 3 (YOLOv3) [85] object detector.

The detector will divide the input image into a number of $N*N$ grid cells, in which each cells can predict a fixed number of bounding boxes. The output of the detection results is represented by coordinates of bounding boxes along with its objectness score and C class confidences. The coordinates of bounding boxes will be represented using 4 parameters $[x, y, w, h]$, where $[x, y, w, h]$ are the coordinates of the detected bounding box. The (x, y) coordinates represent the center of the bounding box and w and h denote the width and height of the bounding box respectively. w and h define the size of bounding box, which enclose the text-based traffic sign. Unlike YOLOv1, YOLOv3 predicts the objectness score of each bounding box using logistic regression. The final output of YOLOv3 is a $N*N*[B*(4+1+C)]$ tensor, where B is the number of bounding boxes can be predicted in each grid cell, C is the number of class, 4 is bounding box offsets and 1 stands for number of objectness prediction. During the training process, we set $B = 3, C = 1$ and fine-tune the YOLOv3 pretrained model on our newly collected text-based traffic sign training dataset including the ground truth annotation for the text-based traffic signs in green and blue background. The detection results on Traffic Guide Panel dataset are illustrated in Figure 4.3.

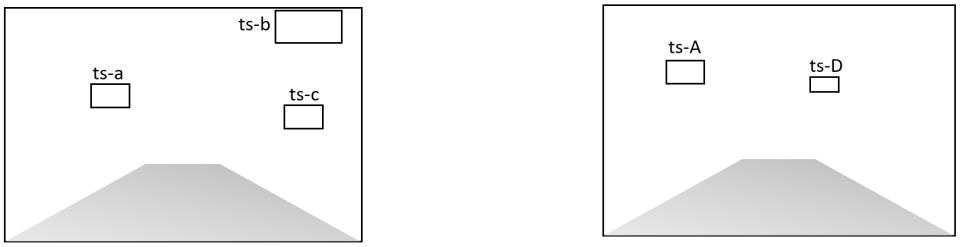
(a)

(b)

Figure 4.3: Illustration of the output bounding box level results in detection stage on Traffic Guide Panel dataset ($N = 8$).

4.3.2 Text-based traffic signs tracking using kinematic automotive motion model

In this section, we introduce a fast and accurate multiple traffic signs tracking algorithm based on the kinematic automotive motion model, which was proposed in Chapter 2. The kinematic automotive motion model details the spatial-temporal relationship between traffic signs and different kinematic vehicles motion status, which will be reviewed later in this section.



(a) Previous frame (b) Current frame

Figure 4.4: Illustration of the tracking mechanism on two consecutive frames. (Shady areas represent road.)

The proposed tracking algorithm associates newly detected traffic signs based on the estimated position of center and size of the traffic signs from previous frame. As it is shown in Figure 4.4 (a), three traffic signs, namely ts-a, ts-b and ts-c are detected at previous frame. Their corresponding position and size are estimated for current frame by using kinematic automotive motion model. Referring to Figure 4.4 (b), two traffic signs, namely ts-A, and ts-D are detected at current frame. Based on the estimation results from previous frame and detection result at current frame, ts-A will be associated with ts-a and its position and size will be updated; ts-b will be deleted, because it moves out of the FOV; the position of ts-c will be predicted (or delete if max disappear time reaches), because the detection of ts-c is missed at current frame; ts-d will be added as a new traffic sign. The details of the algorithm will be explained later in this section.

Kinematic automotive motion model

The physical relationship between a forward moving car and a motionless traffic sign at time t is shown in Figure 4.6, the notations are defined as follows.

- $\boldsymbol{Z}$ - optical axis, which is assumed parallel to the ground. The camera moves

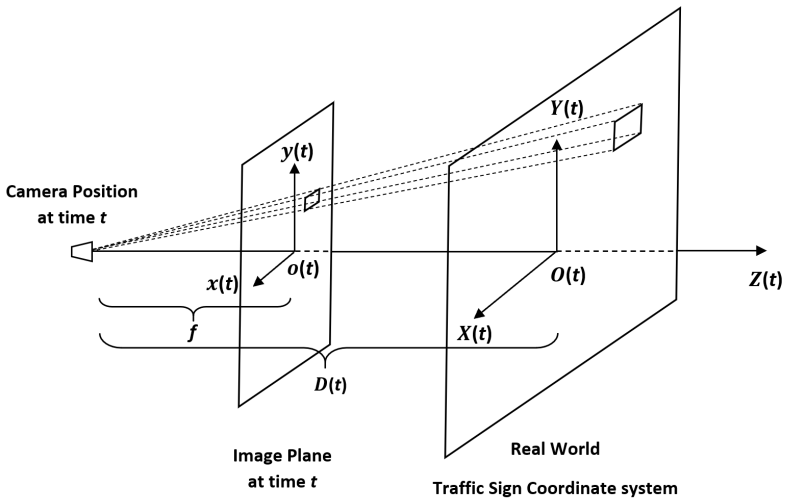


Figure 4.5: Illustration of the physical relationship between a forward moving car and a motionless traffic sign at time t.

along side its optical axis. The direction of $\boldsymbol{Z}$ may different at each time.

- $\boldsymbol{oxy}$ - 2-Dimension camera image plane, which is projected by real world coordinate system $\boldsymbol{OXYZ}$.
  - $\boldsymbol{x}$ - horizontal axis of image plane.
  - $\boldsymbol{y}$ - vertical axis of image plane.
  - $\boldsymbol{o}$ - intersection of optical axis $\boldsymbol{Z}$ and $\boldsymbol{x}$-$\boldsymbol{y}$ plane.
  - $\boldsymbol{o(t)x(t)y(t)}$ - the camera image plane at time t.
- $\boldsymbol{OXYZ}$ - 3-Dimension traffic sign coordinate system.
  - $\boldsymbol{X}$ - horizontal axis, which is parallel to the ground and perpendicular to optical axis $\boldsymbol{Z}$.
  - $\boldsymbol{Y}$ - vertical axis, which is perpendicular to the ground.
  - $\boldsymbol{O}$ - intersection of optical axis $\boldsymbol{Z}$ and $\boldsymbol{X}$-$\boldsymbol{Y}$ plane.

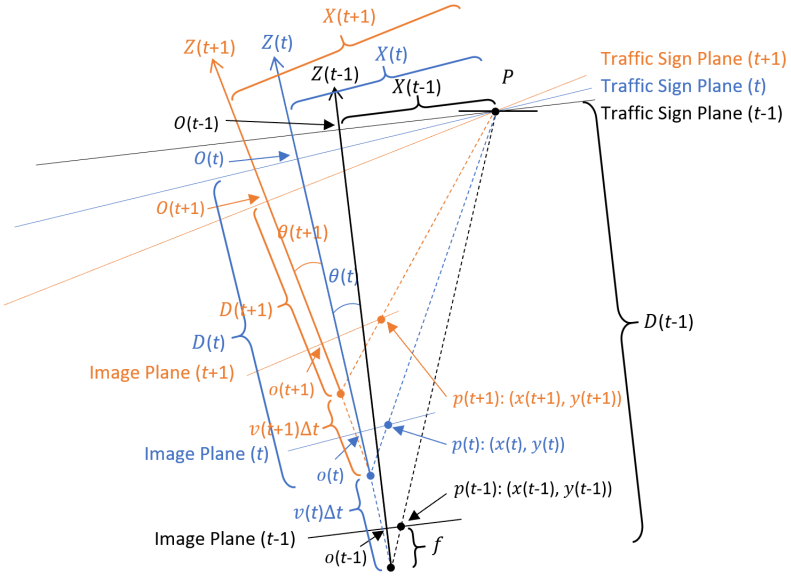


Figure 4.6: Illustration of the spatial-temporal relationships between the motionless traffic signs and moving vehicles at time $t-1$, t and $t+1$.

- $\boldsymbol{O(t)X(t)Y(t)}$ - the assumed traffic sign plane, which always parallels to the camera image plane $\boldsymbol{o(t)x(t)y(t)}$ at time t.

- f - focal length, camera predetermined parameter.

- D - Distance between traffic sign and camera along optical axis $\boldsymbol{Z}$.

The kinematic status of a forward moving vehicle can be roughly summarized in 3 categories, namely, straight line motion, line change and turning. The major difference between a straight line motion case and latter two cases are the directional changes of the optical axis. For a video-based TSR system, we define that the camera is the origin of the 3D real world coordinate system of each time step. The geometric correlation between a moving car and a motionless text-based traffic sign at time $t-1$, t and $t+1$ is shown in Figure 4.6. The real world traffic sign may not always

be parallel to the camera image plane. If and only if the real world traffic sign lies on planar surfaces and the optical axis is perpendicular to the traffic sign, the plane of the traffic and the assumed traffic sign plane will be the same plane. In addition, the image plane of time $t-1$, t and $t+1$ may not always be parallel to each other because of the directional changes of the optical axis. Denote by $\theta(t)$, the horizontal directional difference of optical axis between time $t-1$ and t. We define positive value for left and negative for right. The moving distance between time $t-1$ and t is define as $v(t)\Delta t$.

Position prediction

Let P be an arbitrate point on a text-based road sign in the real world traffic sign coordinate system $OXYZ$. Since the position of the camera is varying with the time, the coordinates of P is changing. Denoted by $P(t) : (X(t), Y(t), D(t))$ and $p(t) : (x(t), y(t))$, the coordinates of P in the real world traffic sign system and its corresponding projected point p on the camera image plane at time t, respectively. Their physical relationship is determined by a pinhole camera model, which are equation (4.3.1), (4.3.2), (4.3.3)

$$\frac{X(t-1)}{x(t-1)} = \frac{Y(t-1)}{y(t-1)} = \frac{D(t-1)}{f} \tag{4.3.1}$$

$$\frac{X(t)}{x(t)} = \frac{Y(t)}{y(t)} = \frac{D(t)}{f} \tag{4.3.2}$$

$$\frac{X(t+1)}{x(t+1)} = \frac{Y(t+1)}{y(t+1)} = \frac{D(t+1)}{f} \tag{4.3.3}$$

$D(t-1)$, $D(t)$ and $D(t+1)$ are the distances between the traffic sign and the

camera alongside the optical axis at time $t-1$, t and $t+1$, respectively. Referring to Figure 4.6, we have

$$D(t-1) = D(t) + v(t)\Delta t \tag{4.3.4}$$

$$D(t) = D(t+1) + v(t+1)\Delta t \tag{4.3.5}$$

By assuming that the road is roughly horizon, the change of optical axis will not affect the vertical axis, we have

$$Y(t-1) = Y(t) = Y(t+1) \tag{4.3.6}$$

To predict the position of point P on the image plane for time $t+1$, we need to know $X(t), Y(t)$ and $D(t)$ first, from (4.3.1), (4.3.2) and (4.3.4), we obtain

$$D(t) = D(t-1)cos\theta(t) - X(t-1)sin\theta(t) - v(t)\Delta t \tag{4.3.7}$$

$$X(t-1) = X(t)cos\theta(t) - (D(t) + v(t)\Delta t)sin\theta(t) \tag{4.3.8}$$

substitute equation (4.3.1), (4.3.2) and (4.3.8) into (4.3.7), we get

$$D(t) = \frac{v(t)\Delta t \cdot (-1 - \frac{f \sin\theta(t)\cos\theta(t)}{x(t-1)} + \sin^2\theta(t))}{1 - (\frac{x(t)\cos^2\theta(t)}{x(t-1)} - \frac{f\sin\theta(t)\cos\theta(t)}{x(t-1)} - \frac{x(t)\sin\theta(t)\cos\theta(t)}{f} + \sin^2\theta(t))} \tag{4.3.9}$$

Substitute (4.3.8) and (4.3.9) into the (4.3.2) we get the estimate coordinate of P at time t

$$X(t) = \frac{v(t)\Delta t \cdot (-1 - \frac{f \sin\theta(t)\cos\theta(t)}{x(t-1)} + \sin^2\theta(t))}{1 - (\frac{x(t)\cos^2\theta(t)}{x(t-1)} - \frac{f\sin\theta(t)\cos\theta(t)}{x(t-1)} - \frac{x(t)\sin\theta(t)\cos\theta(t)}{f} + \sin^2\theta(t))} \cdot \frac{x(t)}{f} \quad (4.3.10)$$

$$Y(t) = \frac{v(t)\Delta t \cdot (-1 - \frac{f \sin\theta(t)\cos\theta(t)}{x(t-1)} + \sin^2\theta(t))}{1 - (\frac{x(t)\cos^2\theta(t)}{x(t-1)} - \frac{f\sin\theta(t)\cos\theta(t)}{x(t-1)} - \frac{x(t)\sin\theta(t)\cos\theta(t)}{f} + \sin^2\theta(t))} \cdot \frac{y(t)}{f} \quad (4.3.11)$$

From equation (4.3.2), (4.3.3) and (4.3.5), we obtain

$$D(t+1) = D(t)\cos\theta(t+1) - X(t)\sin\theta(t+1) - v(t+1)\Delta t \quad (4.3.12)$$

$$X(t+1) = (D(t+1) + v(t+1)\Delta t)\tan\theta(t+1) + \frac{X(t)}{\cos\theta(t+1)} \quad (4.3.13)$$

Substitute (4.3.6), (4.3.12), (4.3.13) into the (4.3.3) we get

$$x(t+1) = \frac{(D(t)\sin\theta(t+1) - X(t)\sin\theta(t+1)\tan\theta(t+1) + \frac{X(t)}{\cos\theta(t+1)})f}{D(t)\cos\theta(t+1) - X(t)\sin\theta(t+1) - v(t+1)\Delta t} \quad (4.3.14)$$

$$y(t+1) = \frac{Y(t)f}{D(t)\cos\theta(t+1) - X(t)\sin\theta(t+1) - v(t+1)\Delta t} \quad (4.3.15)$$

Using (4.3.9), (4.3.10), (4.3.11), (4.3.14) and (4.3.15), we can predict the coordinate of point p on the image plane at time $t+1$.

Size prediction

Let $W(t)$ and $H(t)$ be the width and height of the traffic sign on the traffic sign plane at time t, respectively. $W(t)$ and $H(t)$ equal to the actual traffic sign size W and H if the optical axis Z is perpendicular to the actual traffic sign at time t. Then, we

have

$$W(t) \leq W \tag{4.3.16}$$

$$H(t) \leq H \tag{4.3.17}$$

Denoted by $w(t)$ and $h(t)$, the width and height of the detected traffic sign at time t, respectively. From (4.3.2), (4.3.3) and (4.3.5), we have

$$\frac{W(t)}{w(t)} = \frac{H(t)}{h(t)} = \frac{D(t)}{f} \tag{4.3.18}$$

$$\frac{W(t+1)}{w(t+1)} = \frac{H(t+1)}{h(t+1)} = \frac{D(t+1)}{f} = \frac{D(t) - v(t+1)\Delta t}{f} \tag{4.3.19}$$

Traffic signs in the real world are not always perpendicular to the optical axis. Referring to Figure 4.7, the size of traffic sign in next frame will also rely on the difference of angle between traffic sign and optical axis in current frame.

$$W(t) = W\cos\alpha, \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \tag{4.3.20}$$

$$W(t+1) = W\cos(\alpha - \theta), \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \tag{4.3.21}$$

Denoted by α, the included angle between traffic sign and the assumed traffic sign plane at time $t-1$ ($-\frac{\pi}{2} < \alpha < \frac{\pi}{2}$). When moving in straight line, the value of α will not affect the changing of size. While when tuning left, if $\alpha < 0$, the size of traffic sign may decrease in the following frames, as shown in Figure 4.7 (c).

Eliminate W in (4.3.20) and (4.3.21), we have

$$W(t+1) = W(t)\frac{\cos(\alpha-\theta)}{\cos\alpha} \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \tag{4.3.22}$$

From (4.3.18), (4.3.19) and (4.3.22), we have

$$w(t+1) = \frac{D(t)w(t)}{D(t) - v(t+1)\Delta t} \cdot \frac{\cos(\alpha-\theta)}{\cos\alpha}, \quad -\frac{\pi}{2} < \alpha < \frac{\pi}{2} \tag{4.3.23}$$

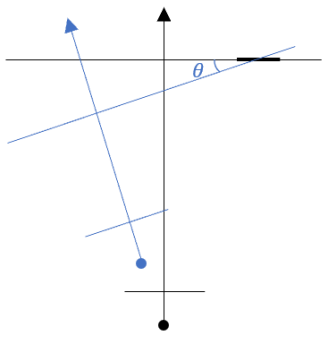
α is an unknown parameter and is hard to estimate. Based on observation, we assume $|\alpha| < \pi/4$, the value of $\cos(\alpha-\theta)/\cos\alpha$ is bounded by $3/2$. In this way, we are able to predict the range of width for the next frame.

From (4.3.6), we have

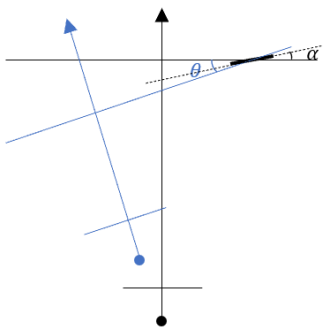
$$H(t) = H(t+1) \tag{4.3.24}$$

With (4.3.18) and (4.3.19), we get

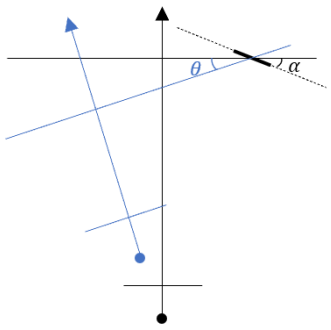
$$h(t+1) = \frac{D(t)h(t)}{D(t) - v(t+1)\Delta t} \tag{4.3.25}$$



(a) $\alpha = 0$



(b) $\alpha > 0$



(c) $\alpha < 0$

Figure 4.7: Illustration of the size changes in next frame based on different value of α in current frame. (Current frame in black line and next frame in blue line.)

For a detected traffic sign TS at time t, we have

$$TS(t) = \Big[x(t-1), y(t-1), w(t-1), h(t-1), x(t), y(t), w(t), h(t)\Big] \tag{4.3.26}$$

if TS is not in the known list, it will be initialized as

$$TS(t) = \Big[x(t), y(t), w(t), h(t), x(t), y(t), w(t), h(t)\Big] \tag{4.3.27}$$

Using (4.3.9), (4.3.23) and (4.3.25), we can estimate the range of width and the value of height for the next frame.

Estimation of parameters

If we have the velocity and directional information of the vehicle from on-board sensors at each time step, we can use the (4.3.9), (4.3.10), (4.3.11), (4.3.14), (4.3.15), (4.3.23) and (4.3.25) to predict the position of the center point, size of the targeted text-based traffic sign. However, the above information may not always obtainable in

some situations, if we do not have velocity and directional information, we can still predict the range of them by using physical rules. When negotiating a curve safely, the maximum turning angle is in proportion to the vehicle's speed. By using the range of velocity and turning angle, we can estimate positions of the center points and size of the targeted traffic sign with different pairs of velocity and turning angle, and thus narrow down the search region.

From Chapter 3, we have that

$$r \geq \frac{v^2}{\mu_k g} \tag{4.3.28}$$

where r stands for the turning radius, μ_k is the coefficient of kinetic friction, g is the gravitational constant.

The value of μ_k only depends on the two contacted surface and varies between two surfaces. From the coefficient of friction table [80], we learned that the kinetic frictional coefficient of rubber on concrete is $0.6-0.85$ on a sunny day and $0.45-0.75$ on a rainy day. The value of turning radius r should be greater than its manufactured minimum turning radius, and this value is variant based on different vehicle's brand. For example, the turning circle of a year 2020 Toyota Corolla is 5.3 meter [81].

From the equation of angular velocity, we have

$$\omega = \frac{\Delta\theta}{\Delta t} = \frac{v}{r} \tag{4.3.29}$$

$\Delta\theta$ is the directional difference of the turning angle in given sampling time interval Δt. Substitute (4.3.28) into (4.3.29), we get our estimated ideal turning angle, which

is

$$|\theta| = \frac{vt}{r} \leq \frac{\mu_k g t}{v} \; radians \tag{4.3.30}$$

Assume V_{max} is the maximum velocity of a vehicle driving on the road, for every $v \in [0, V_{max}]$, we can calculate its corresponding range of θ. For each different pair of (v, θ), we will have a predicted center, width and height. Using extreme points of predicted center, width and height, a search region of the target traffic sign will be formulated.

Algorithm description

Let $\{TS_1(t), TS_2(t), ..., TS_n(t)\}$ be the known traffic sign list at time t, (*i.e.* $TS_i(t) = [x_i(t), y_i(t), w_i(t), h_i(t)]$) their corresponding search regions for time $t+1$ are $\{SR_1(t), SR_2(t), ..., SR_n(t)\}$. The bounding box of detected traffic sign ts_k at time $t+1$ are presented as $[x_k(t+1), y_k(t+1), w_k(t+1), h_k(t+1)]$.

The proposed multi-traffic signs tracking algorithm has 5 phases: update, delete, predict, add and estimate.

- Update: Associate and update the traffic signs in the known list for $TS_n(t)$, if $(x_k(t+1), y_k(t+1))$ is in its search region $SR_n(t)$, and the distance between $(x_n(t), y_n(t))$ and $(x_k(t+1), y_k(t+1))$ is the shortest and $w_k(t+1)$ and $h_k(t+1)$ is within the range of estimated width and height. Set their disappeared time to 0.

- Delete: For the rest unassociated traffic signs in the known list, delete the traffic sign $TS_m(t)$ from the known list, if predicted position is out of FOV or max

disappeared time reaches.

- Predict: For the rest undeleted and unassociated traffic signs in the known list, predict their bounding box using $v = V_{max}/2$ and $\theta = 0$, then increase its disappeared time by 1.

- Add: For the rest of unassociated detected traffic sign, add them to the known list and assign a unique ID.

- Estimate: For all the traffic signs $TS_i(t+1)$ in the known list, estimate their corresponding search region $SR_i(t+1)$ using kinematic automotive motion model.

4.4 Experiments and results

4.4.1 Data Preparation for Training

To train a reliable text-based traffic sign detector based on YOLOv3 model, as much labelled data as possible is required in each different categories of traffic signs. The recent text-based detection algorithms in [15] and [4] presented their self-collected Traffic Guide Panel dataset and Text-based Traffic Sign Dataset in Chinese and English (TTSDCE), respectively. The Traffic Guide Panel dataset is a benchmark dataset containing $3,841$ images in total ($2,315$ images contain highway guide panels and $1,526$ contains no traffic signs). And the TTSDCE is a bilingual text-based traffic sign dataset, which is composed of $1,800$ images from Internet and car camera with resolution ranging from approximately $300 * 300$ to $1,280 * 720$. However, these two works did not release their datasets for training.

It is time consuming and expensive to collect training data by driving a car with a

car-mounted camera on the highway. In [42] and [86], the images from Google Street View were used to help develop the vision-based driver assistance system. Due to the limitation of available text-based traffic sign training dataset, we collect highway view images from Google Street View as training data for our text-based traffic sign detection system. Unlike graphics-based traffic sign, there are larger within-class variations in text-based traffic signs. Therefore, in this Chapter, we focused on the traffic signs majorly in green and blue background. The self-collected text-based traffic sign dataset has 810 highway and street scene images with resolution approximately $1,323 * 965$. All the acquired images contain at least one text-based traffic sign and then were labelled based on the following principle. First, the text-based traffic sign was not labelled if it is too far to recognize the characters on it. Second, every traffic sign was labelled with information such as location, size and class. Some examples of annotated training data are shown in Figure 4.8.

4.4.2 Evaluation Metrics and Testing Datasets

To evaluate the effectiveness of the proposed algorithm in real-life scenario, the proposed method is tested on both Traffic Guide Panel dataset and our self-collected the ETFLab Text-based Traffic Sign Video Dataset (ETFLab-TTSVD) [78], which is proposed in Chapter 2. The testing data of Traffic Guide Panel dataset contains 404 images. The ETFLab-TTSVD is collected using a fixed camera mounted on the interior of the windshield, captured with $1,080$p $(1,920 * 1,080)$ resolution at 24 frame per second rate (fps) for a duration of one second each. There are 10 random selected videos from highway scene, 2400 images in total. The focal length of the camera is fixed at 24 mm. And the speed of the vehicle is roughly constant in each video.

(a)

(b)

Figure 4.8: Examples of annotated Google street view images.

The accuracy of our proposed framework is measured by using the Intersection over Union (IoU) [4], which is commonly used in the area of object detection and segmentation. The IoU is defined as

$$\text{IoU} = \frac{\text{Intersection}(G, P)}{\text{Union}(G, P)} \tag{4.4.1}$$

where G is the ground truth bounding box and P is the predicted bounding box. If the value of IoU is greater than 0.5, it means the predicted bounding box can be considered as true detection, otherwise, the detection is false. From experimental results, if the value of IoU less than 0.75, some characters on the traffic sign may be lost, make it harder to be used in post recognition stage.

We calculate the Precision, Recall and $\text{F}_{\text{measure}}$ by setting the threshold of IoU equals to 0.5, which are defined as [4]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.4.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.4.3}$$

$$\text{F}_{\text{measure}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4.4}$$

where TP, FP and FN are the true positive, false positive (false detection) and false negative (miss) rates respectively .

4.4.3 Experimental Details and Comparative Analysis

During the training process, the initial learning rate is set to 0.001, weight decay = 0.0005 and momentum = 0.9. The training process stops at $4,000$ iterations. During

(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.9: Examples of detection results on Traffic Guide Panel dataset ((a)-(d)) and ETFLab-TTSVD ((e),(f)).

the testing process , the images are resized to $256 * 256$ (i.e. $N = 8$) when testing on the Traffic Guide Panel dataset. All experiments are implemented on a PC with i7-7700K CPU running at 4.20 GHz with 16 GB RAM and a NVIDIA GeForce GTX 1080 Ti GPU with 11 GB RAM. Some example are shown in Figure 4.9 (a)-(d).

We compare our detection results with existing text-based traffic sign methods in Table 4.1. In [4], a cascaded segmentation detection framework for text-based traffic sign detection was presented with significant improvement in Precision, Recall and $F_{measure}$ compared to the other deep learning-based methodologies proposed in [15], [53]. Precision, Recall and $F_{measure}$ of our detection results can achieve 0.99, 0.78 and 0.88 on the Traffic Guide Panel dataset, and the averaged computational time is 0.02 second. Compare to the state-of-the-art algorithm, the proposed method has achieved better results in Precision and computational speed, while the Recall rate is lower. As shown in Figure 4.10, the method fails on many challenging highway scene, such as backlighting, low resolutions and blur text. Under backlighting situation, the traffic sign ROIs are too dark to be detected. Low resolutions of small traffic sign regions will also cause failure. The Traffic Guide Panel dataset contains approximately 1/3 testing images with low light and backlighting, which affects Recall value of the proposed method.

Table 4.1: Precision, Recall, $F_{measure}$, and running times of different text-based traffic sign detection methods on the Traffic Guide Panel dataset

Method	Precision	Recall	$F_{measure}$	Time(s)	Device
Epshtain et al. [46]	0.35	0.41	0.38	2.51	CPU
Gómez and karatzas [45]	0.46	0.53	0.49	1.32	CPU
Jaderberg et al. [53]	0.59	0.71	0.64	4.53	GPU&CPU
Rong et al. [15]	0.73	0.64	0.68	0.16	GPU
Zhu et al. [4]	0.90	**0.87**	**0.88**	0.15	GPU&CPU
Our framework	**0.99**	0.78	**0.88**	**0.02**	GPU

(a)

(b)

(c)

Figure 4.10: Failure detection cases on Traffic Guide Panel dataset((a), (b)) and ETFLab-TTSVD (c).

(a) Frame 23

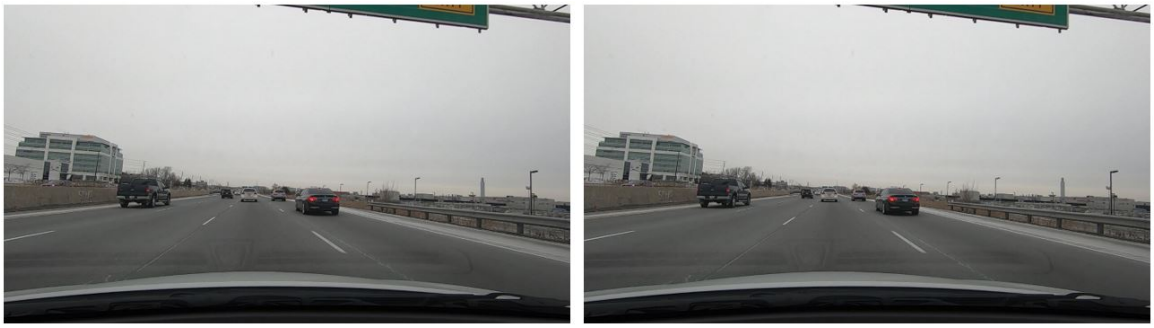
(b) Frame 28

(c) Frame 36

(d) Frame 99

(e) Frame 137

(f) Frame 143



(g) Frame 146

Figure 4.11: Illustration of the comparison results with detection functionality only (left) and detection with tracking functionality (right) on ETFLab-TTSVD. Detection and tracking results are in yellow and red bounding boxes, respectively.

To test the effectiveness of the proposed detection and tracking framework in real life scenario, the proposed framework is evaluated on ETFLab-TTSVD. The images are resized to $416 * 416$ (i.e. $N = 13$) during the detection stage, and the threshold of detection probability is set to 0.8, some sample results are shown in

Figure 4.9 (e)-(f). During the tracking process, we set $V_{max} = 240\ km/h$, $\mu_k = 0.7$ and $g = 9.8\ ms^{-2}$ for estimation stage. From the experiments, Precision, Recall and $F_{measure}$ of our detection results can achieve 0.98, 0.87 and 0.92, respectively, with detection functionality only. With additional tracking functionality, the value of Precision decreases slightly to 0.96, while Recall and $F_{measure}$ increase to 0.98 and 0.97, respectively. The decrease in Precision is related to delayed deletion when the traffic sign moves out of FOV. In the mean time, the increment in Recall is caused by the predict phases in the tracking algorithm.

As shown in Figure 4.11, the left and right column are the the results of detection functionality only and detection with tracking functionality, respectively. First traffic sign is initially detected at frame 23 (4.11a), it is assigned traffic sign ID 0. Miss detection occurs while the vehicle moving forward, the tracker predict its position at frame 28 (4.11b). The traffic sign is detected again at frame 36 (4.11c), the tracker is able to preserve its identity. Another traffic sign is initially detected at frame 99 (4.11d), it is assigned traffic sign ID 1. Two traffic signs are detected at frame 137 (4.11e), though the detected bounding boxes are overlapping, the tracker maintain their identity. Only one traffic sign is detected at frame 143 (4.11f), the tracker judges the traffic sign ID 1 moves out of FOV, delete its ID. No traffic sign is detected at frame 146 (4.11g) and the tracker judges the traffic sign ID 0 moves out of FOV, delete its ID.

The limitations of the proposed multi-target detection and tracking algorithm are as follows: first, the proposed algorithm is able to track the traffic signs with only one initial detection, but will have better prediction with two initial detections. Second, the prediction of traffic sign may be tricked when partial of the traffic sign moves out

of FOV. This may lead to delayed deletion in the tracking stage. Third, the proposed framework cannot handle false alarm without recognition functionality, hence a high Precision detector is needed. The effective detected distance depends on the size of traffic sign, from experimental result, the presented text-based traffic sign detector can detect the traffic sign on average 3 seconds (approximate 80 meters) before the sign moves out of FOV for 1080p videos.

The computational time for the detection stage is 0.02 second per frame with the GPU, and the tracking process takes 0.2 ms (CPU) with $2 \sim 3$ traffic signs per frame, more time may expect when more traffic signs need to be tracked. The proposed framework can achieve 24 fps per frame in real application. Our proposed tracking algorithm is implemented using the Python language, faster runtimes could be achieved with C++ implementations.

4.5 Conclusion

In this Chapter, a text-based traffic sign detection and tracking framework is proposed for the video-based TSR system. For detection stage, an effective text-based traffic sign detector is trained on street view images. The proposed multi-target tracking algorithm have 5 phases, update existing traffic signs, delete the traffic signs if move out of FOV or max disappear time is reached, predict the position of traffic signs when missed detection occurs, add newly detected traffic signs and estimate search regions for the next frame. To estimate the position and size for the same traffic sign in the following frame, the kinematic automotive motion model is used. The kinematic automotive motion model details the kinematic states of vehicles and the spatial-temporal relationships between the motionless traffic signs and moving vehicles. From

the experimental results on public and real-life datasets, the proposed framework is useful and effective to be used in real time application. Future work will increase the number and variations of the training dataset, and integrate the framework with recognition functionality.

Chapter 5

Conclusions and Future Works

5.1 Research Summary

In this thesis, multiple text-based traffic signs detection and tracking related tasks are studied. A traffic sign localization algorithm, a CPU-based text-based traffic sign detector, and a text-based traffic signs detection and tracking framework for video are presented.

For text-based traffic sign localization task, a kinematic automotive motion model is presented. This kinematic model details the spatial-temporal relationship between motionless traffic signs and different kinematic states of vehicles. Based on the kinematic automotive motion model, the text-based traffic sign localization algorithm is developed. The proposed localization algorithm takes the advantages of on-board data source such as rotation of the wheels and directional information of vehicles, and is able to predict the position of the traffic sign after initial detections. In this way, the localization algorithm can handle environmental complexities such as uneven lighting and occlusion since the prediction is independent of data acquisition from camera.

The experimental results show that, with good initial detections, the proposed localization algorithm can be used to significantly reduce the computational cost as well as maintain good values of IoU of predicted text-based traffic sign bounding boxes in real-time TSR applications.

For text-based traffic sign detection task, a CPU-based detection algorithm is presented. To relax the restriction of data acquisition from on-board sensor, a parameter estimation method is first developed for kinematic automotive motion model based on different environmental/weather conditions. Next, with the estimated turning angle parameter, the search region can be modelled and calculated for the text-based traffic signs using kinematic automotive motion model. The ROIs are then extracted within the defined search regions by using a fast and effective CE-MSERs detector. During the the final selection stage, the spatial information is used to reduce the computation time. From the experimental results, the method can be used to significantly reduce the computational cost of traffic sign detection while maintaining the high performance of detection results in real-time applications.

For detection and tracking framework task, a multiple text-based traffic signs detection and tracking framework is proposed for the video-based TSR system. During the detection stage, a fast and accurate data-driven text-based traffic sign detector is trained on street view images, which are collected with low cost. In tracking stage, a multiple text-based traffic signs tracking algorithm is presented. The proposed multi-target tracking algorithm have 5 phases, update the position of existing traffic signs, delete the traffic signs if out of FOV or max disappear time is reached, predict the position of traffic signs when missed detection occurs, add newly detected traffic signs and estimate the position and size for the next frame. To estimate the position

and size for the same traffic sign in the following frame, the kinematic automotive motion model is used. From the experimental results on public datasets, the proposed framework is useful and effective to be used in real-time application.

5.2 Future Works

For future research, the proposed framework can be further extended and improved in the following aspects:

1) A unified traffic sign detector: The current work are only focusing on the detection of text-based traffic signs. The data driven approach in Chapter 4 can be further extended to graphics-based traffic signs. In this way, both categorizes of traffic signs can be detected with one unified detector at the same time.

2) Multilingual environment: Detecting the traffic signs in the multilingual environment and translating them to the designated language is one of the important TSR application. In future work, the traffic sign detector can be further extended to adapt to multilingual environment.

3) Recognition functionality: The current framework does not include traffic sign recognition functionality, in the future work, the proposed framework will be integrated with a traffic sign recognition module.

4) Night vision situation: Night vision is a challenging situation in TSR studies. Detecting the traffic signs under night vision and low light situation are worth exploring in the future.

5) Optimizing for embedded systems: In real world applications, TSR systems are implemented on embedded systems. Compared to desktop computer, embedded systems have less computational power and memory size, which will limit the speed

and accuracy of the current work. In the future, the proposed framework will be optimized and tested on embedded systems, and make it ready for real consumer-level TSR systems .

Bibliography

[1] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. Seelen, "An image processing system for driver assistance," *Image and Vision Computing*, vol. 18, no. 5, pp. 367–376, April 2000.

[2] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Machine Vision and Applications*, vol. 25, no. 3, pp. 633–647, April 2014.

[3] "Ontario traffic manual, book 5," *Ministry of Transportation Ontario*, March 2000.

[4] Y. Zhu, M. Liao, M. Yang, and W. Liu, "Cascaded segmentation-detection networks for text-based traffic sign detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 209–219, January 2018.

[5] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, December 2012.

[6] A. Gudigar, S. Chokkadi, and U. Raghavendra, "A review on automatic detection

and recognition of traffic sign," *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 333–364, January 2016.

[7] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, July 2015.

[8] S. Šegvić, K. Brkić, Z. Kalafatić, V. Stanisavljević, M. Ševrović, D. Budimir, and I. Dadić, "A computer vision assisted geoinformation inventory for traffic infrastructure," *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 66–73, September 2010.

[9] I. Bonači, I. Kusalić, I. Kovaček, Z. Kalafatić, and S. Šegvić, "Addressing false alarms and localization inaccuracy in traffic sign detection and recognition," *16th Computer Vision Winter Workshop*, pp. 1–8, January 2011.

[10] K. Brkić, A. Pinz, S. Šegvić, and Z. Kalafatić, "Histogram-based description of local space-time appearance," *Scandinavian Conference on Image Analysis*, pp. 206–217, May 2011.

[11] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, August 2012.

[12] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," *The 2013 international joint conference on neural networks (IJCNN)*, pp. 1–8, August 2013.

[13] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," *Scandinavian conference on image analysis*, pp. 238–249, May 2011.

[14] J. Greenhalgh and M. Mirmehdi, "Recognizing text-based traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1360–1369, June 2015.

[15] X. Rong, C. Yi, and Y. Tian, "Recognizing text-based traffic guide panels with cascaded localization network," *European Conference on Computer Vision*, pp. 109–121, October 2016.

[16] J. Miura, T. Kanda, S. Nakatani, and Y. Shirai, "An active vision system for on-line traffic sign recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 85, no. 11, pp. 1784–1792, November 2002.

[17] A. De la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and vision computing*, vol. 21, no. 3, pp. 247–258, March 2003.

[18] H. Fleyeh, "Color detection and segmentation for road and traffic signs," *IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, pp. 809–814, December 2004.

[19] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2022–2031, July 2016.

[20] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition," *Machine Vision and Applications*, vol. 22, no. 2, pp. 359–375, March 2011.

[21] M. A. Garcia-Garrido, M. A. Sotelo, and E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions," *IEEE Intelligent Transportation Systems Conference*, pp. 811–816, September 2006.

[22] Á. González, M. Á. Garrido, D. F. Llorca, M. Gavilán, J. P. Fernández, P. F. Alcantarilla, I. Parra, F. Herranz, L. M. Bergasa, M. Á. Sotelo *et al.*, "Automatic traffic signs and panels inspection system using computer vision," *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 2, pp. 485–499, January 2011.

[23] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," *IEEE Intelligent Vehicles Symposium*, pp. 566–571, June 2004.

[24] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 70–75, September 2004.

[25] N. Barnes, A. Zelinsky, and L. S. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 322–332, June 2008.

[26] Y. Gu, T. Yendo, M. P. Tehrani, T. Fujii, and M. Tanimoto, "Traffic sign detection in dual-focal active camera system," *IEEE Intelligent Vehicles Symposium*, pp. 1054–1059, June 2011.

[27] X. Baró, S. Escalera, J. Vitrià, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary adaboost detection and forest-ecoc classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 113–126, February 2009.

[28] I. M. Creusen, R. G. Wijnhoven, E. Herbschleb, and P. de With, "Color exploitation in hog-based traffic sign detection," *IEEE International Conference on Image Processing*, pp. 2669–2672, September 2010.

[29] A. Møgelmose, D. Liu, and M. M. Trivedi, "Detection of U.S. traffic signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3116–3125, June 2015.

[30] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," *Asian Conference on Computer Vision*, pp. 770–783, November 2010.

[31] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 4, pp. 1498–1506, August 2012.

[32] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, "A traffic sign detection pipeline based on interest region extraction," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, August 2013.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, September 1995.

[34] Z. Huang, Y. Yu, S. Ye, and H. Liu, "Extreme learning machine based traffic sign detection," *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, pp. 1–6, September 2014.

[35] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, August 2013.

[36] Chiung-Yao Fang, Sei-Wang Chen, and Chiou-Shann Fuh, "Road-sign detection and tracking," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 5, pp. 1329–1341, September 2003.

[37] Wen Wu, Xilin Chen, and Jie Yang, "Detection of text on road signs from video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 378–390, December 2005.

[38] A. V. Reina, R. L. Sastre, S. L. Arroyo, and P. G. Jiménez, "Adaptive traffic road sign panels text extraction," *Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation*, pp. 295–300, February 2006.

[39] A. Gonzalez, L. M. Bergasa, and J. J. Yebes, "Text detection and recognition on traffic panels from street-level imagery using visual appearance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 228–238, August 2013.

[40] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," *Workshop on statistical learning in computer*

vision, European Conference on Computer Vision, vol. 1, no. 1-22, pp. 1–2, May 2004.

[41] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," *European Conference on Machine Learning*, pp. 4–15, April 1998.

[42] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic sign recognition using a multi-task convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1100–1111, June 2017.

[43] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," *International Conference on Document Analysis and Recognition*, pp. 1491–1496, September 2011.

[44] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazn, and L. P. de las Heras, "ICDAR 2013 robust reading competition," *12th International Conference on Document Analysis and Recognition*, pp. 1484–1493, August 2013.

[45] L. Gómez and D. Karatzas, "Multi-script text extraction from natural scenes," *International Conference on Document Analysis and Recognition*, pp. 467–471, August 2013.

[46] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2963–2970, June 2010.

[47] X. Yin, X. Yin, K. Huang, and H. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, May 2014.

[48] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–II, June 2004.

[49] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2296–2305, June 2013.

[50] X. Yin, Z. Zuo, S. Tian, and C. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2752–2773, June 2016.

[51] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3304–3308, November 2012.

[52] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2558–2567, 2015.

[53] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," *European Conference on Computer Vision*, pp. 512–528, September 2014.

[54] Y. Pan, X. Hou, and C. Liu, "A hybrid approach to detect and localize texts in

natural scene images," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, March 2011.

[55] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," *2011 18th IEEE International Conference on Image Processing*, pp. 2609–2612, September 2011.

[56] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083–1090, June 2012.

[57] L. Neumann and J. Matas, "Real-time scene text localization and recognition," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3538–3545, June 2012.

[58] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *European Conference on Computer Vision*, pp. 21–37, October 2016.

[59] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," *Thirty-First AAAI Conference on Artificial Intelligence*, February 2017.

[60] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2529–2541, June 2016.

[61] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," *European Conference on Computer Vision*, pp. 497–511, September 2014.

[62] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," *European Conference on Computer Vision*, pp. 56–72, October 2016.

[63] Z. Boxin, J. Jun, N. Yifeng, and S. Lincheng, "Real-time detection and tracking of traffic sign in video sequences for autonomous mobile robot," pp. 711–715, January 2012.

[64] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," pp. 255–260, June 2005.

[65] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition," *Image and Vision Computing*, vol. 14, no. 3, pp. 209–223, April 1996.

[66] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognition*, vol. 43, no. 1, pp. 416–430, January 2010.

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

[68] Y. Yuan, Z. Xiong, and Q. Wang, "An incremental framework for video-based

traffic sign detection, tracking, and recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1918–1929, July 2017.

[69] Z. Zheng, H. Zhang, B. Wang, and Z. Gao, "Robust traffic sign recognition and tracking for advanced driver assistance systems," pp. 704–709, September 2012.

[70] H. Zhu, K. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2584–2601, October 2017.

[71] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," *2006 IEEE Intelligent Transportation Systems Conference*, pp. 1078–1083, Spetember 2006.

[72] H. Huang and L. Hou, "Speed limit sign detection based on Gaussian color model and template matching," *International Conference on Vision, Image and Signal Processing (ICVISP)*, pp. 118–122, September 2017.

[73] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, February 2016.

[74] V. A. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. V. Gool, "Integrating object detection with 3D tracking towards a better driver assistance system," *2010 20th International Conference on Pattern Recognition*, pp. 3344–3347, August 2010.

[75] S. Khalid, N. Muhammad, and M. Sharif, "Automatic measurement of the traffic sign with digital segmentation and recognition," *IET Intelligent Transport Systems*, vol. 13, no. 2, pp. 269–279, September 2018.

[76] R. Biswas, H. Fleyeh, and M. Mostakim, "Detection and classification of speed limit traffic signs," *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pp. 1–6, January 2014.

[77] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2003.

[78] J. Hu, R. Tharmarasa, R. Lee, and T. Kirubarajan, "Benchmark dataset for localization of text-based traffic signs using a kinematic automotive model," June 2020. Accessed on: June 3, 2020. [Online]. Available: http://www.ece.mcmaster.ca/~kiruba/ETFLab-TTSVD.zip

[79] R. A. Serway, J. S. Faughn, and C. Vuille, *College physics*. Saunders College Pub., 1999.

[80] "Friction factors," Accessed on: July 27, 2020. [Online]. Available: https://roymech.org/Useful_Tables/Tribology/co_of_frict.html

[81] "Toyota-corolla specifications," Accessed on: July 27, 2020. [Online]. Available: https://www.toyota.ca/toyota/en/vehicles/corolla/models-specifications

[82] X. Kuang, W. Fu, and L. Yang, "Real-time detection and recognition of road traffic signs using mser and random forests," *International Journal of Online and Biomedical Engineering*, vol. 14, no. 03, pp. 34–51, March 2018.

[83] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," pp. 1–9, June 2015.

[84] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," pp. 779–788, June 2016.

[85] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, April 2018.

[86] J. Salmen, S. Houben, and M. Schlipsing, "Google street view images support the development of vision-based driver assistance systems," pp. 891–895, June 2012.

[87] Z. Zhong, L. Jin, and S. Huang, "Deeptext: A new approach for text proposal generation and text detection in natural images," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1208–1212, March 2017.

[88] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, December 2008.

[89] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, September 2004.

[90] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," November 2011, pp. 1457–1464.

[91] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 8, no. 4, pp. 280–296, September 2006.

[92] R. Girshick, "Fast r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, December 2015.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, December 2012.

[94] L. Neumann and J. Matas, "Text localization in real-world images using efficiently pruned exhaustive search," *International Conference on Document Analysis and Recognition*, pp. 687–691, September 2011.

[95] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2594–2605, March 2011.

[96] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 97–104, December 2013.