Journal of Advanced Transportation

Advanced Measurements and Control Techniques for Intelligent Vehicles

Lead Guest Editor: Guang Chen
Guest Editors: Peng Hang and Yang Xing

Advanced Measurements and Control Techniques for Intelligent Vehicles

Journal of Advanced Transportation

Advanced Measurements and Control Techniques for Intelligent Vehicles

Lead Guest Editor: Guang Chen
Guest Editors: Peng Hang and Yang Xing

Contents

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2673191, 20 pages
https://doi.org/10.1155/2022/2673191

Research Article

Mixed Event-Frame Vision System for Daytime Preceding Vehicle Taillight Signal Measurement Using Event-Based Neuromorphic Vision Sensor

Zhengfa Liu,[1] **Guang Chen**,[1,2] **Ya Wu**,[1] **Jiatong Du**,[1] **Jörg Conradt**,[3] **and Alois Knoll**[2]

[1]*College of Automotive Engineering, Tongji University, Shanghai 201804, China*
[2]*Chair of Robotics, Artificial Intelligence and Real-Time Systems, Technical University of Munich, Munich 85748, Germany*
[3]*KTH Royal Institute of Technology, Stockholm 11428, Sweden*

Correspondence should be addressed to Guang Chen; guangchen@tongji.edu.cn

Received 9 October 2021; Revised 30 July 2022; Accepted 30 August 2022; Published 22 September 2022

Academic Editor: Gabriella Mazzulla

An important aspect of the perception system for intelligent vehicles is the detection and signal measurement of vehicle taillights. In this work, we present a novel vision-based measurement (VBM) system, using an event-based neuromorphic vision sensor, which is able to detect and measure the vehicle taillight signal robustly. To the best of our knowledge, it is for the first time the neuromorphic vision sensor is paid attention to for utilizing in the field of vehicle taillight signal measurement. The event-based neuromorphic vision sensor is a bioinspired sensor that records pixel-level intensity changes, called *events*, as well as the whole picture of the scene. The events naturally respond to illumination changes (such as the ON and OFF state of taillights) in the scene with very low latency. Moreover, the property of a higher dynamic range increases the sensor sensitivity and performance in poor lighting conditions. In this paper, we consider an event-driven solution to measure vehicle taillight signals. In contrast to most existing work that relies purely on standard frame-based cameras for the taillight signal measurement, the presented mixed event/frame system extracts the frequency domain features from the spatial and temporal signal of each taillight region and measures the taillight signal by combining the active-pixel sensor (APS) frames and dynamic vision sensor (DVS) events. A thresholding algorithm and a learned classifier are proposed to jointly achieve the brake-light and turn-light signal measurement. Experiments with real traffic scenes demonstrate the performance of measuring taillight signals under different traffic conditions with a single event-based neuromorphic vision sensor. The results show the high potential of the event-based neuromorphic vision sensor being used for optical signal measurement applications, especially in dynamic environments.

1. Introduction

Vehicle safety technology is playing a more and more important role in intelligent vehicles [1]. Advanced driver assistance systems have been developed to assist in driving and avoid potential hazards by warning drivers based on environmental perception [2–7]. During driving, vehicle deceleration is possible to cause rear-end collisions, especially when drivers are distracted. For human drivers, taillights are critical warning signals of the deceleration of former vehicles. Therefore, taillight signal measurement is a promising approach to collision avoidance and mitigation. Currently, taillight signal measurement algorithms are based on a standard frame-based camera. Images acquired from the standard frame-based cameras always produce poor quality with low resolution and motion blur when confronted with rapid movements. The standard

frame-based camera also has trouble measuring taillight signals when there are sudden changes in light, like when you enter a tunnel, or when there is a strong light source, like the sun.

The event-based neuromorphic vision sensors, such as the dynamic vision sensor (DVS) [8–10], can overcome the above-mentioned limitations of the standard frame-based cameras. Different from the standard frame-based cameras, in which measurements arrive at fixed time intervals, the event-based sensors can generate data according to the relative light intensity changes asynchronously. By registering these changes on the order of tens of microseconds, the event-based sensors have almost instant feedback, with a high temporal resolution and much less motion blur. Another important feature of the event-based sensors is their much higher dynamic range (120 dB), while the dynamic range of the standard frame-based cameras is usually about 60 dB. These features make the event-based sensors particularly suitable for daytime taillight signal measurement under different light conditions [11], e.g., at noontime or dusk, in different traffic scenes. As shown in Figure 1, standard frame-based cameras sample their environment with a fixed frequency and produce a series of frames, which lose all the information between two adjacent frames. In contrast to standard frame-based cameras, event-based sensors asynchronously respond to pixel-level brightness changes with microsecond latency and do not report anything when everything is at rest. In the taillight signal measurement system, we assume that the vision sensor and the detected vehicle are relatively stationary. Therefore, the event stream is generated only when the taillight state changes, i.e., ON $\rightarrow$ OFF or OFF $\rightarrow$ ON. From the point of view of event generation, the density change of the event stream closely follows the transition of the taillight state. Specifically, it corresponds to a large number of dense event streams when the state changes and a small number of sparse event streams when the state remains unchanged. Therefore, the event-based sensors are suitable for taillight signal measurements.

However, events do not provide absolute brightness values and contain no RGB information. Thus, it presents difficulties in performing robust and long-term taillight signal measurement. Considering the advantageous features and drawbacks, in this paper, we propose a robust taillight signal measurement system based on a novel event-based neuromorphic vision sensor named the dynamic active-pixel vision sensor (DAVIS) [12–14]. The DAVIS contains an active-pixel sensor and a dynamic vision sensor in the same pixel array, which generates fixed frame-rate APS frames and asynchronous DVS events. It inherits the advantages of the event-based sensor while ensuring consistency with the standard frame-based camera.

Figure 2 illustrates the proposed taillight signal measurement system. We define all the five states of the preceding vehicle taillights as braking-on, braking-off, turning-off, turning-right-on, and turning-left-on. The following vehicle installs a single DAVIS sensor with a resolution of 346×260 pixels to capture APS frames and DVS events while following the preceding vehicles. The taillight signal measurement system first locates the taillight regions of the preceding vehicles based on the APS frame. Then, the frequency domain features of each taillight region are extracted from the corresponding DVS events, which are used for taillight signal measurement. As the first attempt to use event-based sensors in the taillight signal measurement system, we believe that even-based sensors can be of interest for many vision-based measurement system research efforts. The contributions of this work are summarized as follows:

(1) We present a novel taillight signal measurement system that combines APS frames with DVS events by using a single event-based neuromorphic vision sensor.

(2) We extract the frequency domain features from the spatial and temporal signals of each taillight region based on DVS events, thus restraining the influences of low-frequency backgrounds.

(3) A thresholding algorithm and a learned AdaBoost classifier are proposed to jointly achieve brake-light and turn-light signal measurement.

The remainder of this paper is organized as follows. Section 2 provides a review of the traditional frame-based vehicle taillight signal measurement system and describes the event-based neuromorphic vision sensor. Section 3 specifies the details of the proposed taillight signal measurement system. Section 4 discusses the experimental results and analysis. Section 5 presents the conclusions drawn from this paper.

2. Related Work

2.1. Frame-Based Vehicle Taillight Signal Measurement. Existing works of the frame-based vehicle taillight signal measurement can be classified into three categories based on the information used.

2.1.1. Temporal Information-Based Methods. The time-dependent information is often used for the tracking of the taillights [15–18]. [15] applies a detection-tracking model and uses a trained WaldBoost detector to obtain the new tracker. Then, the tracking is performed by a flock of trackers. [16] proposes a perceptive algorithm to track candidate vehicles, and the turn signals are detected by analyzing the continuous intensity variation of the vehicle box sequences. [19–22] focus on extracting the invariant features from the tracked regions of taillight lights in the frequency domain. [19, 20] train an AdaBoost classifier for turn signal detection. [21, 22] use a measure function where the current frame value is normalized by the value of the current frame and last frame for brake signal measurement.

2.1.2. Color Information-Based Methods. The color information-based methods often extract features via morphology [23, 24] and color/intensity thresholds [25–27]. Different color spaces are used, such as RGB [28], HSV [29], YCbCr [27], YUV [30], and Lab [31]. [25] introduces a lamp response function for rear lamp detection and a high-pass

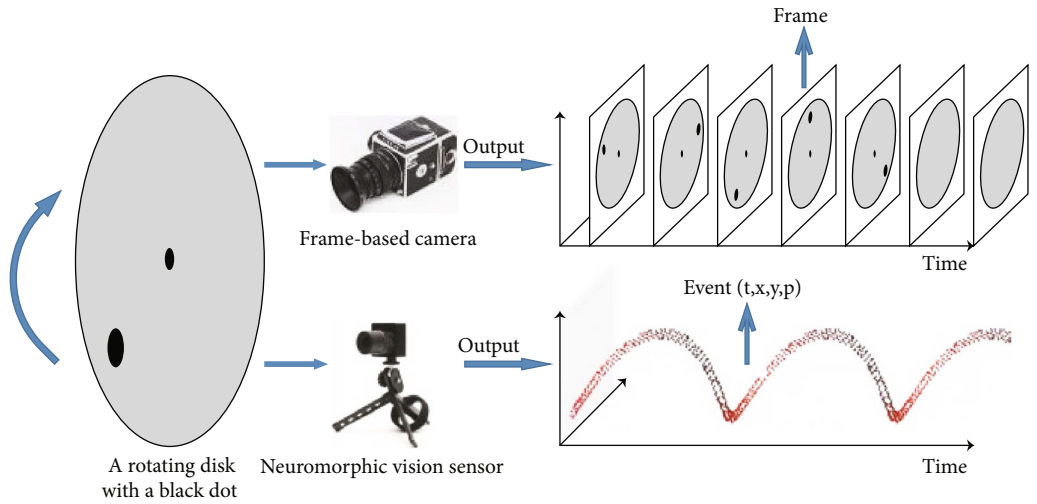


Figure 1: Comparison of the output of a standard frame-based and an event-based sensor when facing a rotating disk with a black dot. The standard camera outputs frames at a fixed rate, thus sending redundant information when no motion is present in the scene. Standard cameras also suffer from motion blur during rapid motion and lost all the information between two adjacent frames. Event-based sensors instead respond to pixel-level brightness changes with microsecond latency. Therefore, they do not suffer from motion blur and do not report anything when everything is at rest. This figure is cited from [66].

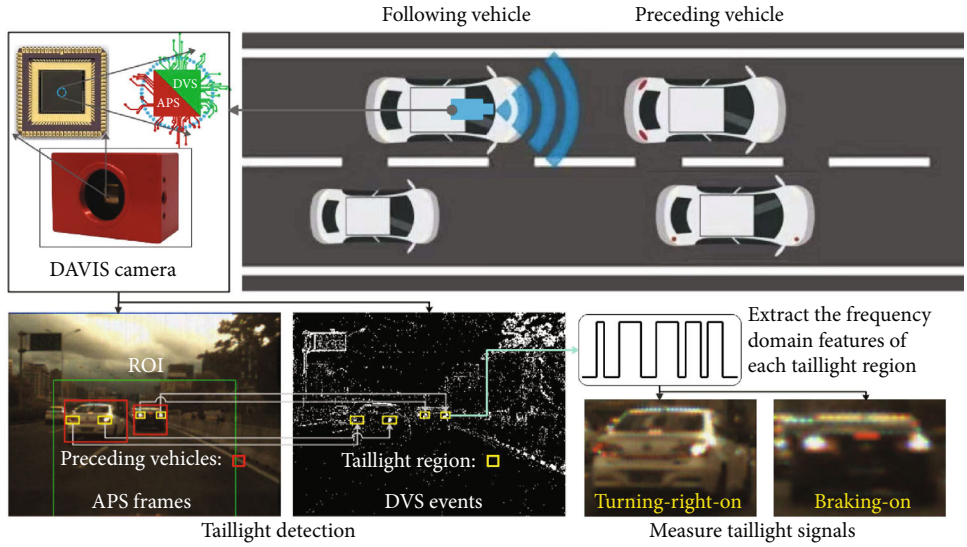


Figure 2: Illustration of the taillight signal measurement system.

mask for rear lamp signal measurement. Based on the Nakagami-m distribution, [26] adopts color thresholds to detect turn signals at night by scattering modeling of taillights. It utilizes a contrast of reflectance to describe the direction. [27] uses YCrCb color space as feature space to detect brake behavior.

2.1.3. Mixed Information-Based Methods. Other methods combine advantages from the above two categories to increase their reliability and efficiency by combining temporal and color information for detecting and tracking vehicle taillights [17, 32–34]. [34] utilizes both luminance and radial symmetry features for brake-light state determination, in which a detection refinement process using temporal information is employed for miss recovery. Most of the above methods make use of the color/brightness information and the symmetric information of vehicle taillights for object localization, combined with a trained classifier for object confirmation. Recently, deep learning approaches have also been applied to learn features for vehicle taillight detection. [35] firstly uses Fast-RCNN to detect vehicles, and then segments the vehicle taillight regions using FCN to extract features and detect the brake lights using an SVM classifier. [36] uses the brake-light patterns learned from the vehicle taillights appearance by a fine-tuning AlexNet model to measure brake-light signal. [37] uses a brake-light classifier

based on chromatic and CNN features. [38, 39] propose a CNN-LSTM framework paired with a spatiotemporal attention model for taillight signal measurement, where the networks for brake and turn signals are trained separately.

Although frame-based sensors provide rich RGB information, which is beneficial for taillight signal measurement, a low temporal resolution and limited dynamic range reduce the sensor performance in challenging environments, e.g., motion blur in fast moving scenes, or poor image quality under the condition of a sudden change of light. To meet requirements for taillight signal measurement in different traffic conditions and fast grasping applications, it is essential to consider a sensor with a higher sampling rate and sensitivity while maintaining rich RGB information.

2.2. Event-Based Neuromorphic Vision Sensor

2.2.1. Dynamic Vision Sensor. The standard frame-based cameras output visual information in the form of frames at a constant rate. In contrast, event-based neuromorphic vision sensors, such as the dynamic vision sensor (DVS) [8], exhibit a far more efficient encoding manner. Each independent pixel in a DVS only outputs data in response to the log-intensity brightness changes [40–42]. Given a static scene, these pixels will not produce any output, and therefore, the data rate from the device is dependent on the activity in the scene. Each pixel emits an AER (address event representation) [43] event $e = \{x, y, t, p\}$ containing the physical location of the pixel in the array (x, y), and generally a single bit of information to indicate whether the illumination on the pixel increased or decreased at time t. The direction of the change in illumination is encoded as $p \in \{-1, 1\}$, in which $p = 1$ is conventionally referred to as an ON event, representing an increase in illumination, and $p = -1$ correspondingly representing an OFF event in which a decrease in illumination occurs. The temporal resolution is limited by the rate at which events can be read from the physical hardware (usually on the order of microseconds). Unlike the standard frame-based cameras, there is no concept of frames for the event-based sensors, as the data arrives entirely asynchronously. In summary, the event-based sensors offer multiple advantages over the standard frame-based cameras, mainly (1) high temporal resolution, which allows the capture of multiple events in microseconds; (2) high dynamic range, which allows the information capture in difficult lighting environments, such as night or very bright scenarios; and (3) low power and bandwidth requirements.

In this work, a specific event-based sensor named the dynamic and active-pixel vision (DAVIS) [12] is used, which implements a standard frame-based camera and an event-based sensor in the same array of pixels. Therefore, the output consists of a stream of asynchronous high-rate events together with a stream of synchronous color frames (APS frames) acquired at a low rate. An example of the output of the DAVIS sensor is shown in Figure 2. It is important to note that the notion of frames is absent from the event-based sensor acquisition process. Event frames can be reconstructed, when needed, by buffering the events generated over a given period. As can be seen from Figure 3(b), for representation, the DVS events are collected every 20 ms to form the accumulated event frame.

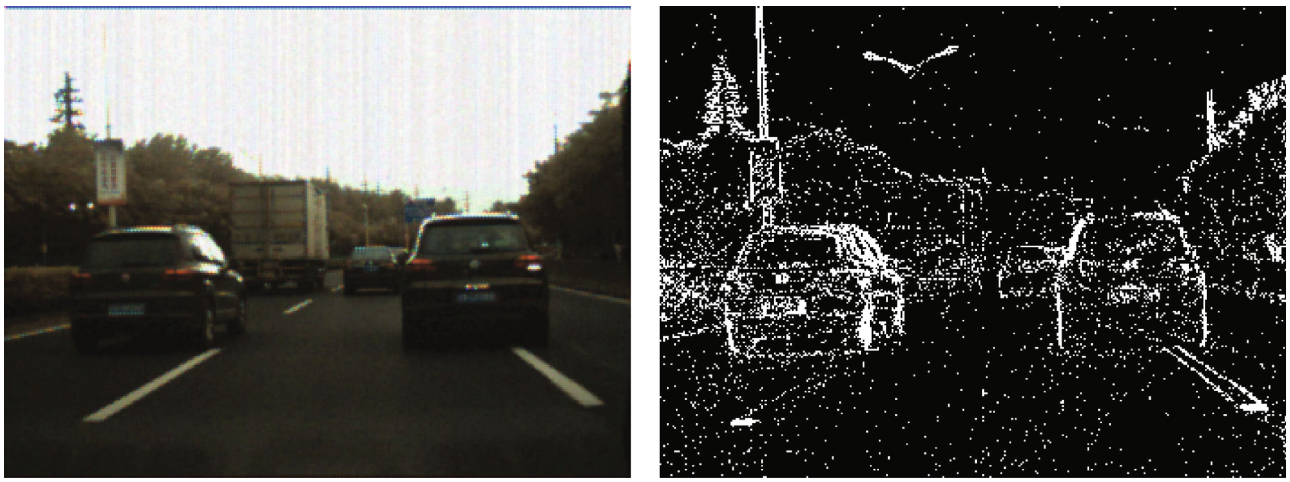
2.2.2. Algorithms for Event Processing. Since the output of an event-based sensor is an asynchronous stream of events, existing computer vision techniques that are designed for standard frame-based cameras cannot be directly applied to process events. Consequently, many algorithms have been specifically tailored to leverage events, either by processing the event stream in an event-by-event fashion or by building intermediate, "image-like" representations from event data [44]. The former methods can achieve minimal latency but are sensitive to parameter tuning and are computationally intensive because they all perform an update step for each event. In contrast, methods operating on event images trade-off latency for computational efficiency and performance [45]. Despite their differences, both paradigms have been successfully applied to recognition tasks [46–50]. [46] describes a real-time hand gesture recognition system based on a stereo pair of DVSs and an event-driven processing technique based on LIF neurons. [47, 48] introduce novel event-based feature representation, a hierarchy of time surfaces (HOTS) [47] and histogram of average time surfaces (HATS) [48] for object recognition. Instead of [47, 48], where pure event counts are measured and summed for each pixel and polarity to generate an event count image, [49, 50] use event timestamp to construct the surface of active events (SAE) for each pixel and polarity.

The event-based sensors mentioned above mainly focus on computer vision tasks. They also are used for vision-based measurement (VBM) systems. Event-based neuromorphic vision sensors have become significantly popular recently and introduce a paradigm in computer vision applications for (VBM) systems [11, 51–56]. Thanks to the low latency and low power consumption of the event-based sensor, an event-based frame approach is proposed to measure the contact force in grasping applications by attaching the event-based sensor to an elastic material in [11, 51] and [53, 56] for incipient slip detection. [11, 54, 55] use the event-based sensor for force estimation. The results show the high potential of the event-based sensor used for manipulation applications, especially in a dynamic environment. Different from tactile sensing, in this paper, we propose a VBM approach to measure the frequency characteristics of each taillight region for the taillight signal measurement using an event-based sensor mounted on the following vehicle, where APS frames and DVS events are captured by the camera.

3. Method

3.1. System Architecture. The overview architecture of the proposed system is illustrated in Figure 4, and it consists of four stages: vehicle detection, taillight localization, feature extraction, and signal measurement.

Due to the DVS events that can provide dense temporal information about the changes in scenes, a simple but effective vehicle taillight signal measurement method becomes



(a) (b)

Figure 3: DAVIS camera output. (a) APS frame. (b) Accumulated event frame.

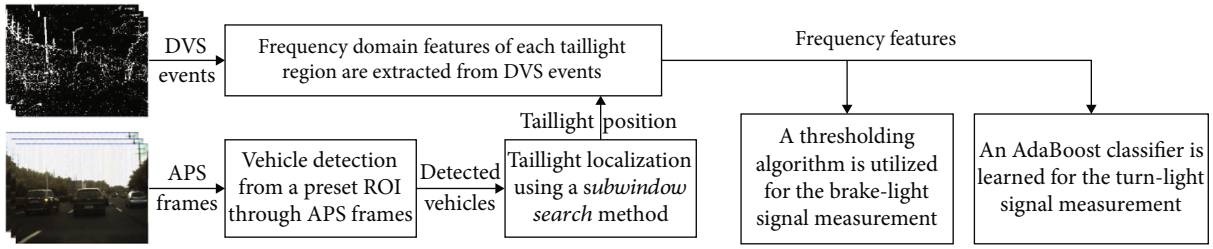


Figure 4: Schematic flowchart of the proposed system.

achievable in the wild. In this work, we explore and study how to use the event-based neuromorphic sensor in a vehicle taillight signal measurement task. Specifically, building on the DVS events triggered by per-pixel brightness changes in the scene asynchronously, we measure the taillight signals based on the frequency characteristics of the sequential events within the taillight regions. The DVS event density changes drastically when the taillight state changes, especially within the taillight regions; i.e., the number of DVS events increases when a preceding vehicle changes its taillight state from braking-off to braking-on. Otherwise, the number of DVS events fluctuates within some range in the holding phase. The trend in the number of DVS events is consistent with the changes in the brightness of the taillight accordingly. Thus, we can utilize the frequency characteristics of the number of DVS events within one taillight region to measure the vehicle taillight signals. In the end, a simple but effective preceding vehicle taillight signal measurement method can be achieved, with fewer computation requirements compared to conventional vision-based techniques where only APS frames are used [39].

3.2. Vehicle Detection. The proposed preceding vehicle taillight signal measurement system mainly pays attention to vehicles in front of the DAVIS sensor. Therefore, we reduce the processing area for vehicle detection by presetting a ROI on the APS frame. The ROI is the centered two-thirds area at the bottom of an APS frame, and the vehicles outside the ROI are ignored. It not only greatly reduces computation consumption but also avoids unnecessary disturbances from the taillights of nearby vehicles.

Vehicle detection can be said to be a very mature technology. A lot of work has been done in the field of vehicle detection [57], and different studies in object detection classification [58] have also attempted to automatically detect and classify different classes of vehicles. For vehicle detection, in this paper, we adopt a multistage object detection architecture, the Cascade R-CNN [59]. With a simple and effective detection architecture and the released source code, Cascade R-CNN is suitable for the taillight measurement system requiring lightweight and rapidity. Specifically, this architecture is composed of a sequence of detectors trained with increasing intersection-over-union (IOU) thresholds for minimizing overfitting and eliminating quality mismatches at inference. It shows good performance in general object detection tasks. We build the Cascade R-CNN on the FPN framework and utilize ResNet-101 as a backbone. A sample result for preceding vehicle detection is shown in Figure 5(a). Then, the bounding boxes of the detected vehicles are grouped into tracks via a simple IOU tracker [60]. The IOU tracker relies on the coincidence area between adjacent frame targets for tracking. With fast tracking speed and low computational cost, it is suitable for use in the proposed taillight signal measurement system. Because robust

(a) (b) (c)

Figure 5: Illustration for vehicle detection. (a) Vehicle detection result. (b) Vehicle detected in an APS frame. (c) LR of the detected vehicle in (b).

vehicle detection and tracking results are the basis for the following steps, we compute the average recall rate of the detected vehicles in the ROI. Our result shows that the average recall rate is close to 100%. The vehicle detection method is not the main focus of this work. Considering that Cascade R-CNN has achieved satisfactory detection results in this work, we do not set up a special comparative experiment to analyze the impact of different vehicle detection methods on the experimental performance.

It is important to note that the success of our method depends on the fact that the taillight region is unique and that different signals share a pair of taillights (refer to Section 3.3.1). However, trucks and buses violate these facts, where different signals correspond to different taillight regions. Therefore, in this work, we only measure the taillight signal of the car and do not focus on the measurement for other vehicle types, such as trucks and buses. To achieve this, we have made two efforts: (1) we focus on the acquisition of the taillight signal of the car in the experimental data acquisition. (2) After "vehicle detection," we only select vehicles labeled "car" by Cascade R-CNN for subsequent steps. On the other hand, in order to apply the proposed measurement system to trucks and buses, we first need to establish a unique correspondence between signal-taillight region pairs before the "taillight localization" step. Considering that the establishment of uniqueness between signal-taillight region pairs is related to the position and setting of the taillights, it is beyond the scope of this study. Hence, in this work, we do not discuss this establishment in detail.

3.3. Taillight Localization. We introduce a *subwindow search* method to locate the taillight position. The *subwindow search* method uses *lamp response* (*LR*) [25] as a quality function to perform clustering and IOU searching operations to finally achieve taillight localization.

3.3.1. Lamp Response. The *LR* is defined in [25]. It measures the relative intensity of the red component compared to the blue and green components of a pixel in an APS frame, which can be described by the following equation:

$$LR(p) = (R_p - G_p) + (R_p - B_p) - 2|G_p - B_p|, \quad (1)$$

where R_p, G_p, and B_p are the RGB color values of pixel p. The taillight is displayed in high red chromaticity. The red component is larger than the green and blue components when the blue and green components are close to each other [25]. Based on the above characteristics of the taillight, we use LR to isolate pixels with large red components in an APS frame for taillight localization. Figures 5(b) and 5(c), respectively, show the APS frame of a detected vehicle and its LR isolation results. As shown in Figure 5(c), pixels with large red components are highlighted when other pixels are suppressed. It can be seen that most of the highlighted regions correspond to the taillight regions of the detected vehicle in Figure 5(b).

3.3.2. Subwindow Search. Based on the LR defined in Equation (1), we formulate taillight localization as a clustering search problem. Specifically, we introduce a *subwindow search* method (see Figure 6) to find the taillight position. This method is divided into three main steps. First, the bounding box of the detected vehicle is traversed by a rectangle of fixed width and height in each dimension of an APS frame using a sliding window algorithm. Overlapping blocks are set between adjacent sliding windows. Then, a quality function (Equation (8)) is defined to measure the intensity of each rectangular subregion (subwindow) on the APS frame. Second, the intensity values of all the subwindows in each dimension are fed into a K-means clustering algorithm to merge these subwindows. Finally, an IOU searching operation is followed to find the taillight position. The process of the *subwindow search* method is illustrated in Figure 6, and the details are described below.

3.3.3. Subwindow Selection. We first divide the bounding box of the detected vehicle into five equal horizontal stripes. Because the taillights tend to appear in the middle of the vehicle, we only use the middle three horizontal stripes as the searching space (SS) for the taillight position to reduce the computational cost. Then, we generate candidate subwindows S ($S = \{S_1, S_2, \cdots, S_i, \cdots, S_n\}$, where n is the number of the subwindows) with overlapping blocks in each dimension of the SS using a sliding window algorithm. These subwindows have a fixed width and height. Suppose the size of the subwindow S_i is $w \times h$. For subwindows in the horizontal direction, w is one-tenth of the width of SS, and h is the same as the height of SS. And similar settings are also applied to subwindows in the vertical direction. For subwindows in the vertical direction, w is the same as the width of SS, and h is one-tenth of the height of SS. The quality function R_{S_i} of the subwindow S_i is defined as

$$R_{S_i} = \frac{\sum_{p \in S_i} LR(p)}{N}, \tag{2}$$

where N is the number of pixels within the subwindow S_i.

3.3.4. Clustering. Based on the intensity value R_{S_i} defined in Equation (2), we aggregate the subwindows in each direction into k clusters through a K-means clustering algorithm. k is the number of clusters expected to be generated. The input of the K-means is the intensity value R_S ($R_S = \{R_{S_1}, R_{S_2}, \cdots, R_{S_i}, \cdots, R_{S_n}\}$) of all subwindows. The output of the K-means is k subwindow clusters. The subwindows that belong to the cluster with the lowest intensity value usually relate to the background. We filter out background interference by removing subwindows in the lowest cluster. After that, the adjacent subwindows in the rest are merged into a group G_j. The quality function R_{G_j} of the group G_j is expressed as R_{G_j}:

$$R_{G_j} = \frac{\sum_{S_i \in G_j} R_{S_i}}{M}, \tag{3}$$

where M is the number of subwindows within the group G_j.

3.3.5. IOU Searching. The best taillight position is found by calculating the intersection area between the best matching groups in both directions (horizontal and vertical). Considering that vehicle taillights are usually located on the left-hand and right-hand sides of the vehicle, the largest group region in the vertical direction is searched, and the first two largest group regions in the horizontal direction are searched.

3.4. Feature Extraction. After locating the taillight position, another challenging task is to measure their ON or OFF signals. Different from existing methods where APS frames are used for vehicle taillight signal measurement, we use DVS events to perform the signal measurement task. To extract the features of the brake-light signal and the turn-light signal, we first convert the DVS events of each taillight region into an event frame. Furthermore, we extract the spatial and temporal characteristics of the signal of each taillight region and transform them into the frequency domain for feature extraction.

3.4.1. Events to Frame Conversion. In order to convert asynchronous DVS events into a synchronous event frame, we accumulate DVS events in a time interval T in a pixel-wise manner to generate a 2D event frame I_E. Similar to [61], we simply use the number of events triggered at pixel location (x_i, y_i) as the intensity value of the pixel, which is expressed as follows:

$$I_E(x, y) = \sum_{i=1, t_i \in T}^{N} \delta(x_i, x)\delta(y_i, y), \tag{4}$$

where δ is the Kronecker delta function (the function is 1 if the variables are equal, and 0 otherwise). N is the total number of events triggered at pixel location (x_i, y_i) within time interval T. Because the event-based sensors naturally respond to illumination changes and moving edges, the raw events of the event-based sensor output are maps encoding changes in taillight brightness and relative motion between the sensor and the vehicle. They cannot be processed directly for feature extraction by using prevalent algorithms. Thus, before generating the event frame, we execute a thresholding algorithm on the event counts to filter out

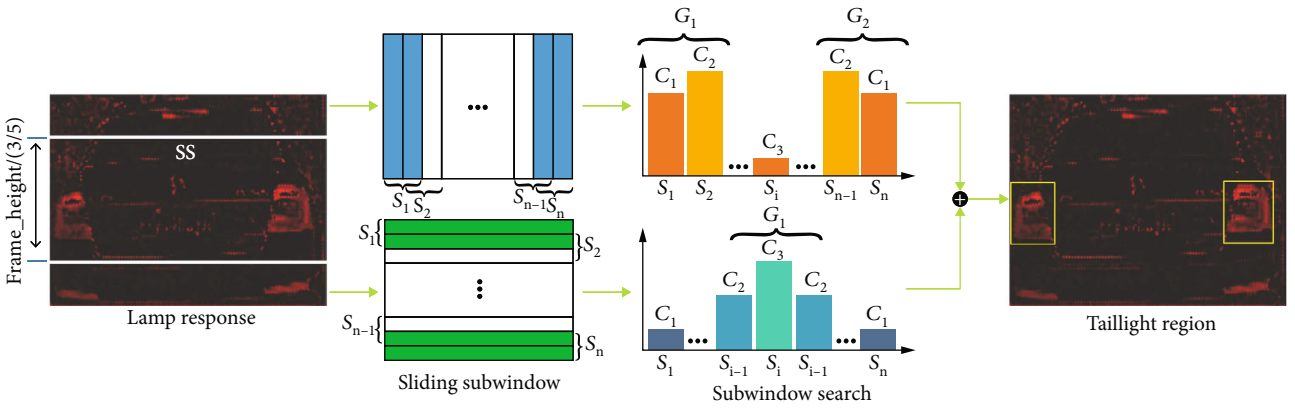
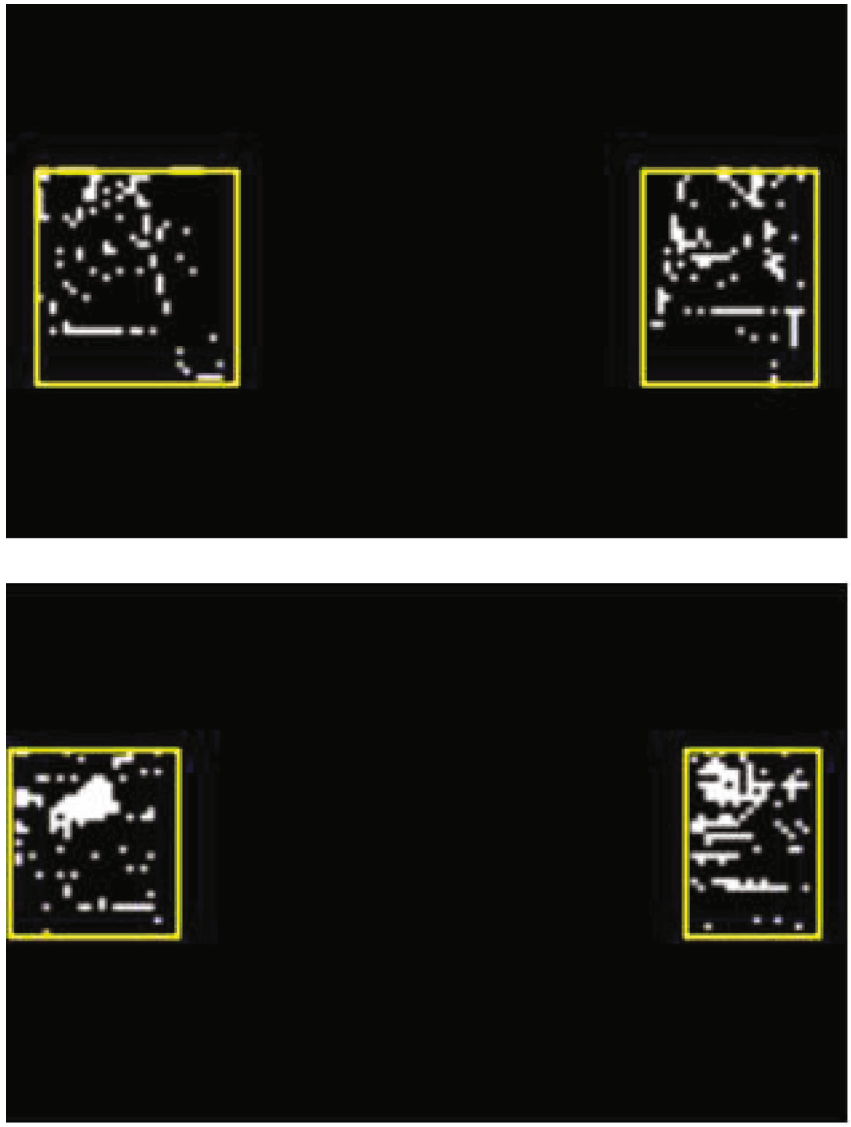


Figure 6: Overview of the proposed *subwindow search* method. First, we narrow down the searching space (SS) of the taillight position to the middle three-fifths area of an APS frame. Then, candidate subwindows S are generated with overlapping blocks in each dimension of SS through a sliding window algorithm. Second, we aggregate the subwindows in each direction into k subwindow clusters. Furthermore, we remove these subwindows in the lowest cluster to filter out background interference and merge the adjacent subwindows in the rest into a group G_j. Finally, an IOU searching operation is followed to find the taillight position. Considering that vehicle taillights are usually located at the left-hand and right-hand sides of the vehicle, the largest group region in the vertical direction is searched, and the first two largest group regions in the horizontal direction are searched.

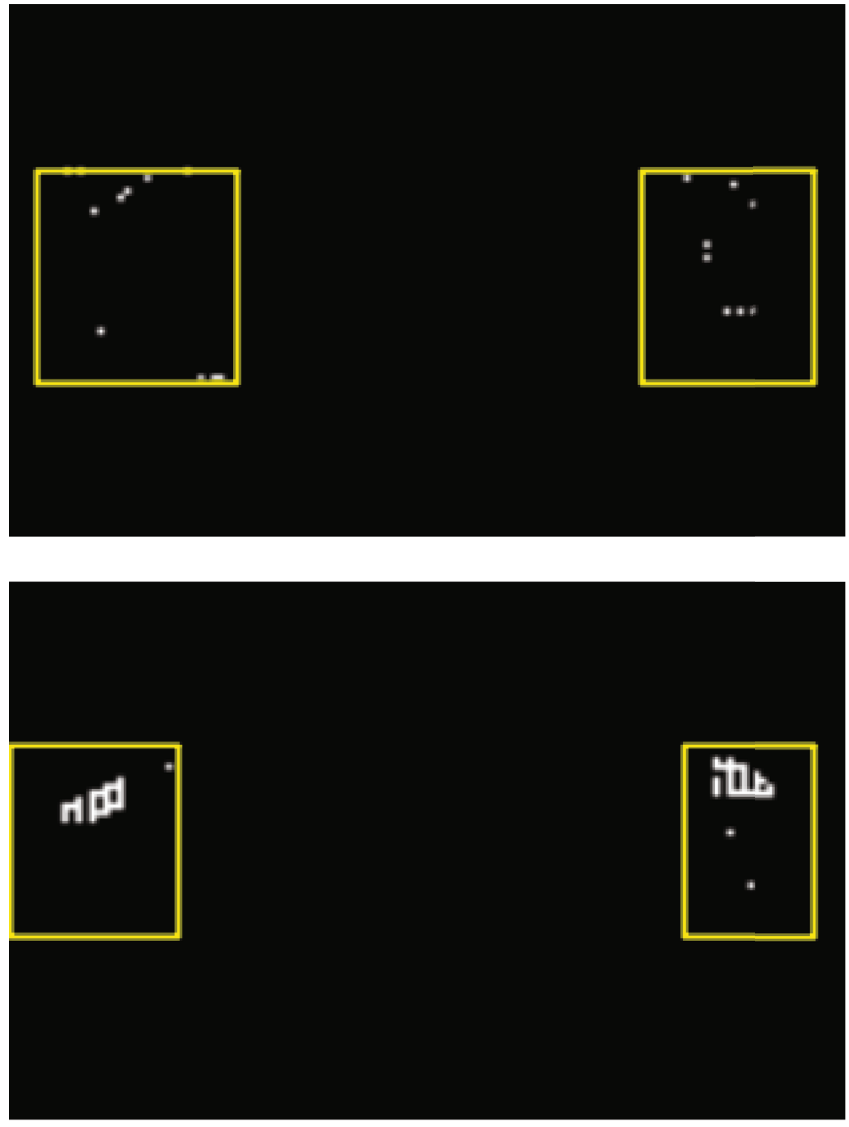
(a)

Figure 7: Continued.



(b)

Figure 7: Continued.



(c)

Figure 7: The APS frame of a detected vehicle and the corresponding synchronized event frame. The top row indicates that the taillight is ON, and the bottom row indicates that the taillight is OFF. (a) APS frame of a detected vehicle. (b) Synchronized event frame from raw DVS events of the vehicle in (a). (c) Synchronized event frame from filtered DVS events of the vehicle in (a).

Table 1: Setting of parameter values used in the experiment.

Parameters	k	T	σ
Value	3	20 ms	0.3

k: the number of clusters generated by the K-means clustering algorithm. T: the time interval size for generating synchronized event frames. σ: the threshold for measuring the brake-light signal.

disturbances from events caused by motion. The events-to-frame conversion process mainly consists of the following two steps:

(1) Event Accumulation. The DVS events are synthesized at a constant time interval of 20 ms. Accurate synchronization between DVS events and the APS frame only occurs near the timestamp of the corresponding APS frame. For a given APS frame timestamp t, we generate the synchronized event frame from the DVS events in period $[t - 10\text{ms}, t + 10\text{ms}]$. Besides, the event frame alleviates the noise impact for a high signal-to-noise ratio (SNR).

(2) Filtering. To generate motion-corrected event frames to establish reliable taillight features, outliers in the distribution of event counts are removed by clipping values of less than three. Figure 7 shows an example of an APS frame and the corresponding synchronized event frame.

3.4.2. Feature Extraction for Brake-Light Signal. According to [21, 22], the brake-light signal can be measured in the frequency domain based on the fact that the activated brake-

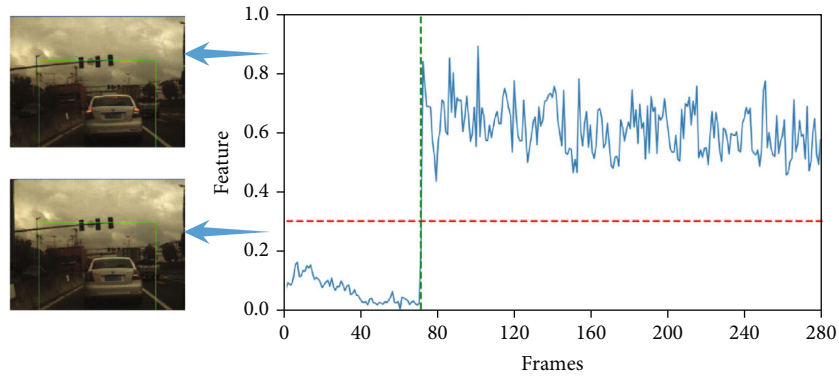


Figure 8: The curve of $\Omega_{c,\max}$ for a testing sequence. The horizontal axis is the frame number and the vertical axis denotes the value of $\Omega_{c,\max}$.

Table 2: Brake-light signal measurement results.

Real state (column)	Braking-off	Braking-on
Braking-off	*99.7%*	0.3%
Braking-on	12.8%	*87.2%*

light possesses a higher frequency property than the nonactivated one. In the DVS frame, the size of the pixel value depends on the number of events. The number of events depends on the level of change in the brightness of the taillight. The brake lights spread wider and brighter, which means that different levels of event streams can be generated in this region. In other words, the DVS frame successfully encodes the scattering of the taillight as a gradient change in pixel values using the number of events. Figure 7(c) demonstrates this statement, where different levels of events are generated at different locations in the taillight region. Therefore, we propose to use the DVS frame instead of the APS frame to extract domain features for brake-light signal measurement. Specifically, we extract the spatial characteristics of the signal of each taillight region and transform them into the frequency domain for the brake-light signal measurement. Because of the computational effectiveness for online implementation, fast Fourier transform (FFT) is a technique often used for frequency analysis [62]. Moreover, FFT has some important properties, such as separability, translation invariant, and rotation invariant. Hence, we adopt a 2D-FFT algorithm to get the high-frequency components of the event frame. Assuming that the event frame I_E is of size $M \times N$, it is transformed into the frequency domain using the 2D-FFT as follows:

$$\Omega(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} I_E(x,y)e^{-j2\pi((ux/M)+(vy/N))}, \quad (5)$$

where $u = 0, 1, 2, \cdots, M-1$ and $v = 0, 1, 2, \cdots, N-1$. $\Omega(u, v)$ represents the frequency domain value of the event frame $I_E(x, y)$. After transformation, the real and complex parts are combined by

$$\Omega(u,v) = R(u,v) + I(u,v),$$
$$\Omega_c(u,v) = \sqrt{R^2(u,v) + I^2(u,v)}. \quad (6)$$

Then, we find the maximum value $\Omega_{c,\max}$ of $\Omega_c(u, v)$ for the following brake-light signal measurement task.

3.4.3. Feature Extraction for Turn-Light Signal. We take into account the frequency of changes in the brightness of the taillight for the turn-light signal measurement. According to [20], the frequency of the turn-light signal is 1.5 ± 0.5 Hz. To be both fast and robust, we calculate the temporal characteristics of the taillight signal within the 2 s time window to extract features. It includes two steps, i.e., feature extraction and feature transformation.

(1) Feature Extraction. The temporal characteristics of the signal of each taillight region are used for feature extraction. Specifically, first, the average of all pixel values for each frame of all frames in the past 2 s is calculated. Here, the frame means the event frame I_E generated in Section 3.4.1. Then, we connect all the averages into one feature vector. With a frame rate of 20 fps, the size of the feature vector is 40, and we represent it as $V(t) = \{V_{t\_0}, V_{t\_1}, \cdots, V_{t\_i}, \cdots, V_{t\_n-1}\}$. In this representation, V_{t_i} refers to the average of the last i^{th} frame.

(2) Feature Transformation. We transform the feature vector V into the frequency domain by using a 1D-FFT algorithm. The frequency domain features are used for the subsequent turn-light signal measurement task. The 1D-FFT is defined as

$$f_k = \sum_{i=0}^{n-1} e^{-j2\pi ki/n} \cdot V_{t\_i}, \quad (7)$$

where $k = 0, 1, \cdots, n-1$. The transformed feature vector is $f(t) = \{f_0, f_1, \cdots, f_i, \cdots, f_{n-1}\}$.

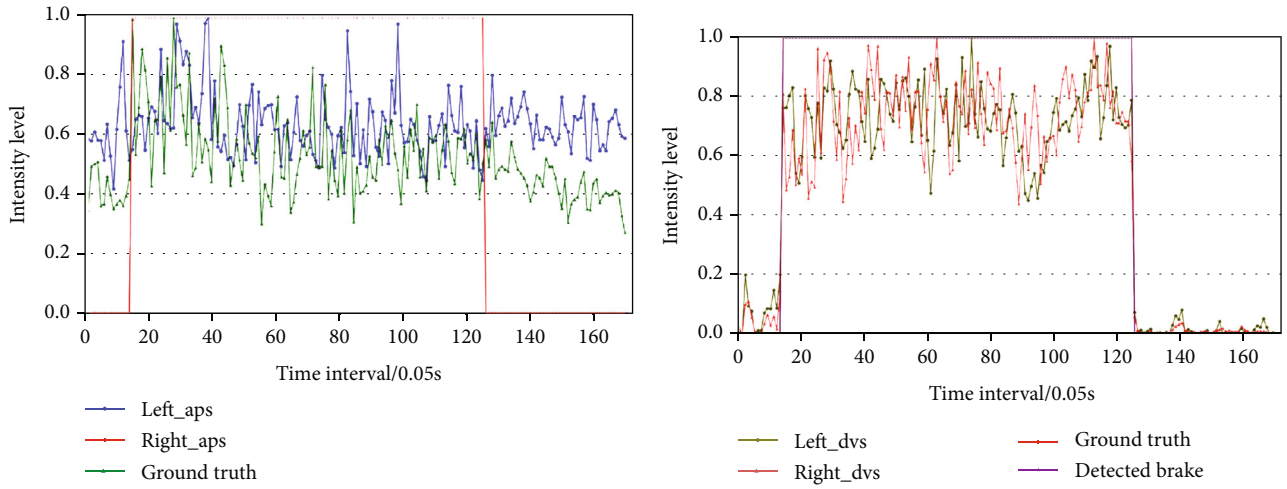


FIGURE 9: The features extracted from APS frames and DVS events on a testing sequence. In this testing sequence, a preceding vehicle changes its taillight state twice. The first time is the transition from braking-off state to braking-on state, and the second time is the transition from braking-on state to braking-off state. The Y-axis represents the feature intensity level, which is the relative values compared to the largest feature obtained in the experiment. The actual taillight state of the preceding vehicle is marked in the red line (ground truth), which has no relationship with the intensity values. (a) The features extracted from APS frames. The blue (left_aps) and green (right_aps) lines represent the features levels of the left and right taillights, respectively. (b) The features extracted from DVS events. The olive (left_dvs) and Indian red (right_dvs) lines represent the feature levels of the left and right taillights, respectively. The purple line (detected brake) is the measured signal state by our method.

3.5. Signal Measurement

3.5.1. Brake-Light Signal Measurement. We utilize a thresholding algorithm for the brake-light signal measurement. It captures the taillight state based on a threshold σ as follows:

$$\text{state}\begin{cases}\text{braking}-\text{on}, & \Omega_{c,\max} > \sigma,\\ \text{braking}-\text{off}, & \text{otherwise.}\end{cases} \tag{8}$$

The measured taillight signal state is braking-on when $\Omega_{c,\max} > \sigma$, vice versa the measured taillight signal state is braking-off. Ideally, the threshold σ is close to 0. As shown in the top row of Figure 7(c), there is almost no gradient change in pixel values of the DVS frame when the brake-light state is OFF, i.e., $\Omega_{c,\max} \approx 0$. However, in real traffic scenarios, some environmental factors, such as noise, vehicle speed, and the distance between sensor and vehicle, will affect the measure of $\Omega_{c,\max}$. Therefore, we reset the σ value based on the real data analysis to attenuate these effects. It is worth noting that a fixed value of σ can meet the measurement requirements of most scenarios since the event stream of DVS can automatically filter out the effect of nontarget regions on frequency. The exception is in extreme scenarios, such as in fog or snow, where fog or snow can severely interfere with the event-based sensor response to changes in the brightness of the brake light. These extreme scenarios are beyond the applicable scope of the proposed system. To improve the brake-light signal measurement accuracy, we combine the measured states of a pair of taillights. Only both the taillight signals on the left and the right are measured as braking-on states; then, the final returned state is braking-on.

3.5.2. Turn-Light Signal Measurement. The transformed feature vectors in Section 3.4.3 are learned by an AdaBoost classifier for the turn-light signal measurement. The AdaBoost classifier linearly combines $a \in m$ different weak classifiers c_a into a single strong one H as follows:

$$H(\mathbf{f}) = \sum_{a=1,\cdots,m} w_a \cdot c_a(\mathbf{f}), \tag{9}$$

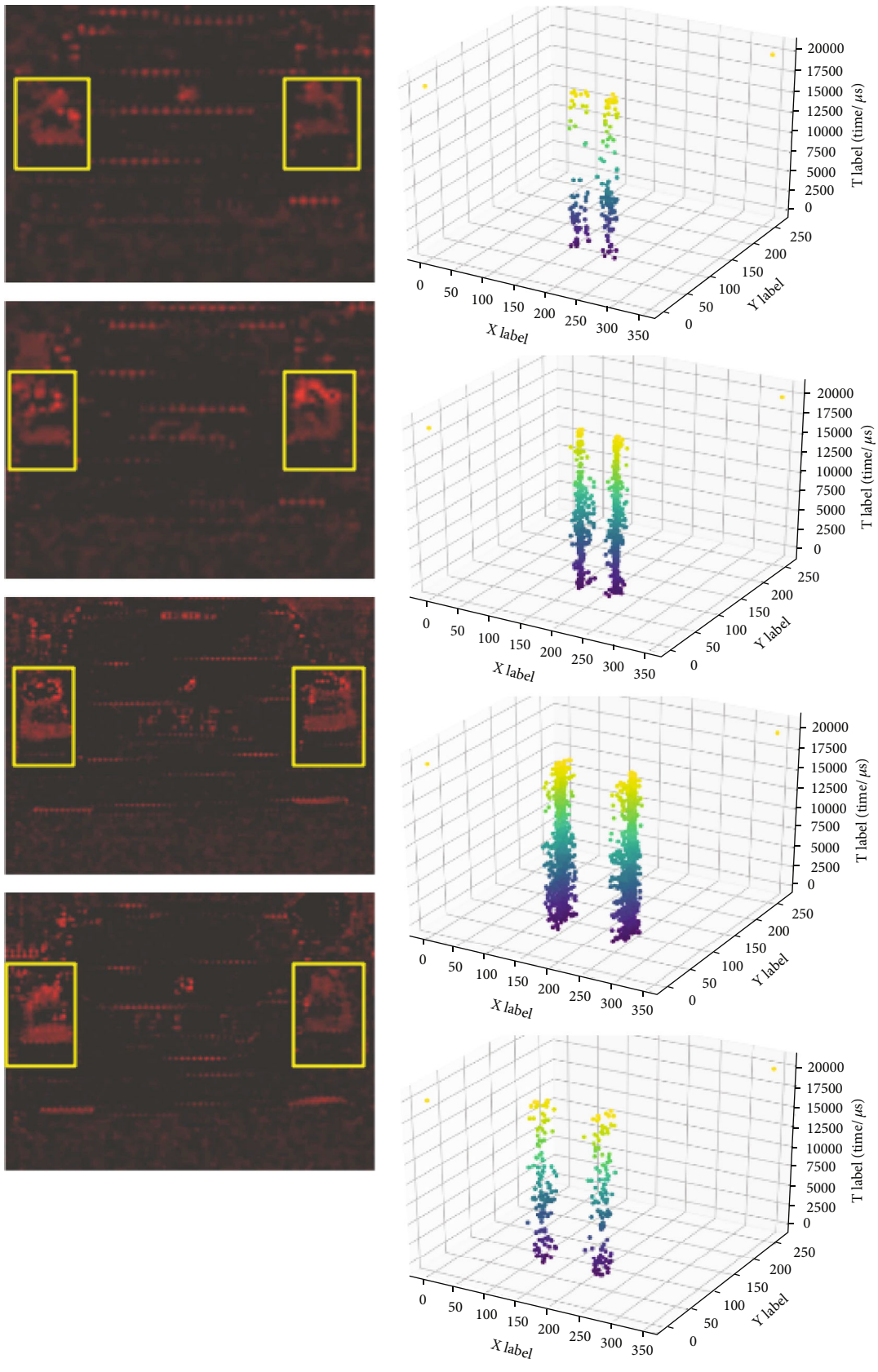
where w_a is the weight of the weak classifier c_a. In our case, a weak classifier c_a is a simple threshold from the feature vectors to split the feature space into two disjunct sets. The state of the vehicle's taillights is identified by integrating the measurement results of a pair of taillights. If and only if one of the taillight pairs is measured as the turning-on state, the final returned state is turning-on. Besides, a flashing turn-light signal is measured after 2 s, because the feature extraction of the turn-light signal is based on a 2s time window.

4. Experimental Results and Discussions

In this section, we conduct experiments to evaluate the performance of the proposed system and discuss its general validity. The experimental data are captured using a front-mounted event-based neuromorphic vision sensor named Color-DAVIS346 in different traffic environments with various light conditions. The recordings contain both APS frames and DVS events. Specifically, the recordings contain open roads and driving scenarios, ranging from urban,

(a) (b)

Figure 10: Continued.



(c) (d)

Figure 10: The qualitative measurement results of the testing sequence in Figure 9. (a) Measurement results. (b) The preceding vehicle detected. (c) The position of the vehicle's taillights located. (d) DVS events for each taillight region.

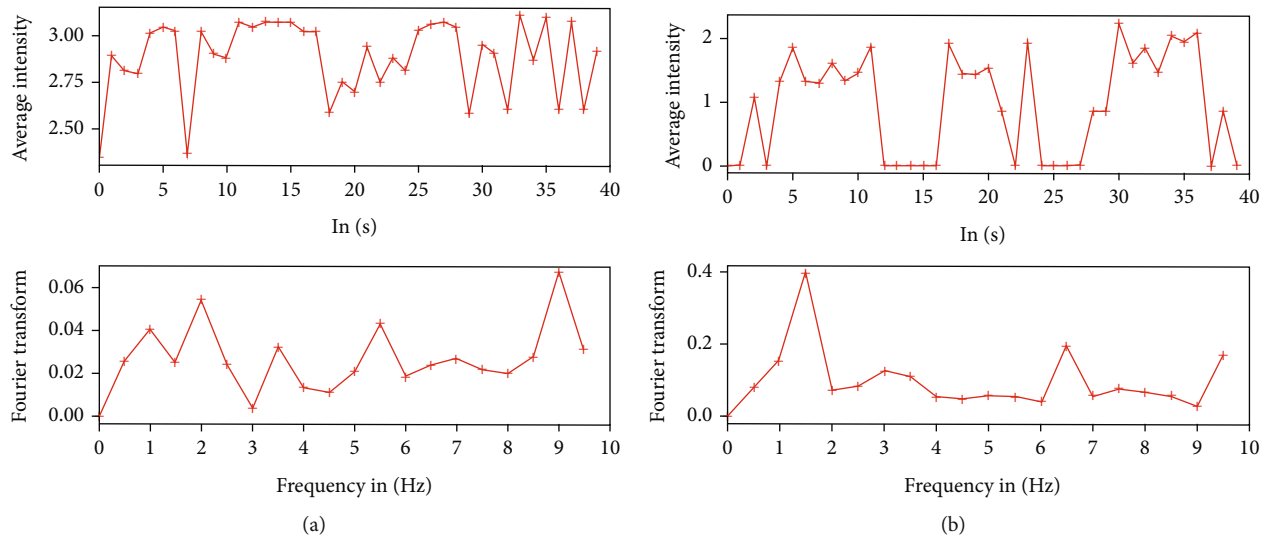


Figure 11: The features of the turn-light signal extracted from APS frames and DVS events in 2 s time interval. The top row is the feature of before the transformation. The bottom row is feature after the transformation, and the DC component is set to zero in order to simplify the visualization. (a) The feature extracted from APS frames. (b) The feature extracted from DVS events.

Table 3: Turn-light signal measurement results.

Real state (column)	Turning-off	Turning-on
Turning-off	89.7%	10.3%
Turning-on	2.3%	97.7%

Turning-on: turning-right-on state or turning-left-on state.

highway, and suburbs, as well as different illumination conditions, including morning, afternoon, and dusk. All annotations are done on APS frames. Manual bounding box annotations of cars and taillights contained in the recordings are provided at a frequency of 20 Hz. We annotated a total of 60 sequences with 26,614 car examples, where 20 sequences are labeled for brake-light signal measurement, while the remaining 40 sequences are labeled for turn-light signal measurement. Each sequence lasts approximately 20 s. For brake-light signal measurement, all 20 sequences are used as the test set. For turn-light signal measurement, we divide 40 sequences into training and test sets based on 7 : 3. The values of these parameters used in the experiment are listed in Table 1. All these parameters are set based on experience and experiments. The performance of the measurement is evaluated using the accuracy metric shown below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{10}$$

where TP and TN are the number of true positives and true negatives, respectively. FP and FN are the number of false positives and false negatives, respectively.

4.1. Brake-Light Measurement Results. For the brake-light signal measurement, the spatial characteristics of the signal of each taillight region are extracted and transformed into the frequency domain, and the maximum value $\Omega_{c,\max}$ in the frequency domain is found as the extracted feature. Figure 8 shows the variation of $\Omega_{c,\max}$ in a testing sequence. Obviously, there are two dominant levels for $\Omega_{c,\max}$ in this sequence. Although $\Omega_{c,\max}$ vibrates significantly due to noises, we can still correctly measure the brake-light signal based on Equation (8) by setting the threshold σ to 0.3 (the red line in Figure 8). As shown in Figure 8, the braking-on state occurs at the 71^{th} frame (the green line in Figure 8), where the first $\Omega_{c,\max}$ greater than the threshold σ appears. Table 2 lists the brake-light signal measurement results of the proposed system on all testing sequences. Each row in the matrix represents the measurement accuracy rate of each state, and the true positives are in italics. As indicated in Table 2, the brake-light signal can be measured with an accuracy higher than 93.4%.

To verify our statement, we analyze the measurement results of a testing sequence. In this testing sequence, a preceding vehicle changes its taillight state twice. The first time is the transition from braking-off state to braking-on state, and the second time is the transition from braking-on state to braking-off state. In this testing experiment, we separately record the feature intensity variations of the two taillights (left and right taillights) extracted from APS frames and DVS events, as shown in Figure 9. Figure 9(a) is the features extracted from APS frames. The blue (left_aps) and green (right_aps) lines indicate the feature levels of the left and right taillights, respectively. Figure 9(b) is the features extracted from DVS events. The olive line (left_dvs) and Indian red (right_dvs) lines indicate the feature levels of the left and right taillights, respectively. Figure 9 also presents the actual state of the vehicle's taillights, which is represented by the red line (ground truth), and the measured signal state using the proposed system is indicated by the purple line (detected brake). As we can see from Figure 9(b), the brake-light is activated within 0.7 s to 6.3 s, and the measured signal state is similar to this. However, if

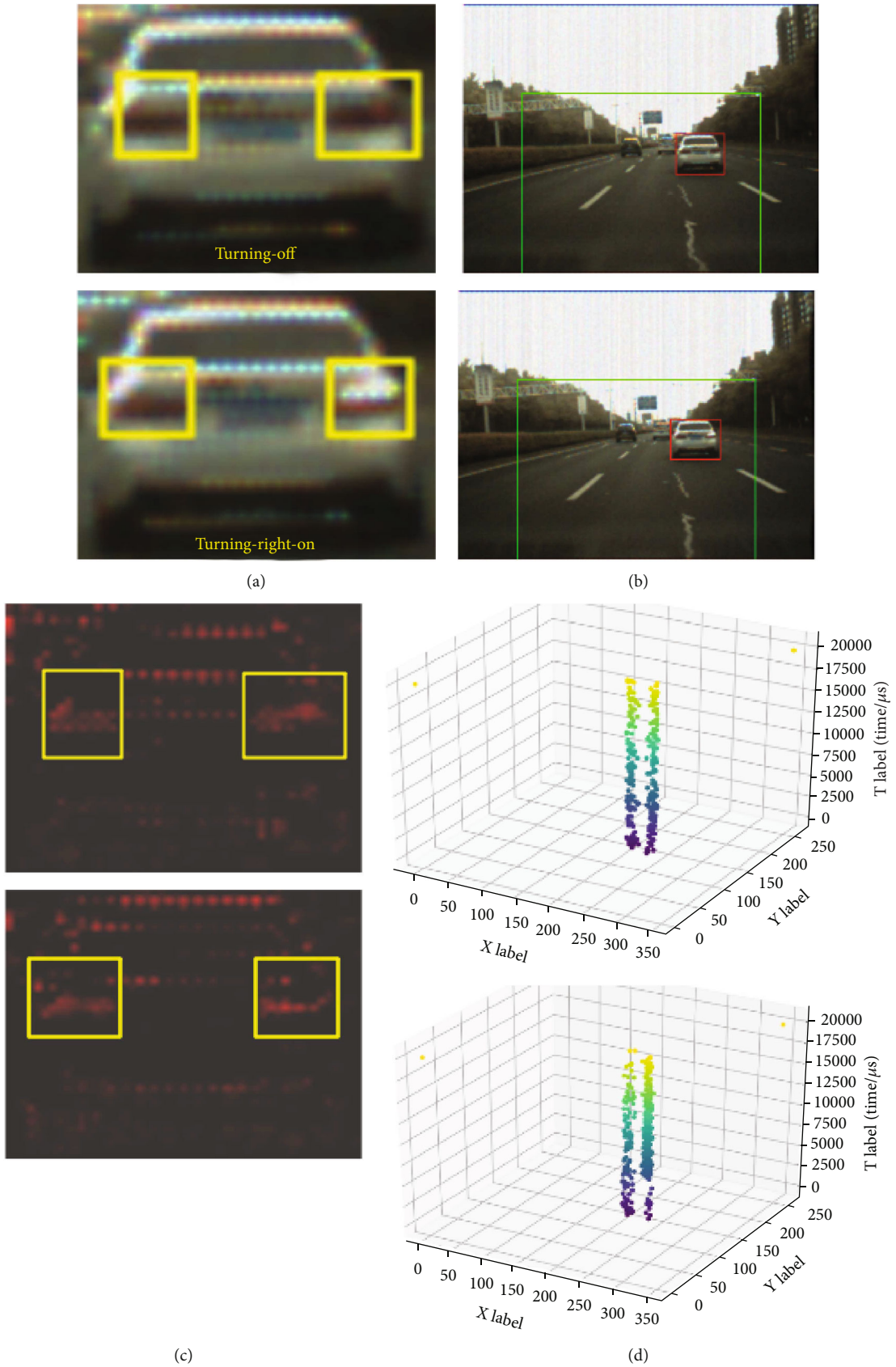


Figure 12: Measurement results from a preceding vehicle. The preceding vehicle switches its taillight state from turning-off state (the top row) to turning-right-on (the bottom row). (a) Measurement results. (b) The preceding vehicle detected. (c) The position of the vehicle's taillights located. (d) DVS events for each taillight region.

we use the features extracted from APS frames (blue and green lines in Figure 9(a)) for the brake-light signal measurement, it is hard to select a suitable threshold. Because the event-based sensor naturally responds to changes in the brightness of a vehicle's taillights, the features extracted from DVS events can more directly reflect the characteristics of the brake-light signal than the features extracted from APS frames.

Figure 10 shows the qualitative measurement results of the above testing sequence. The top row represents the beginning of the testing sequence, and the vehicle's brake lights are not activated. The middle two rows indicate that the vehicle's brake lights are first activated and last activated, respectively. The bottom row is the end of the testing sequence, and the vehicle's brake lights are not activated. Figure 10(a) is the final measurement results of the vehicle taillights. Figures 10(b)–10(d) show the intermediate processing procedures. Figure 10(b) shows the preceding vehicle detected. Figure 10(c) shows the position of the taillights located. Figure 10(d) shows the DVS events for each taillight region. The regions labeled by green bounding boxes are the ROI. The red ones represent the detected preceding vehicle. The yellow ones mean that these regions are the position of the taillights. From the middle two rows of Figure 10(a), we see that the brake lights are activated and measured correctly. From Figure 10(d), we observe a higher event density when the brake lights are activated than when they are not. Comparing the top and the bottom rows of Figure 10(d) and the second and the third rows, we observe that the event density attenuates as the vehicle moves away from the sensor. The attenuation will affect the quality of the brake-light signal features and result in false alarms. Therefore, we mainly measure the brake-light signals of vehicles in the forward direction.

4.2. Turn-Light Measurement Results. For the turn-light signal measurement, the temporal characteristics of the signal of each taillight region are extracted and transformed for feature extraction. Figure 11 shows the features of the turn-light signal extracted from APS frames and DVS events. The top row represents the feature before the transformation, and the bottom row represents the feature after the transformation. Figure 11(a) is the features extracted from APS frames, and Figure 11(b) is the features extracted from DVS events. As can be seen from Figure 11(b), three peaks within a 2 s time window (the top row) correspond to the 1.5 Hz (3 times within 2 s) maximum frequency value (the bottom row) for the feature extracted from DVS events. However, there is no clear frequency characteristic for the feature extracted from APS frames (see Figure 11(a)). This indicates that the transformed feature from DVS events can provide more effective information for the turn-light signal measurement than the feature from synchronous APS frames. Therefore, we use the transformed feature from DVS events for the turn-light signal measurement. Table 3 presents the measurement results by using an AdaBoost classifier. Tuning-on in Table 3 means turning-right-on state or turning-left-on state. As indicated in Table 3, we can measure the turn-light signal after 2 s since it starts to flash, and the accuracy is 93.7%.

For qualitative evaluation, the measurement results of a testing sequence are shown in Figure 12. In this testing sequence, a preceding vehicle switches its taillight state from turning-off (the top row) to turning-right-on (the bottom row). Each column in Figures 10 and 12 has the same meaning. From the bottom row of Figure 12(d), we clearly see that event density in the right taillight region is higher than that in the left taillight region. After analyzing the measurement results of all the testing sequences, we find that the measurement accuracy rate is low when the frequency value of the turn-light signal is less than 1 Hz. Because the feature extraction of the turn-light signal depends on the dynamic characteristics of the brightness of the turn-light signal in 2 s, the number of flashes plays an important role in the accuracy of measurement.

4.3. Discussion for General Validity. Integrating APS frame-based taillight localization and DVS event-based feature extraction (the frequency characteristics of the number of DVS events within one taillight region) based on a neuromorphic vision sensor DAVIS, daytime preceding vehicle taillight measurement in the real environment is targeted, as shown in the experiment results. It is mainly due to the property of event-based sensors that events naturally respond to illumination changes asynchronously, which better reveals the brightness changes of the taillight. Moreover, since the event-based sensor has the advantages of low-motion blur and high dynamic range, the proposed method can guarantee good performance under complicated driving scenarios, e.g., highway or tunnel exit or entrance, and different light conditions, e.g., noontime or dusk. In these scenes, the event-based sensor can effectively capture taillight brightness changes when the standard frame-based camera may fail. On the other hand, due to microsecond temporal resolution for events, some noises, such as measurement distance, camera vibrations, and relative motion, may lead to measuring uncertainty. We alleviate these noise impacts by a presetting measurement ROI, event filtering, and accumulating event frames. Although these techniques reduce the measurement uncertainty to a certain extent, other uncertainty factors (i.e., ambient light) are not paid attention to in this work, which will be one of the key research questions in our next work.

Although the advantages of event sensors have been explored for taillight signal measurement, low spatial resolution, such as the DAVIS346 with a resolution of 346×260, is certainly a limitation for its application. From the point of view of the measurement distance that the system can be adapted to, the low spatial resolution (346×260 used in this work) limits the measurement performance to some extent. However, from the point of view of vision-based measurement system research, a low resolution does not limit the use of the event-based neuromorphic vision sensor in autonomous vehicles. The competitiveness of event-based sensors in autonomous vehicles lies in their main features, not high image quality [63]. These main features include low power and bandwidth consumption and the ability to respond to dynamically changing scenarios. This is also the purpose of this paper, to explore how to use these features of event-

based sensors to measure vehicle taillight signals instead of focusing only on high measurement performance. Taking a broader perspective, event-driven measurement represents an exciting opportunity to enable power-efficient intelligent robots. On the other hand, the trend toward increased spatial resolution on event-based sensors [64, 65] also offers the potential to further improve the measurement performance of the proposed system.

5. Conclusion

In this paper, we propose a novel vision-based autonomous measurement system that can measure the daytime preceding vehicle taillight signal using an event-based neuromorphic vision sensor by analyzing signals in the frequency domain. Unlike traditional approaches that only employ APS frames, we focus on combining APS frames with DVS events. We explore the potential capacity of the event-based neuromorphic vision sensor for the vehicle taillight signal measurement task. Experiments with real traffic scenes demonstrate the performance of the system. The accuracy of the brake-light signal and turn-light signal measurements is 93.4% and 93.7%, respectively, which verifies its feasibility in real-world environments. The results suggest that the event-driven paradigm is a promising line of enquiry. From a research perspective, this paper is more focused on exploring how to leverage and amplify the advantages of event sensors to address the possible limitations, i.e., low dynamic range, interframe information loss, and motion blur, of standard frame-based cameras, while the state-of-the-art performance is a secondary concern. We hope this work can spur researchers to explore more applications of the event-based sensor in the VBM system, such as optical signal measurement for optical communication applications. At the same time, we hope that this work will encourage researchers to add more technologies for visual perception to VBM systems.

Data Availability

The datasets supplied by this work are temporarily unavailable for release due to privacy concerns. The authors can be contacted if necessary, and data can be provided after evaluation.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Shanghai Automotive Industry Sci-Tech Development Program under Grant 1838 and in part by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement 945539 (Human Brain Project SGA3).

References

[1] G. Chen, K. Chen, L. Zhang, L. Zhang, and A. Knoll, "VCANet: vanishing-point-guided context-aware network for small road object detection," *Automotive Innovation*, vol. 4, no. 4, pp. 400–412, 2021.

[2] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi et al., "Yawning detection using embedded smart cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 570–582, 2016.

[3] S. Muhlbacher-Karrer, A. H. Mosa, L.-M. Faller et al., "A driver state detection system—combining a capacitive hand detection sensor with physiological sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 4, pp. 624–636, 2017.

[4] A. Mammeri, T. Zuo, and A. Boukerche, "Extending the detection range of vision-based vehicular instrumentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 856–873, 2016.

[5] Q. Xie, D. Li, Z. Yu, J. Zhou, and J. Wang, "Detecting trees in street images via deep learning with attention module," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5395–5406, 2020.

[6] S. Ali, B. George, L. Vanajakshi, and J. Venkatraman, "A multiple inductive loop vehicle detection system for heterogeneous and laneless traffic," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 1353–1360, 2012.

[7] Y. Liu, G. Chen, and A. Knoll, "Globally optimal vertical direction estimation in Atlanta world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1949–1962, 2022.

[8] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120 db 15 us latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[9] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020.

[10] G. Chen, F. Wang, W. Li et al., "NeuroIV: neuromorphic vision meets intelligent vehicle towards safe driving with a new database and baseline evaluations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1171–1183, 2022.

[11] F. Baghaei Naeini, A. M. AlAli, R. Al-Husari et al., "A novel dynamic-vision-based approach for tactile sensing applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 1881–1893, 2020.

[12] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[13] G. Chen, S. Qu, Z. Li et al., "Neuromorphic vision-based fall localization in event streams with temporal–spatial attention weighted network," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9251–9262, 2022.

[14] H. Cao, G. Chen, Z. Li, Y. Hu, and A. Knoll, "NeuroGrasp: multimodal neural network with Euler region regression for neuromorphic vision-based grasp pose estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[15] C. Caraffi, T. Vojır, J. Trefny, J. Sochman, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 975–982, Anchorage, AK, USA, 2012.

[16] L. Chen, X. Hu, T. Xu, H. Kuang, and Q. Li, "Turn signal detection during nighttime by CNN detector and perceptual hashing tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3303–3314, 2017.

[17] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 453–462, 2010.

[18] Y.-M. Chan, S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, "Vehicle detection under various lighting conditions by incorporating particle filter," in *2007 IEEE Intelligent Transportation Systems Conference*, pp. 534–539, Bellevue, WA, USA, 2007.

[19] C.-L. Jen, Y.-L. Chen, and H.-Y. Hsiao, "Robust detection and tracking of vehicle taillight signals using frequency domain feature based adaboost learning," in *2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 423-424, Taipei, Taiwan, 2017.

[20] B. Frohlich, M. Enzweiler, and U. Franke, "Will this car change the lane?- turn signal recognition in the frequency domain," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 37–42, Dearborn, MI, USA, 2014.

[21] D.-Y. Chen and Y.-J. Peng, "Frequency-tuned taillight-based nighttime vehicle braking warning system," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3285–3292, 2012.

[22] D.-Y. Chen and Y.-H. Lin, "Frequency-tuned nighttime brake-light detection," in *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 619–622, Darmstadt, Germany, 2010.

[23] C.-L. Chien, H.-M. Hang, D.-C. Tseng, and Y.-S. Chen, "An image based overexposed taillight detection method for frontal vehicle detection in night vision," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–9, Jeju, Korea (South), 2016.

[24] N. Boonsim and S. Prakoonwit, "An algorithm for accurate taillight detection at night," *International Journal of Computer Applications*, vol. 100, no. 12, pp. 31–35, 2014.

[25] L.-C. Chen, J.-W. Hsieh, S.-C. Cheng, and Z.-R. Yang, "Robust rear light status recognition using symmetrical surfs," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2053–2058, Gran Canaria, Spain, 2015.

[26] D.-Y. Chen, Y.-J. Peng, L.-C. Chen, and J.-W. Hsieh, "Nighttime turn signal detection by scatter modeling and reflectance-based direction recognition," *IEEE Sensors Journal*, vol. 14, no. 7, pp. 2317–2326, 2014.

[27] X. Wang, J. Tang, J. Niu, and X. Zhao, "Vision-based two-step brake detection method for vehicle collision avoidance," *Neurocomputing*, vol. 173, pp. 450–461, 2016.

[28] M.-Y. Chern and P.-C. Hou, "The lane recognition and vehicle detection at night for a camera-assisted car on highway," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, pp. 2110–2115, Taipei, Taiwan, 2003.

[29] Z. Cui, S.-W. Yang, and H.-M. Tsai, "A vision-based hierarchical framework for autonomous front-vehicle taillights detection and signal recognition," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 931–937, Gran Canaria, Spain, 2015.

[30] A. Almagambetov, S. Velipasalar, and M. Casares, "Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3732–3741, 2015.

[31] I. Cabani, G. Toulminet, and A. Bensrhair, "Color-based detection of vehicle lights," in *IEEE Proceedings. Intelligent Vehicles Symposium*, pp. 278–283, Las Vegas, NV, USA, 2005.

[32] A. Almagambetov, M. Casares, and S. Velipasalar, "Autonomous tracking of vehicle rear lights and detection of brakes and turn signals," in *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications*, pp. 1–7, Ottawa, ON, Canada, 2012.

[33] M. Casares, A. Almagambetov, and S. Velipasalar, "A robust algorithm for the detection of vehicle turn signals and brake lights," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pp. 386–391, Beijing, China, 2012.

[34] H.-T. Chen, Y.-C. Wu, and C.-C. Hsu, "Daytime preceding vehicle brake light detection using monocular vision," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 120–131, 2016.

[35] F. I. Vancea, A. D. Costea, and S. Nedevschi, "Vehicle taillight detection and tracking using deep learning and thresholding for candidate generation," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 267–272, Cluj-Napoca, Romania, 2017.

[36] J.-G. Wang, L. Zhou, Y. Pan et al., "Appearance-based brake-lights recognition using deep learning and vehicle detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 815–820, Gothenburg, Sweden, 2016.

[37] G. Zhong, Y.-H. Tsai, Y.-T. Chen et al., "Learning to tell brake lights with convolutional features," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1558–1563, Rio de Janeiro, 2016.

[38] H.-K. Hsu, Y.-H. Tsai, X. Mei et al., "Learning to tell brake and turn signals in videos using cnn-lstm structure," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, Yokohama, Japan, 2017.

[39] K.-H. Lee, T. Tagawa, J.-E. M. Pan, A. Gaidon, and B. Douillard, "An attention-based recurrent convolutional network for vehicle taillight recognition," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2365–2370, Paris, France, 2019.

[40] G. K. Cohen, *Event-Based Feature Detection, Recognition and Classification, [Ph.D. thesis]*, Western Sydney University (Australia), 2015.

[41] G. Chen, P. Liu, Z. Liu et al., "Neuroaed: towards efficient abnormal event detection in visual surveillance with neuromorphic vision sensor," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 923–936, 2021.

[42] G. Chen, F. Wang, X. Yuan, Z. Li, Z. Liang, and A. Knoll, "NeuroBiometric: an eye blink based biometric authentication system using an event-based neuromorphic vision sensor," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 206–218, 2021.

[43] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision-processing systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 9, pp. 1064–1071, 1999.

[44] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1964–1980, 2019.

[45] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "Endto-end learning of representations for asynchronous event-based data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5633–5643, 2019.

[46] J. Lee, T. Delbruck, P. K. J. Park et al., "Live demonstration: gesture-based remote control using stereo pair of dynamic vision sensors," in *2012 IEEE International Symposium on Circuits and Systems*, pp. 741–745, Seoul, Korea (South), 2012.

[47] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, "HOTS: a hierarchy of event-based time-surfaces for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.

[48] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. B. Benosman, "HATS: histograms of averaged time surfaces for robust event-based object classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1731–1740, 2018.

[49] R. B. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 407–417, 2014.

[50] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: selfsupervised optical flow estimation for event-based cameras," 2018, https://arxiv.org/abs/1802.06898.

[51] A. Rigi, F. B. Naeini, D. Makris, and Y. H. Zweiri, "A novel eventbased incipient slip detection using dynamic active-pixel vision sensor (DAVIS)„" *Sensors (Basel, Switzerland)*, vol. 18, 2018.

[52] T. Taunyazov, W. Sng, H. H. See et al., "Event-driven visual-tactile sensing and learning for robots," 2020, https://arxiv.org/abs/2009.07083.

[53] R. Muthusamy, X. Huang, Y. Zweiri, L. Seneviratne, and D. Gan, "Neuromorphic event-based slip detection and suppression in robotic grasping and manipulation," *IEEE Access*, vol. 8, pp. 153364–153384, 2020.

[54] X. Huang, R. Muthusamy, E. Hassan et al., "Neuromorphic vision based contact-level classification in robotic grasping applications," *Sensors*, vol. 20, no. 17, p. 4724, 2020.

[55] F. Baghaei Naeini, D. Makris, D. Gan, and Y. Zweiri, "Dynamic-vision-based force measurements using convolutional recurrent neural networks," *Sensors*, vol. 20, no. 16, p. 4469, 2020.

[56] K. Kumagai and K. Shimonomura, "Event-based tactile image sensor for detecting spatio-temporal fast phenomena in contacts," in *2019 IEEE World Haptics Conference (WHC)*, pp. 343–348, Tokyo, Japan, 2019.

[57] A. F. Abbas, U. U. Sheikh, F. T. AL-Dhief, and M. N. H. Mohd, "A comprehensive review of vehicle detection using computer vision," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 3, pp. 838–850, 2021.

[58] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: a survey," 2019, https://arxiv.org/abs/1905.05055.

[59] Z. Cai and N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2017.

[60] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-bydetection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, Lecce, Italy, 2017.

[61] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, *Real-Time Visualinertial Odometry for Event Cameras Using Keyframe-Based Nonlinear Optimization*, University of Zurich University Library, 2017.

[62] J. D. J. Rangel-Magdaleno, R. D. J. Romero-Troncoso, R. A. Osornio-Rios, E. Cabal-Yepez, and A. Dominguez-Gonzalez, "FPGA-based vibration analyzer for continuous CNC machinery monitoring with fused FFT-DWT signal processing," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3184–3194, 2010.

[63] J. A. Leñero-Bardallo, R. Carmona-Galán, and A. Rodríguez-Vázquez, "Applications of event-based image sensors—review and analysis," *International Journal of Circuit Theory and Applications*, vol. 46, pp. 1620–1630, 2018.

[64] Y. Suh, S. Choi, M. Ito et al., "A 1280×960 dynamic vision sensor with a 4.95-μm pixel pitch and motion artifact minimization," in *2020 IEEE international symposium on circuits and systems (ISCAS)*, pp. 1–5, Seville, Spain, 2020.

[65] G. Gallego, T. Delbruck, G. Orchard et al., "Event-based vision: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.

[66] G. Chen, H. Cao, M. Aafaque et al., "Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system," *Journal of Advanced Transportation*, vol. 2018, Article ID 4815383, 13 pages, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 9681455, 11 pages
https://doi.org/10.1155/2022/9681455

Research Article

The Robust Semantic SLAM System for Texture-Less Underground Parking Lot

Chongjun Liu and Jianjun Yao

College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Chongjun Liu; chongjunliu@hrbeu.edu.cn

Received 21 October 2021; Revised 9 April 2022; Accepted 19 April 2022; Published 6 July 2022

Academic Editor: Peng Hang

Automatic valet parking (AVP) is the autonomous driving function that may take the lead in mass production. AVP is usually needed in an underground parking lot, where the light is dim, the parking space is narrow, and the GPS signal is denied. The traditional visual-based simultaneous location and mapping (SLAM) algorithm suffers from localization loss because of inaccurate mapping results. A new robust semantic SLAM system is designed mainly for the dynamic low-texture underground parking lot to solve the problem mentioned. In this system, a 16-channel Lidar is used to help the visual system build an accurate semantic map. Four fisheye cameras mounted at the front, back, left, and right of the vehicle are also used to produce the bird's eye view picture of the vehicle by joint calibration. The vehicle can localize itself and navigate to the target parking lot with the semantic segmented picture and the preobtained semantic map. Based on the experiment result, the proposed AVP-SLAM solution is robust in the underground parking lot.

1. Introduction

The public traffic jam situation worsens with the increasing number of automobiles. Researchers in the automobile field are now devoting their effort to automatic driving systems to ease traffic pressure and present a safe way for driving. As one of the most promising and meaningful functions in automatic driving, the automatic valet parking (AVP) system has become the focus of scholars because it can provide drivers, particularly the new ones, an achievable and safe way to park vehicles under crowded parking conditions. This function can be achieved by providing vehicles with a high-definition (HD) map for vehicle path planning. Thus, the AVP function is achievable if the HD map and the global positioning system-inertial measurement unit (GPS-IMU) camera-based localization method can be used to locate a vehicle at a preknown place. However, a vehicle cannot possibly acquire environmental knowledge when this vehicle is located in an unknown place. A vehicle must locate itself and build an environment map while moving by itself to overcome the difficulty mentioned. Therefore, the SLAM problem was proposed in 1986 [1].

SLAM technology can be divided into two categories, namely, Lidar and Vision, depending on the sensors used. Lidar-based SLAM schemes are extensively analyzed by researchers [2]. Lidar can measure the angle and distance of obstacle points with higher accuracy, which is convenient for positioning and navigation. Lidar-based SLAM has high accuracy and no cumulative error when building maps. Excellent performance and dense point clouds can be obtained using the 3D Lidar. However, the 3D Lidar with 64 channels is expensive for commercialization [3]. The corridor of the underground parking lot is long and straight, with smooth walls on both sides, and it is easy to lose positioning only by relying on Lidar-based SLAM. Therefore, the vision-SLAM system receives attention from researchers worldwide because of its high perception ability and low cost.

In addition to vision-SLAM, other traditional feature methods and road-based feature methods are available. In traditional feature methods, sparse points, lines, and dense

planes in a real environment are taken as geometrical features, which can be used for vehicle localization [4]. Furthermore, corner features are widely used for visual mileage calculation [5–7]. The pose of the camera and feature positions can be estimated with these methods. Moreover, SIFT [8], SURF [9], BRIEF [10], and ORB [11] descriptors are widely used by researchers in describing the features to make these features unique. ORB-SLAM [12, 13] is a representative SLAM framework based on nonlinear optimization. The ORB features are used as tracking feature points while driving.

For the methods based on road features, lane lines, curbs, and traffic signs are widely used as landmarks, which can be used to localize the camera pose by comparing the landmarks with previously established maps. Compared with the traditional-feature-based method, road-feature-based methods use these landmarks, which are robust even the illumination conditions change. Yan [14] proposed a nonlinear optimization problem to localize a 6-DOF camera pose in terms of localization. In this method, the geometry of road markings and the odometry and epipolar geometry constraints of the vehicle were considered. The experiment results showed that submeter localization error is achieved on the road with sufficient road markings. Schreiber et al. [15] came up with a novel approach to establish precise and robust localization by using a stereo camera system and a highly accurate map with curbs and road markings. In this method, global navigation satellite systems are used only to obtain the initial location, and they are not used during a 50 km test. Ranganathan et al. [16] presented a new scheme for precise localization. In this scheme, the signs marked on the road were used to localize the automobile in a global coordinate. Furthermore, the mechanism combining road-mark-based map and sparse-feature-based map was adopted to obtain a high localization accuracy. In addition to location, many studies focus on mapping. Rehder et al. [17] proposed a novel approach to generate the local grid map by detecting the lane on the image taken by a camera. A globally consistent map can be constructed with the help of the local grid map. Jeong et al. [18] proposed a road-SLAM algorithm by considering the road markings obtained from images taken by a camera. In this algorithm, the random forest method was used to improve the matching accuracy by using a submap containing road information. Based on the experiment results, the accuracy of this mapping method can be improved to 1.098 m over 4.7 km of the path length. This result was validated by comparing the obtained data with the data from RTK-GPS.

In addition to pure vision odometer, vision-aided inertial navigation algorithm is becoming increasingly popular in the autonomous driving field. In this scheme, IMU is added into a vision-based scheme to improve localization precision. Mourikis et al. [19] proposed an extended Kalman filter-based algorithm for real-time vision-aided inertial navigation. The result showed that a very accurate pose estimation can be conducted with this sensor-fusing algorithm. Leutenegger et al. [6] came up with a keyframe-based visual-inertial odometry scheme. Although this scheme demands considerable computation resources, superior accuracy performance was obtained. The monocular visual-inertial system is the most commonly used VIO algorithm at present [6, 12]. In this scheme, a camera and a low-cost IMU are used to obtain high-accuracy localization.

Semantic segmentation is a new image clustering task at the pixel level, and it is widely applied in perception in the automatic driving domain and medical image diagnosis [20–23]. In recent years, deep convolution neural network has been widely used in semantic segmentation tasks [24], and the majority of the networks are based on various convolution network structures. Among them, U-Net [20, 25] is widely accepted and improved as the basic network that can be trained with pictures taken by a camera and could classify pixels of the pictures into parking lines and signs. The basic framework of U-Net is shown in Figure 1. These classified results are critical data for building maps or localizing automobiles. The residual network [26] can achieve good results to adapt to highly complex segmentation scenarios through a very deep layer depth and a large number of parameters. However, lightweight networks, such as ERF-Net [27], consider the real-time performance and accuracy with the method of distillation [28] to be deployed to edge computing devices, such as onboard computers.

Automotive valet parking is a complex function that can be equipped on vehicles and help drivers, particularly new drivers, park their cars in a carport. However, the light conditions in underground parking lots are usually very dim, and smooth walls, floors, and columns can be found inside. All these conditions complicate the parking task. Moreover, traditional vision equipment is influenced and becomes unfit in this scenario. In order to solve the above problems, we first build a vehicle platform with a 16-channel Lidar and four surrounding cameras. The robot operating system (ROS) is adopted to call both the camera and Lidar for collecting data. Also, a method consisting of image semantic segmentation, lidar supplemental mapping, semantic mapping, and localization is proposed. The rest of this paper is organized as follows. The detailed system architecture is introduced in Chapter 2. Then, the methodology proposed in this study is described in chapter 3. Finally, the experiment results are presented to show the robustness of the proposed AVP-SLAM solution in chapter 4, and our conclusion is drawn in chapter 5.

2. System Architecture

Four surround-view cameras and a 16-channel Lidar are applied in the proposed mapping and localization system, as shown in Figure 2. The framework for this system consists of two parts. One is offline mapping, and the other is for localization. For the offline mapping system, the 16-channel Lidar is used to provide the odometry and build the point cloud map. The semantic information is added to this map by keyframe matching. We select Lidar keyframes at 0.2 seconds intervals. We use ROS to call both the camera and Lidar for collecting data, so we have their own time stamps in the header of the message. Based on the time stamp, we select the semantic map corresponding to the Lidar to overlay the semantic map according to the position and posture of the

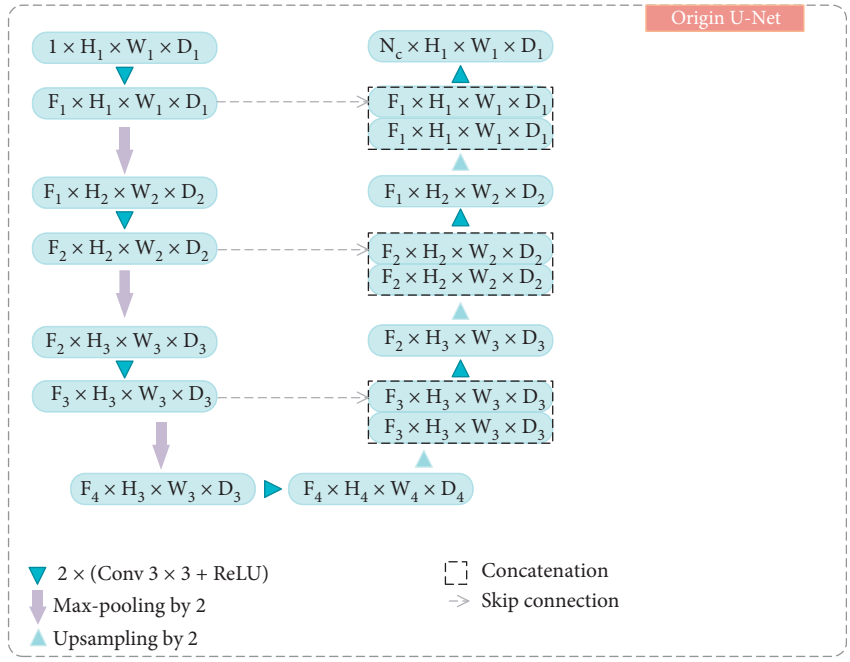


Figure 1: Basic framework of U-Net.

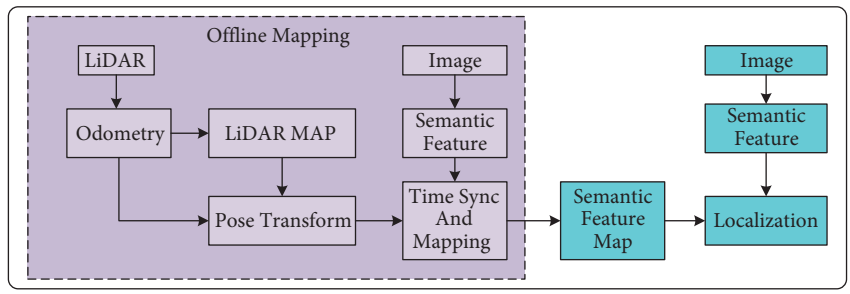


Figure 2: System summary.

Lidar frames. We drive a data collection vehicle across the road in an underground parking lot, select 3707 point cloud keyframes, and then select the corresponding image in the simultaneously recorded image data. The ORB features are extracted from the global map to build the visual semantic dataset and obtain the initial pose for localization. The global map is divided into several zones, and the dataset is established based on the number of character types. Furthermore, the dictionary can be built by zones. The initial pose can be determined with ORB features of the semantic image. Then, the localization can be done with the pose data in the last frame and the obtained real-time semantic data.

3. Semantic Mapping and Localization Methodology

3.1. Image Processing. Four surround-view cameras are used in this project. The position and visual angle of each camera should be adjusted to have a good surrounding picture of the vehicle, as shown in Figure 3. The four purple points on the vehicle are cameras with the fish lens that looks downward. The dashed line is the field of view for each camera. Figure 3 shows that four overlapping areas exist between every two adjacent cameras. Thus, the camera should be calibrated offline, and the weight of each camera should be appropriately set to integrate four separate pictures into one picture. The well-calibrated result can be seen in Figure 4(a). The synthesized picture taken by the camera during driving in the underground parking lot is shown in Figure 4(b). The results shown in Figure 4 indicate that the cameras are well-calibrated to provide enough visual information that can be used to localize the vehicle.

3.2. Image Semantic Segmentation. After theoretical exploration and practical verification, this study adopts U-Net with attention mechanism [29] to perform semantic segmentation tasks, which can make the network sensitive to the characteristics of specific locations. Data enhancement method is also used in increasing the training samples to overcome the disadvantages caused by the size limitation of the dataset. Attention coefficient $\alpha \in [0, 1]$ preserves the activation for specific tasks by identifying remarkable image regions and simplifying feature responses. The output of the

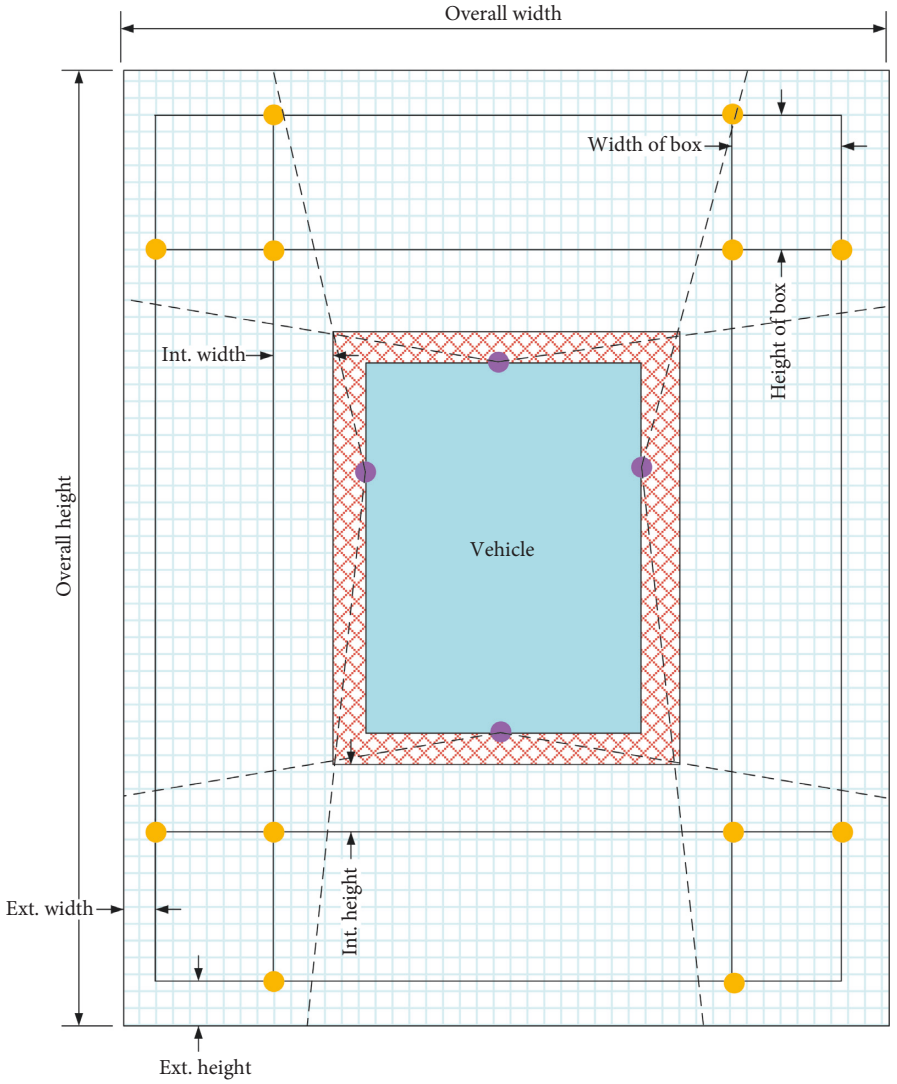


Figure 3: Configuration of four surround-view cameras used in the AVP-SLAM system.

(a) (b)

Figure 4: Synthesized picture. (a) Well-calibrated synthesized pictures. (b) Synthesized picture during driving.

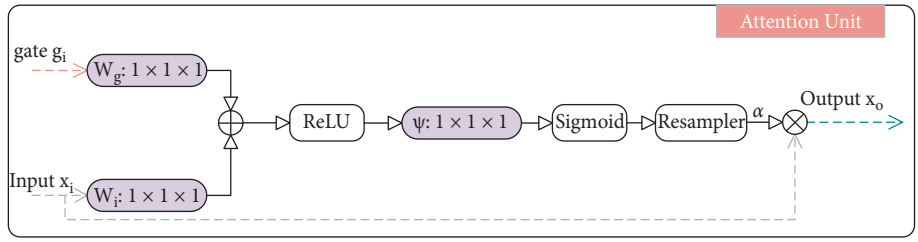


Figure 5: Schematic diagram of attention unit.

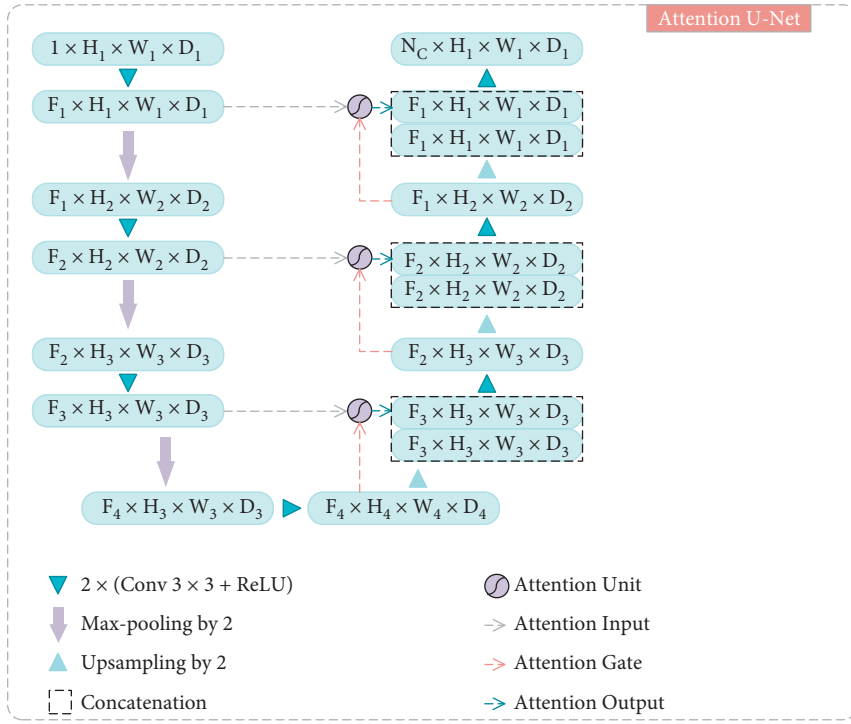


Figure 6: Structure of ATT-U-Net.

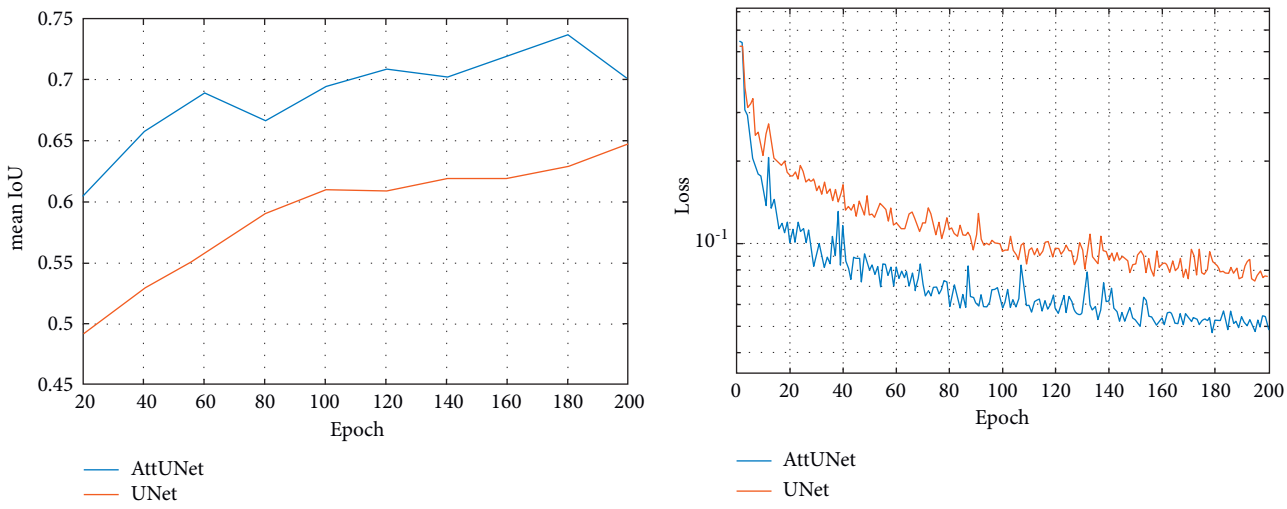
unit is the multiplication of the characteristic graph input and the attention coefficient. Each input pixel matrix is $x_i \in \mathbb{R}^{F_L}$, which has a corresponding single-scale feature, and F_L presents the number of feature maps at layer L. Feature-map x_L is obtained at the output of layer L by sequentially applying a linear transformation. For multiple segmentation classes, multidimensional attention coefficients can be used to learn a classified subset of objects in each dimension. A gate vector $g_i \in \mathbb{R}^{F_g}$ can be used to determine the high-attention area by acting on each pixel I. Gate coefficient can be obtained using the additional attention, which can be expressed as follows [27, 28]:

$$\begin{aligned} q_{\text{att}}^{l} &= \psi^T\left(\sigma_1\left(W_x^T x_i^L + W_g^T g_i + b_g\right)\right) + b_\psi, \\ \alpha_i^L &= \sigma_2\left(q_{\text{att}}^L\left(x_i^L, g_i; \Theta_{\text{att}}\right)\right). \end{aligned} \tag{1}$$

$\sigma_2(x_i, c) = (1/1 + e^{-x_{i,c}})$ represents sigmoid activation function. The attention unit is defined by the parameter set Θ_{att}, including linear transformation $W_x \in \mathbb{R}^{F_L \times F_{int}}$, $W_g \in \mathbb{R}^{F_g \times F_{int}}$, $\psi \in \mathbb{R}\hat{}(F_{\text{int}} \times 1)$, and offset $b_\psi \in \mathbb{R}$, $b_g \in \mathbb{R}^{F_{\text{int}}}$. The structure of the attention unit is shown in Figure 5. The structure of the attention unit added U-Net (ATT U-Net) is shown in Figure 6. The convolution parameter updating the rules of L-1 layer is as follows. The function $f(x^L; \Phi^L) = x^{L+1}$ applied in convolution layer L is characterised by trainable kernel parameters Φ^L.

$$\begin{aligned} \frac{\partial\left(x_i^L\right)}{\partial\left(\Phi^{L-1}\right)} &= \frac{\partial\left(\alpha_i^L f\left(x_i^{L-1}; \Phi^{L-1}\right)\right)}{\partial\left(\Phi^{L-1}\right)} \\ &= \alpha_i^L \frac{\partial\left(f\left(x_i^{L-1}; \Phi^{L-1}\right)\right)}{\partial\left(\Phi^{L-1}\right)} + \frac{\partial\left(\alpha_i^L\right)}{\partial\left(\Phi^{L-1}\right)} x_i^L. \end{aligned} \tag{2}$$

The first gradient term α_i^L on the right is scaled. In the multidimensional attention unit, α_i^L corresponds to a vector containing each grid scale. In each subattention unit, supplementary information is extracted and fused to define the output of the residual connection. In order to reduce the number of training parameters and the computational complexity of attention units, the linear transformation $(1 \times 1 \times 1)$ without any spatial support is implemented, and



(a) (b)

Figure 7: Mean IoU and loss curve during training with 200 epoch.

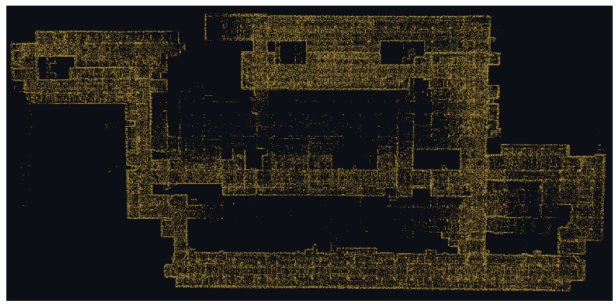
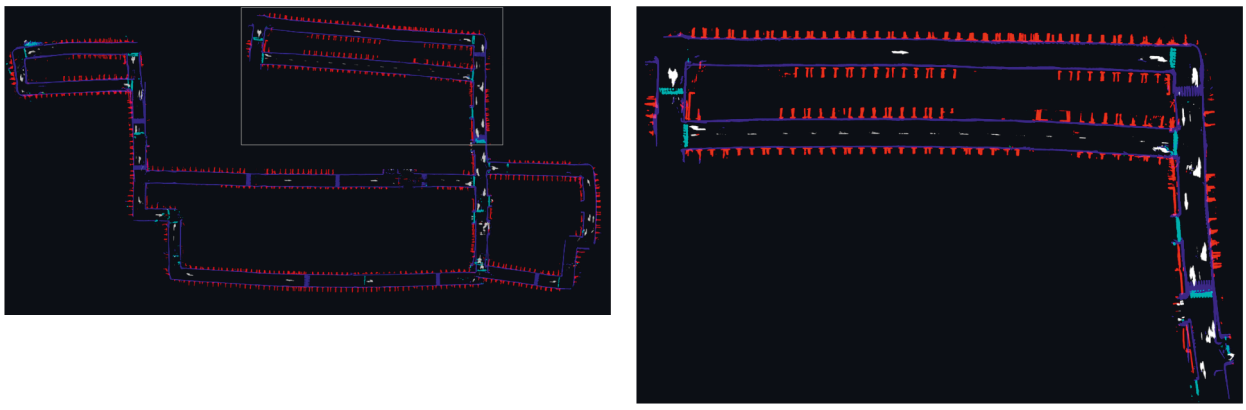


Figure 8: Point cloud map by Lidar.



(a) (b)

Figure 9: Optimized semantic map. (a) Global view. (b) Zoom-in view.

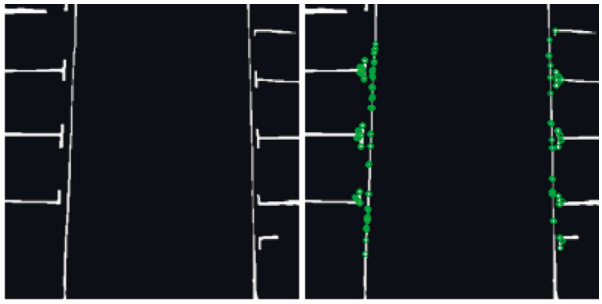
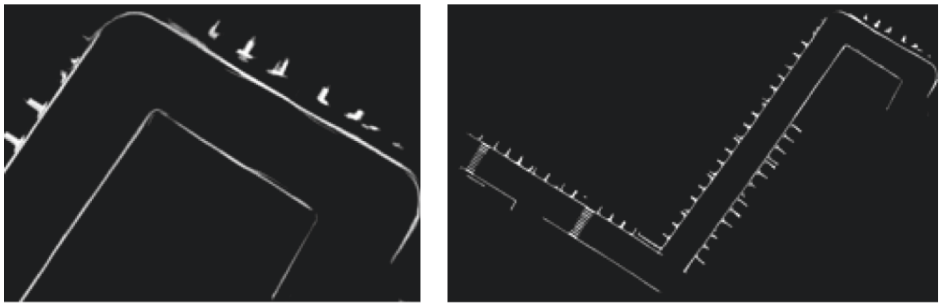


Figure 10: ORB feature detection result.



(a) (b)

Figure 11: (a) Area map with the highest ORB matching scores. (b) Local map made by ICP method.

the input characteristic map is downsampled to the same resolution as the gated signal. The relevant linear transformation couples the feature diagrams and rearranges them to the low dimensional space to implement the gating operation. Low-dimensional feature maps, such as the first residual connection, do not perform a gating function, because they cannot represent input data in high-dimensional space. We use depth supervision to force the medium feature map to be semantically recognizable at each image scale, which ensures that the attention unit has the ability to affect the response of a wide range of image foreground content at different scales.

A performance comparison between ATT U-Net and the original U-Net is shown in Figure 7. As shown in Figure 7, the decrease of loss and the IoU performance when ATT-U-Net is used are faster and better than those when traditional U-Net is used. Cross-entropy loss was used here.

3.3. Lidar Supplemental Mapping. In this study, a 16-channel Lidar is used to build the segmentation map supplementarily. The SC-LeGO-LOAM framework [30–32] is used to assist in building the map. ROS/C++ is selected as the code framework. An image-based segmentation method [33] divides the distance map made by Lidar into multiple groups of clusters, and classes with less than 30 points are discarded as environmental noise to improve the efficiency of processing and the accuracy of feature extraction. The mark (ground point or segmentation point), coordinates in the distance graph, and the distance to the sensor for each point can be obtained by segmentation. These characters of the ground and segmentation points are used for character extraction. In the loop detection of Lidar-based SLAM, the scan context descriptor encodes the radar point cloud and scores the similarity of loop detection. The established Lidar point cloud map after a series of optimization is shown in Figure 8, providing pose information for the construction of the semantic map.

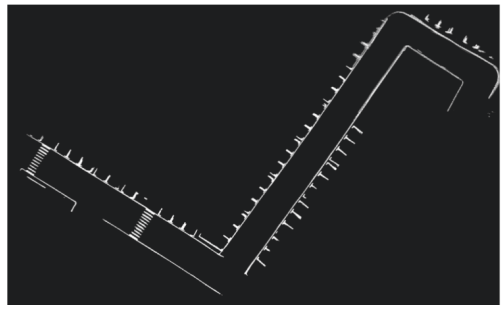


Figure 12: Global view of the matched area map within the local map.

3.4. Semantic Mapping. After obtaining the point cloud map, we can obtain the pose information of every frame with high accuracy. The semantic map and Lidar map with the same timestamp are combined based on the pose information of every Lidar frame. After being matched, the pose transformation between keyframes is used to accumulate the semantic map and obtain pose transformation frames. The established semantic maps obtain the overlapped parts because of the pose data error for point cloud and matching error. The iterative closest point (ICP) algorithm is used in

Table 1: The IoU result.

Lane line	Parking line	Speed bump	Traffic signs	Average IoU
0.78	0.68	0.84	0.78	0.77

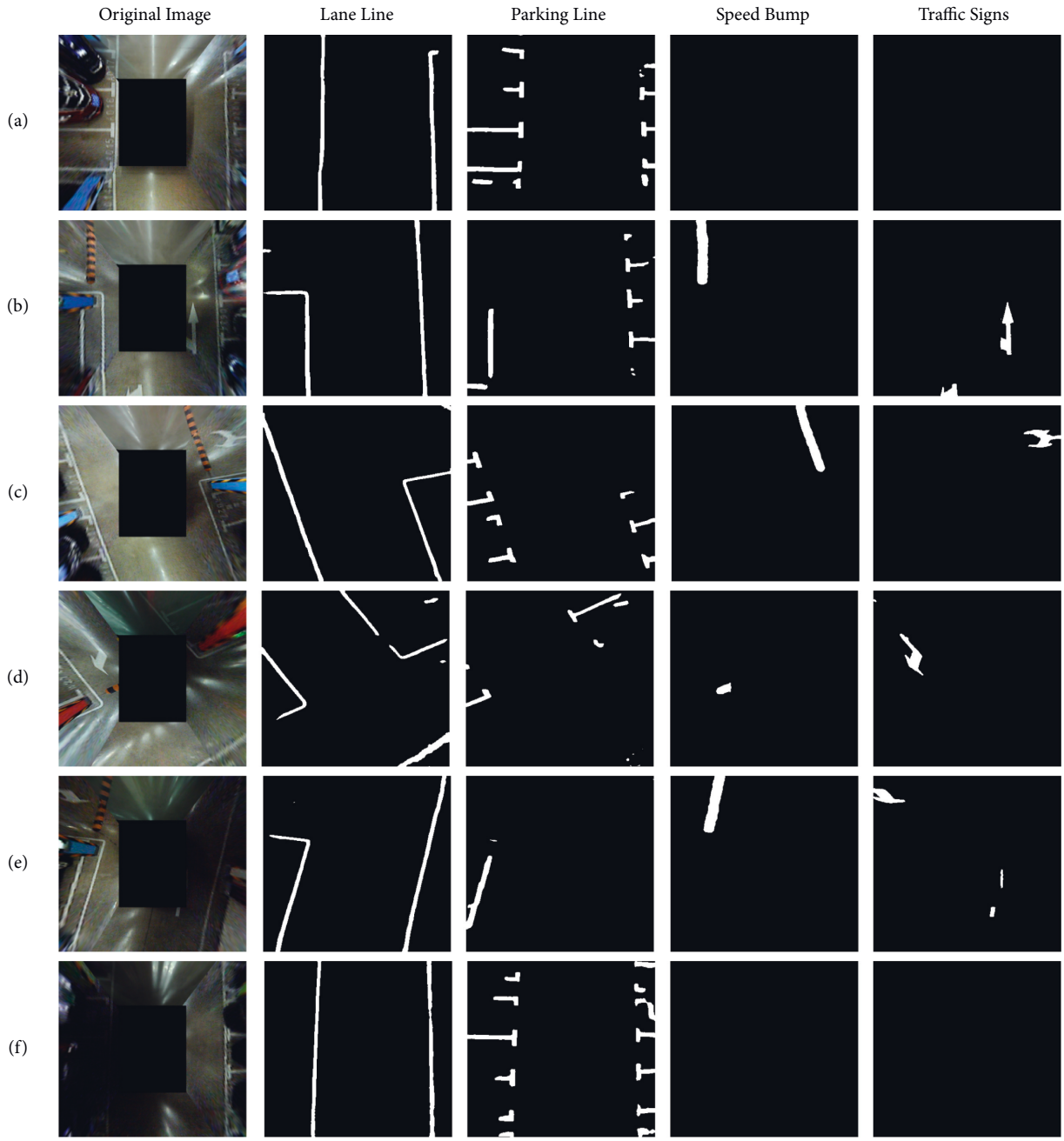


Figure 13: Semantic segmentation result.

calculating the best fusion transformation for the overlapped parts between two maps to solve the problem mentioned. The optimized global map is shown in Figure 9.

3.5. Semantic Localization. The ORB features are extracted from the global map and used to make a visual semantic word bag to determine the initial position and posture. The global map is divided into regions. The word bags and dictionaries are established based on the number of the feature points and regions, respectively. Then, the ORB features are extracted from the initial semantic image input, and a word bag is built. This word bag is used to score the similarity in the dictionary, and the area where the vehicle is located is determined by the score. The ORB feature detection is shown in Figure 10. After the vehicle localization

area is obtained using ORB feature matching, the ICP method is used to overlay several continuous semantic maps into a new one. The local map localization process is shown in Figures 11 and 12. The purple marks are the matched pixel points, whereas the green points and the white points in Figure 12 are the pixel points for the local map and the area map, respectively. After the location of the local map is obtained, the semantic map of the current frame is used to match with the local map and obtain the vehicle posture and location. The new local map from the global map can be chosen to be used for the matching and localization in the next step with the help of the current vehicle posture.

4. Results and Discussion

Several experiments were performed to validate the proposed AVP-SLAM system. All the presented data were taken from the vehicle platform. Four cameras mounted at the front, rear, left, and right sides of the vehicle with fish lens were used in this SLAM system. Furthermore, a 16-channel Lidar was used to help build the map. A Neousys computer with 32G RAM size and 11 G video memory was used for good system efficiency. The front image was taken by the front camera at 30 Hz with a resolution of 1920×1080 pixels. The images taken by the rear, left, and right cameras were recorded at 20 Hz with a resolution of 640×480 pixels. After image stitching optimization and synchronization, an image with a resolution of 1090×860 pixels was output by the system at 18 Hz.

4.1. Real Semantic Segmentation Experiment Result. The experiments were performed with several harsh external conditions to test the robustness performance of the proposed AVP-SLAM algorithm. There is no open-source dataset of the underground parking lot with semantic segmentation for bird's eye view. We have made an AVP dataset, and it is a single underground parking lot dataset for bird's eye view. Please refer to the link of supplementary materials for the data set. The performance of network ATT U-Net in the 520th epoch verification set is the highest, and the loss value is 0.38. The loss value decreases in the subsequent training process, but its performance in the verification set decreases, and the network is overfitted. Finally, the ATT U-Net parameters of the 520th epoch are used for the subsequent segmentation process. The IoU result is shown in Table 1. The final semantic segmentation result with ATT U-Net is shown in Figure 13. Although the parking line is blocked by the car, the parking line can still be clearly seen in Figure 13(a). Figure 13(b) has various marks in the same figure, but they are accurately identified. Although the reflected light overlaps with the white traffic signs, the traffic signs are segmented correctly, as shown in Figure 13(c). This case is the same as the speed bumps and traffic signs, which are shielded by other parked automobiles, as shown in Figures 13(d) and 13(e). Based on the result shown in Figure 13(f), the semantic segmentation result was not affected by the relatively dim light in the parking lot. The result of the U-Net segmentation mechanism on the underground parking lot is very good. In our system, the semantic is visualized and output, and the type of map is also a pixel map. The output of the system is 10fps-12fps, which can meet the real-time positioning requirements of low-speed vehicles. In summary, every specific feature was segmented precisely under different environments.

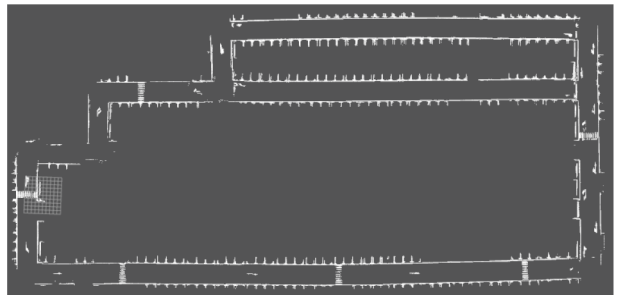


Figure 14: Global optimized semantic map.

4.2. Mapping and Localization. The experiment was performed in a dim underground parking lot. We used an additional Lidar in this system because the semantic map precision was easily affected by the initial values because of the relatively large error during matching between frames and the semantic map. Thus, we adopted Lidar, which is used for building maps. The SC-LeGO-LOAM framework was used to build the map, and the ROS/C++ was used as the code framework. Furthermore, the loop detection with scene context algorithm was used to optimize the mapping precision. The semantic map could be built with the posture data in the established Lidar map by matching the corresponding semantic image. Finally, the global optimized semantic map is shown in Figure 14.

Localization precision is more important than mapping precision because the automobile can localize and drive itself to the correct destination position even with an imprecise semantic map. In our experiment, the localization experiment was performed with the previously established and optimized semantic map. The initial position of the automobile is regarded as known data. These data are usually saved in NVM when the vehicle was parked the last time.

The final localization result is shown in Figure 15. The red line is used to show the motion trail for the experimental vehicle. During the experiment, the vehicle could constantly localize itself from start to end. The detailed localization result can be found in the video material.

4.3. Real Application: Autonomous Valet Parking. The proposed AVP-SLAM system was used under real autonomous valet parking cases in the underground parking lot. The preestablished semantic map was used by the vehicle to localize itself in this parking lot and guide itself to the prechosen parking lot automatically, as shown in Figure 16. Additional detailed experiment results can also be found in the shared video materials. In conclusion, a good SLAM result can be provided with the proposed AVP-SLAM system.

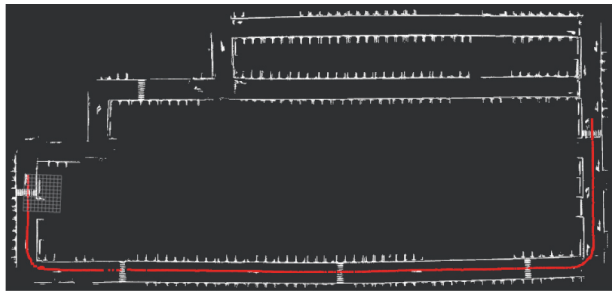


Figure 15: Experimental localization result in the parking lot.

Figure 16: Test vehicle.

5. Conclusions

In this study, a camera-Lidar combined with the SLAM solution was proposed. In this scheme, a 16-channel Lidar was used in assisting the visual system, that is, four surrounding-view cameras with fish lens, to build the map. Moreover, the semantic features, lane lines, parking lines, speed bumps, traffic signs, and other visual features could be detected using ATT U-Net even under harsh situations. Thus, a complete semantic map was built based on the detected features. With the preobtained map, the vehicle could localize itself during driving.

Furthermore, a real AVP experiment was performed to validate the proposed SLAM solution. The result showed that the vehicle can park itself in a correct parking lot autonomously in a dim underground parking lot. Thus far, the proposed SLAM solution is only effective with the AVP scenario. We will continue to develop this solution in the application field in a much more difficult environment.

Data Availability

The data are available through https://pan.baidu.com/s/1ioeXqYIlocYpsQb0KB-_Ng. The extraction code is lxq8.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

C. J. Liu completed the principle derivation, scheme design, scheme implementation, experimental verification, and article writing. J. J. Yao completed the system design and experimental scheme design.

Supplementary Materials

The data, video materials, and code are available through https://pan.baidu.com/s/1ioeXqYIlocYpsQb0KB-_Ng. The extraction code is lxq8. (*Supplementary Materials*)

References

[1] R. Smith, "On the representation of spatial uncertainty," *Int.j.robotics Res*, 1986.

[2] Y. Zhou, "A survey of VSLAM," *CAAI Transactions on Intelligent Systems*, 2018.

[3] J. Yue, *Lidar Data Enrichment Using Deep Learning Based on High-Resolution Image: An Approach to Achieve High-Performance Lidar-Based SLAM Using Low-Cost Lidar*, 2020.

[4] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020.

[5] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2014.

[7] T. Qin, P. Li, and S. Shen, "Vins-mono: a robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

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

[9] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *Proceedings of the 9th European conference on Computer Vision*Springer-erlag, Graz Austria, May 2006.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," computer vision - ECCV 2010," in *Proceedings of the 11th European Conference on Computer Vision*, Heraklion, Crete, Greece, September 2010.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, Barcelona, Spain, Novemer 2011.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] R. Mur-Ar tal and J. D. T Ar Dos, "ORB-SLAM2: an open-source SLAM system for monocular," *Stereo and RGB-D Cameras*, 2016.

[14] L. Yan, "Monocular localization in urban environments using road markings," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Redondo beach, CA, USA, June 2017.

[15] M. Schreiber, C. Knoppel, and U. Franke, "LaneLoc: lane marking based localization using highly accurate maps," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV)*, Gold Coast City, Australia, June 2013.

[16] A. Ranganathan, D. Ilstrup, and W. Tao, "Light-weight localization for vehicles using road markings," *Intelligent Robots and Systems (IROS)*, 2013.
[17] E. Rehder and A. Albrecht, "Submap-based SLAM for road markings," in *Proceedings of the Intelligent Vehicles Symposium*, Seoul, South Korea, 2015.
[18] J. Jeong, Y. Cho, and A. Kim, *Road-SLAM: Road Marking Based SLAM with Lane-Level Accuracy*, pp. 1736–1473, IEEE, New Jersy, NJ, USA, 2017.
[19] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," robotics and automation," in *Proceedings of the 2007 IEEE International Conference on IEEE*, Washington, DC, USA, August 2007.
[20] W. Weng and X. Zhu, "INet: convolutional networks for biomedical image segmentation," *IEEE Access*, 991, p.2021.
[21] M. Z. Alom, *Recurrent Residual Convolutional Neural Network Based on U-Net (R2U-Net) for Medical Image Segmentation*, 2018.
[22] Y. Ye, *Universal Semantic Segmentation for Fisheye Urban Driving Images*, 2020.
[23] H. Zhao, "Pyramid scene parsing network," *IEEE Computer Society*, 2016.
[24] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional Networks for Biomedical Image Segmentation*, Springer International Publishing, New York, NY, USA, 2015.
[25] S. Pasban, "Infant brain segmentation based on a combination of VGG-16 and U-Net deep neural networks," *IET Image Processing*, vol. 14, 2021.
[26] C. Szegedy, "Inception-v4," *Inception-ResNet and the Impact of Residual Connections on Learning*, 2016.
[27] E. Romera, "ERFNet: efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 1–10, 2017.
[28] Y. Hou and Z. C. T.-W. C. C. Ma, "Inter-region affinity distillation for road marking segmentation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, June 2020.
[29] O. Oktay, *Attention U-Net: Learning where to Look for the Pancreas*, 2018.
[30] Z. Ji and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *Proceedings of the Robotics: Science and Systems Conference*, Berkeley, CA, USA, July 2014.
[31] T. Shan and B. Englot, "LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IROS) IEEE, Madrid, Spain, October 2018.
[32] G. Kim and A. Kim, "Scan context: egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IROS) IEEE, Madrid, Spain, October 2018.
[33] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems IEEE*, pp. 163–169, Daejeon, Korea, October 2016.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2304097, 15 pages
https://doi.org/10.1155/2022/2304097

Research Article

Hyperpath Searching Algorithm considering Delay at Intersection and Its Application in CVIS for Vehicle Navigation

Ziyan Ju and Muqing Du

College of Civil and Transportation Engineering, Hohai University, Nanjing, Jiangsu 210098, China

Correspondence should be addressed to Muqing Du; dumqhhu@hhu.edu.cn

Received 23 August 2021; Revised 25 January 2022; Accepted 14 February 2022; Published 9 March 2022

Academic Editor: Peng Hang

The periodic change of intersection signals in urban road systems is the leading cause of uncertain delays. Therefore, aiming to minimize the travel time on the road segments and the expected delay at the intersections, a hyperpath search method based on intersection signal timing is proposed. The expanded network representation is used to capture the different turning delays at intersections. According to the intersection signal timing, the maximum waiting time of each turn and the turning movement ratio is obtained. A "recommendation priority passable phase" guidance strategy is proposed for the cooperative vehicle-infrastructure system (CVIS) based on the optimal hyperpath. The simulation experiments show that the vehicles can shift the driving route in the hyperpath set according to the guidance strategy, which effectively reduces the actual delay at intersections, and further realize an optimized distribution of traffic flows in the road network.

1. Introduction

The intelligent transportation system, along with the driverless technology and the cooperative vehicle-infrastructure system (CVIS), has played a pivotal role in improving efficiency and reducing travel delays on the road network [1]. The route search algorithm is the key technology that makes for vehicle route guidance and traffic distribution optimization. Expressed more formally, in connection with the need to realize reliable route guidance, the intelligent transportation system mostly uses the Dijkstra algorithm and its improved algorithms (such as the A* algorithm considered heuristic information) to solve the shortest path problem [2]. However, the simple-path algorithm cannot consider the several uncertain resources in the transportation networks such as traffic congestion, weather conditions, and vehicle accidents. In this way, traffic is easily guided to the same route, leading to overload along the way, then reducing the navigation system's practicability and credibility [3].

Lee [4] found that the test results indicated that the multiple path routing strategy performed better than the commonly used shortest path routing strategy using traffic simulation. Hence, shortest path methods have been extended to generate alternative routes, perhaps to avoid sites of congestion or specified areas.

Chen et al. [5] reviewed methods for generating multiple paths in the context of route guidance. Traditionally, alternati2ve paths could be calculated by two categories of the algorithm in graph theory, namely, (1) the k-shortest path (KSP) algorithms proposed by Eppstein [6], Martins [7], and Jimenez and Marzal [8] and (2) the totally disjoint path algorithms proposed by Dinic [9] and Torrieri [10].

However, these algorithms have some drawbacks for route guidance. For the k-shortest path algorithms, the searched k edges have considerable overlap, reducing choice diversity. For the disjoint path algorithms, the primary shortest path of the network may not be included, or the length of alternative paths may be unacceptable.

Intersections are the bottleneck in urban traffic networks, and the drivers' turning movement at signalized intersections can impact the traffic distribution on the road network. It may often be observed, apparently indeed in the great majority of experienced drivers, that they will change

lanes early to cross the intersection instead of waiting. Based on the above, all paths, which can reduce the drivers' waiting time at signalized intersections, are defined as the hyperpath set on the road networks. Our numerical results show that the hyperpath algorithm performs better in reducing delays, rather than both the shortest path algorithms and the multiple path algorithms.

At last, this article applies the hyperpath algorithm to vehicle navigation with the CVIS. Under the "recommend priority passable phase" rule, the cooperative vehicle-infrastructure system is introduced to improve the vehicle guidance system based on the hyperpath algorithm. Under the information interaction, the driver can change the lane in advance by voice, text, picture, and video so that the driver can shift their routes in the hyperpath subset. In fact, the traffic distribution will be optimized, reducing the total time and improving the network performance.

The outline of this article is given as follows: Section 2 reviews the literature on multiple path algorithms and discusses the shortcomings of the existing path planning algorithms. Then, we describe the origin of the hyperpath from the transit assignment. Section 3 develops the hyperpath algorithm incorporated with the signal timing and then gives the constraints and the proofs. In Section 4, the hyperpath algorithm is tested on randomly generated road networks to demonstrate the advantages of the proposed algorithm. Section 5 introduces the CVIs to realize the vehicle route guidance. Section 6 presents the conclusions thus far and suggestions for future work.

2. Literature Review

The k-shortest path (KSP) problems involve finding the shortest path, the second shortest path, and so on to the kth shortest path between a given origin and a destination (O–D) pair [11]. KSPs are usually provided in route guidance systems to satisfy various preferences that different users have for path choices [12]. Chen et al. introduced the A* technique to improve KSPs finding performance in stochastic networks under travel time uncertainty [13]. Some scholars proposed the improved KSPs that can find different shortest paths with a reasonable degree of similarity and close travel time, and the numerical result is satisfactory [14–16]. Recently, an efficient deviation path algorithm has been proposed for finding exactly the k-shortest simple paths without loops in road networks, which performs significantly better than the state-of-the-art algorithms [17]. A considerable amount of research has been devoted to developing KSP algorithms with better performance on the algorithm speed and ability to consider uncertainties. However, existing improved KSP algorithms do not change the nature, so that the k-shortest simple paths overlap each other and lack accessibility. The k-shortest simple paths have different optimal targets, which cannot be achieved when the drivers shift their routes.

The total disjoint path algorithm offers a major reduction in computation time for large networks. [10]. Here, "disjointness" can be considered in terms of either nodes or links [18]. A link-disjoint path-pair is a pair of paths between these nodes with no common links, but they still might share common nodes [19]. Accordingly, a node-disjoint path-pair is a pair of link-disjoint paths between these nodes with no common nodes [20]. Recently, scholars have taken into account the total disjoint path algorithm's vital feature to provide reliable service on the network, which guarantees to survive any single link failure [21]. As referred to earlier, the disjoint paths may not be shortest, but they are more suitable to uncertain traffic conditions.

Conventionally, path selection in the road network in the time-dependent vehicle routing problem is denoted as flexibility [22–24]. However, not all alternative paths have flexibility, but the ones with the same constraint conditions can have it. For instance, the k-shortest paths with different objective functions lack flexibility and are the same as the pair of the disjoint paths. The drivers cannot avoid the uncertain delay via shifting their routes among the results of multipaths. Consequently, there is an urgent need to realize path flexibility, which can reduce delays at intersections.

Indeed, empirical studies have shown that travel time variations significantly influence travelers' route-choice behavior [25]. Unlike previous algorithms, the driver can flexibly shift their routes to avoid delay in the hyperpath set. So the hyperpath algorithm has accessibility. Spiess and Florian [26] employed the hyperpath to solve passenger flow distribution on bus lines. In the public transportation model, the hyperpath that includes all the bus lines set between the origin and destination is interpreted as an optimal riding strategy. Bell [27] further considered the uncertainties of travel time on each link in the road network and applied Spiess and Florian's transit hyperpath to the road network. Bell pointed out that the hyperpath search algorithm was a special multipath algorithm that takes the minimum expected travel time as the goal. The hyperpath algorithm searches all paths that will be possible to become the shortest one between the origin and destination and incorporates them into the hyperpath subnet. Based on Bell's research, Ma et al. [28] proposed an improved method to speed up the hyperpath algorithm. However, the current hyperpath algorithms do not consider the impact of intersection delays on the expected travel time.

Moreover, we consider the delays at signalized intersections, which significantly contribute to the uncertainty of travel times, particularly in urban transportation networks [29–32]. In contrast, existing hyperpath studies about the road network have rarely addressed delays at signalized intersections due to the fact that signalized intersections can largely increase the complexity of the algorithm.

We aim to develop an efficient hyperpath searching algorithm typically applied to signalized road networks. The proposed algorithm conquers the inaccessibility of the most

existing multiple path algorithms (e.g., KSP). Table 1 summarizes the differences of our proposed algorithm from the relevant studies. The contributions of this study are presented as follows:

(i) We propose a hyperpath-based vehicle routing method to realize the accessibility among the multiple paths, reducing the delay at intersections, which is more efficient and performs better than the existing multiple path algorithms

(ii) We propose a hyperpath algorithm that considers the uncertain delay at signalized intersections in the road networks

(iii) We introduce the CVIS to realize vehicle navigation under the optimal hyperpath

3. Models

3.1. Hyperpath-Based Choice Strategy. The concept of the hyperpath, i.e., a set of paths any one of which may be optimal, comes from the field of transit assignment and is associated with the common-line issue. We assume that the passenger arrives at bus stop A, whose destination is bus stop D. Assume that three attractive bus lines given the headways arrive randomly in Figure 1. The passenger taking the bus line depends on which one happens to arrive next. The lines that are attractive at a given stop hence constitute the common lines, namely, hyperpath. Under the "take whichever attractive line arrives next" rule, the hyperpath may be found by minimizing the expected travel time and that the resulting problem is a linear program.

(i) We assume that there is only bus line 1 from stop A to stop D, with a headway of 6 minutes. Hence, the expected waiting time of passengers is $T = 6$ min.

(ii) We assume that three bus lines are attractive, which is referred to earlier. It will become evident that the expected waiting time of passengers is $T = 1/(1/6 + 1/3) = 2$ min.

Since the applicability of the transit assignment model is satisfying, some researchers have introduced this concept into the road network.

Bell noted the parallel between link frequency and link delay and then extended the Spiess and Florian algorithm by adding node potentials into the link selection step, yielding an algorithm that resembles the Astar algorithm, but which generates a hyperpath.

Consider a road network with eight links and seven nodes, shown in Figure 2. In Bell's paper [27], each link delay is equal to a random number R. For simplicity, the delay at the intersection is not considered. In this case, it can be seen in Figure 2, as all the links are unreliable, the hyperpath contains two paths, and whichever path of the set of hyperpath will become the shortest one.

Since the delay at signalized intersections is the main component of travel time, we significantly emphasize its influence on the performance of the hyperpath algorithm.

This study improves the original hyperpath search algorithm by expanding the signalized intersection. Aiming to minimize the travel time on the road segments and the expected delay at the intersections, we get a hyperpath set between the origin and destination.

We discuss the relationship of the turning movement, signal timing, and delay, as shown in Figure 3. Take one entrance lane as an example, where it consists of a left-turn movement and a through movement, corresponding to different signal phases. Assuming that the delay at the intersection is entirely controlled by the signal timing, the turning delay varies with the turning movement. In this way, this article refers to the signal timing delay at intersections to the waiting time in the transit network and then proposes the equation of the delay at intersections in the road network.

The original hyperpath algorithm only considers uncertain delays on the road segment, which is the delays of leaving their upstream nodes. The turning delay will depend on the turning movement that vehicle passes through the intersection. As shown in Figure 4(a), the delays caused by signal timing vary in different entrances and exits. This article expands the intersection with signal control in the road network so that the hyperpath algorithm overcomes the original shortcomings that cannot capture the turning delay at intersections. Each turn is described as a virtual link. Link weight is used to represent the turning delay at intersections. Take Figure 4 as an example, and there are four types of turns and four virtual links at one-way intersections; for the two-way, there are twelve types of turns and twelve virtual links.

This approach is to expand each intersection in the network. Thus, taking the one-way intersection as an example, the label rule is shown in Figure 5. The node j is expanded to four nodes. The travel time between nodes i and j is c_{ij}, and the travel time on the virtual link is the turning delay. d_{ijk} denotes the turning delay from road segment (i, j) to road segment (j, k).

3.2. Hyperpath Model. In the extended road network, the optimal hyperpath problem is described as a mathematical optimization model that takes the minimum travel time on the road segment and the expected delay at intersections as the objective function.

Define the following sets and variables:

$G(V, A)$: A directed graph

V: A set of vertices

A: A set of edges

I: Node

J: Downstream node of the node i

K: Downstream node of the node j

L: Set of links

H: Set of links of the hyperpath

$\Gamma^+(i)$: The set of edges leaving the node i

$\Gamma^-(i)$: The set of edges entering the node i

Table 1: The most relevant existing studies on hyperpath.

Literature	Path flexibility	Travel time reliability	Delays at intersections
Chen et al., 2016	×	✓	✓
Shen et al., 2020	×	✓	✓
Bell, 2008	✓	✓	×
Ma et al., 2013	✓	✓	×
This article	✓	✓	✓

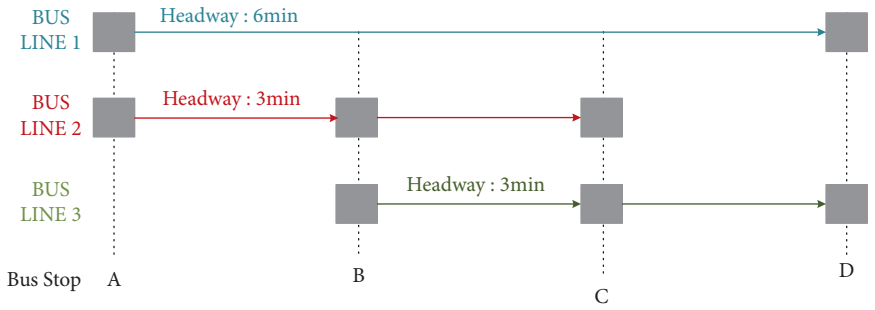


Figure 1: The hyperpath for the transit assignment model.

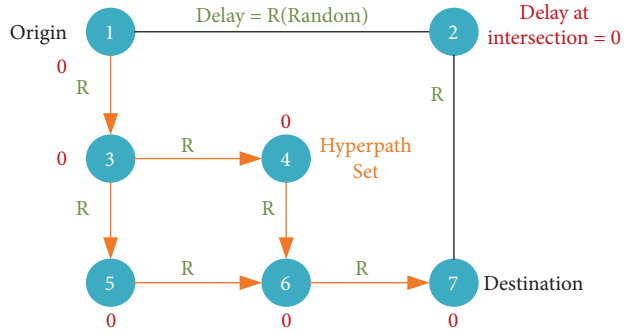


Figure 2: Hyperpath for unreliable links in the road network.

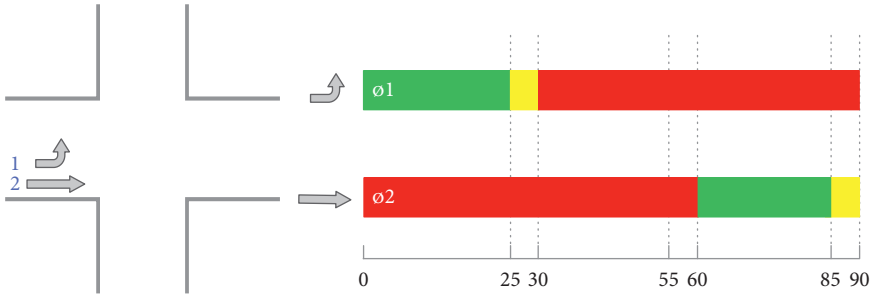


Figure 3: The turn delay varies with the turning movement.

u_i: Minimum travel time from node i to the destination

y_i: Probability of node i being selected

d_{ijk}: Turning delay from node i through node j to node k

c_{ij}: Travel time of on (i, j)

w_j: Expected delay time at node j

p_{ij}: Probability of (i, j) being selected

f_{ijk}: Service frequency of turning movement d_{ijk}, is $1/d_{ijk}$

f_j: Combined service frequency of node j

The hyperpath is identified by the following linear program:

$$\min_{p,w} \sum_{(i,j)\in A} c_{ij} p_{ij} + \sum_{i\in V} w_i, \tag{1}$$

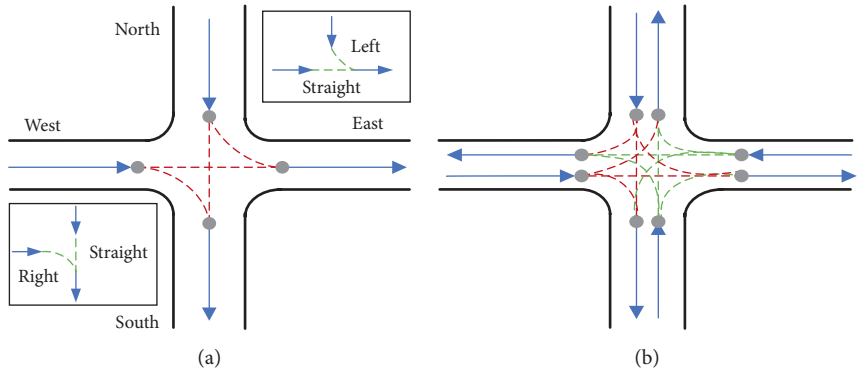


Figure 4: The expanded road at intersections. (a) The one-way intersections. (b) The two-way intersections.

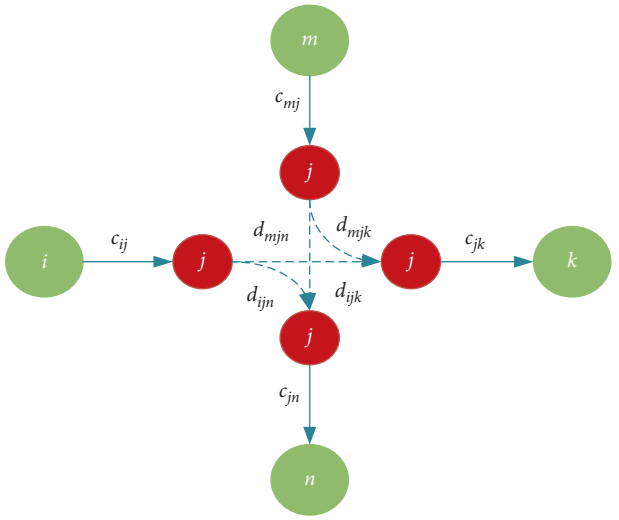


Figure 5: The label at the expanded network.

s.t.

$$\sum_{(j,k)\in A_j^+} p_{jk} - \sum_{(i,j)\in A_j^-} p_{ij} = g_j, \forall j \in V, \tag{2}$$

$$p_{ij} \in [0,1], \tag{3}$$

$$d_{ijk} \geq c_{jk} \cdot p_{jk}, \forall (j,k) \in A_j^+, i \in \Gamma^- (j). \tag{4}$$

In (1), the expected travel time is shown. Note that expected delay is interpreted as what would be expected by a pessimistic driver, namely, its exposure to maximum link delay. The travel time c_{ij} is a state of the free flow; i.e., the travel time does not change with increased traffic.

In (2), g_i is the origin and destination identification. When the node is the starting point r, $g_i = 1$. When the node is the endpoint s, $g_i = -1$; otherwise, $g_i = 0$. During the trip, there can be detours, but it is not allowed to arrive at a place twice. In (2), for the origin, its value is 1, which means the path starts from the origin. For the destination, the value is −1, which means that the trip stops at this point and ends the trip. Otherwise, its value is 0, which means that the path passes through the node with a certain probability. That is, it leaves after reaching the node.

In (3), the choice probability ranges from 0 to 1. The maximum probability of the link selected is 1. If not, the probability is 0.

Equation (4) indicates that the vehicle chooses to detour instead of waiting at the current intersection because the current turning delay is greater than the product of the travel time and the probability of being selected.

For the extended network, the virtual road segment corresponds to the turning movement at intersections, and the maximum delay d_{ijk} is caused by the periodic change of the signal light. This article assumes that the vehicle arrives randomly at the intersection and defines the intersection service frequency f_{r_j} ($f_{r_j} = 1/d_{r_j}$) of the turning r_j. In the hyperpath set, for node j, there may be multiple available turning r_j , and R_j (e.g., $R_j = \{d_{ijk}, d_{ijn}, d_{mjk}, d_{mjn}\}$) is the set of all available turns at this node. Further, we define the intersection service frequency $f_j = \sum_{r_j \in R_j} f_{r_j}$; then at the intersection, the expected waiting time w_j is as follows:

$$w_j = \alpha / \sum_{r_j \in R_j} f_{r_j}. \tag{5}$$

3.3. Optimal Hyperpath Algorithm

3.3.1. Initialization. Specify the origin r and the destination s; create a set L of links and add all links in the road network to the set L; create a set H of the hyperpath so that the set H is initially an empty set. Initialize the variables as follows:

$$\begin{aligned} &u_s = 0, u_i = \infty, \forall i \neq s, \\ &y_r = 1, y_i = 0, \forall i \neq r, \\ &\forall i \neq V, f_i = 0, d_{ijk} = 0, \\ &\forall (i,j) \in A, p_{ij} = 0, f_{ijk} = 0. \end{aligned} \tag{6}$$

3.3.2. Main Steps

Step 1: find the shortest link in the set L, take $(u_j + c_{ij})$ as the current link, and remove it from the set L.

Step 2: if the current link meets the conditions: $u_i \geq u_j + c_{ij}$, go to Step 3; otherwise, go back to Step 1.

Step 3: update the data.

If $i = r$, $f_{ijk} = 1$,

$$u_i = \frac{\beta + \left(u_j + c_{ij}\right) f_{ijk}}{f_i + f_{ijk}}, f_i = f_i + f_{ijk}. \quad (7)$$

If $u_i = \infty$ *or* $f_i = 0$,

$$\beta = 1; \quad (8)$$

otherwise,

$$\beta = u_i f_i. \quad (9)$$

Step 4: add the current link (i, j) to the set H. At this time, if the condition $L = \varnothing$ or $u_i + c_{ij} \geq u_r$ is satisfied, go to Step 5; otherwise, go back to Step 2.

Step 5: get the link and node choice probability: from largest to smallest, sort all links (i, j) in the network according to the value of $u_i + c_{ij}$ and traverse all the links according to the sorted order; if the link $(i, j) \in H$, access the database to obtain the probability of selected for each turn.

Starting from the origin r, trace back the hyperpath to the destination s and output it for route guidance strategy.

3.4. Algorithm Validation. The algorithm validation is as follows.

Proposition 1. *Assumption of maximum pessimistic expectation implies that $c_{jk} \cdot p_{jk} = d_{ijk} > 0$ if $p_{jk} > 0$ and $c_{jk} > 0$ for $(j,k) \in A_j^+$ and $i \in \Gamma^-(j)$.*

Proof 1. The Lagrangian function for equations (1)–(4)is as follows:

$$L_{p,w,\mu,\lambda} = \sum_{(i,j)\in A} c_{ij} p_{ij} + \sum_{i\in V} w_i + \sum_{i\in V} \mu_j \left(\sum_{(j,k)\in A_j^+} p_{jk} - \sum_{(i,j)\in A_j^-} p_{ij} - g_j \right) + \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \left(c_{jk} \cdot p_{jk} - d_{ijk} \right), \quad (10)$$

where μ and λ are the Lagrange multipliers concerning $\mu \geq 0$ and $\lambda \geq 0$. The optimal solution of the Lagrange function is as follows:

$$\frac{\partial L_{p,w,\mu,\lambda}}{\partial w_i} = 1 - \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \frac{\partial d_{ijk}}{\partial w_i} = 0. \quad (11)$$

According to (5), $\partial d_{ijk}/\partial w_i \leq 1$; then from (11), we can know that $1 - \sum_{i\in\Gamma^-(j)} \sum_{(j,k)\in A_j^+} \lambda_i \partial d_{ijk}/\partial w_i \leq 1 - \sum_{i\in\Gamma^-(j)} \lambda_i \Rightarrow \sum_{i\in\Gamma^-(j)} \lambda_i \leq 1$. At the solution, because w_i is equal to 0 or over. If λ_i equals 0, p_{jk} could be reduced to 0 at the solution, violating the assumption that $p_{jk} > 0$. So, by the complementary slackness conditions, $\lambda_i > 0$ implies that $c_{jk} \cdot p_{jk} = d_{ijk} > 0$ for $(j,k) \in A_j^+$ and $i \in \Gamma^-(j)$.

Proposition 2. *At the point at which (i, j) is selected, u_j has been reduced to its final value.*

Proof 2. See the Bell [27].

Proposition 3. *If, at the point at which (i, j) is selected, $c_{ij} + u_j > u_r$, the algorithm should terminate.*

Proof 3. See the Bell [27].

Proposition 4. *If $d_{ijk} = 0$ for all links $(j,k) \in A_j^+$ and all nodes $i \in \Gamma^-(j)$ then the Hyperstar algorithm finds only the path(s) with least undelayed travel time.*

Proof 4. See the Bell [27].

4. Illustrative Example

To demonstrate the Hyperstar algorithm, we use a grid-type transportation network, as shown in Figure 6, with node r as the origin and node s as the destination. All arcs can be traveled in both directions, leading to 62 directional links. Except for the nodes 5 and 16, the other nodes are controlled by signal timing. The network is expanded to virtual links referred to earlier coded in C# language. Figure 7 shows the signal timing at each intersection with signal control. Table 2 presents the travel time of links. Note that every link is two-way, but we just list one case when the travel time of two links, whose upstream and downstream nodes interchange, is the same. Table 3 lists the turning delays at intersections.

Based on the assumption of maximum pessimistic expectation, taking the T-shaped intersection two shown in Figure 6 as an example, the calculation of the turning delays will be described. When the forward intersection number of the path is origin r, and the backward intersection number is three, the vehicle needs to go through intersection two. According to the signal timing of Figure 7, going through the intersection, two corresponds to the first phase, and the green time of the first phase in one cycle time is 25 s, the yellow time is 5 s, and the red time is the 60 s. Since the vehicle stops during the yellow time and the red time, it is determined that the turning delay is 65 s. According to it, the turning delays at intersections in Figure 6 are derived in Table 3.

According to the algorithm flow, this algorithm is coded in C#. Each link is marked with its choice probability of the

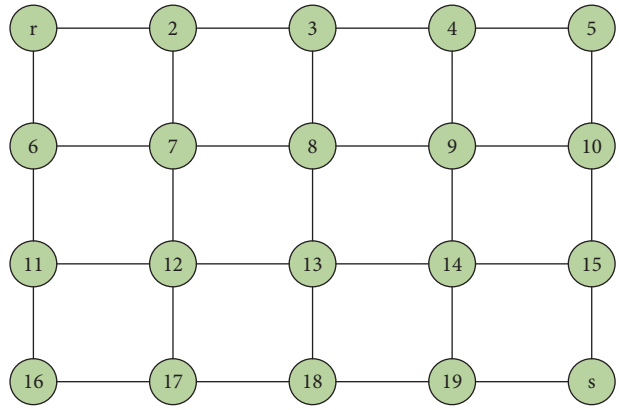


Figure 6: A grid-type network.

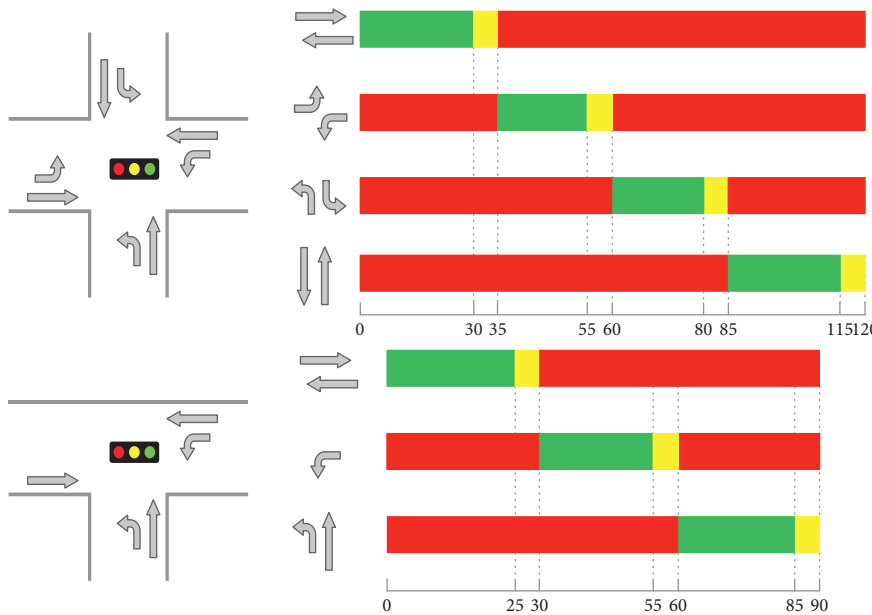


Figure 7: Intersection signal timing.

Table 2: Travel time of links.

Upstream node i	Downstream node j	Travel time (s)	Upstream node i	Downstream node j	Travel time (s)
R	2	60	19	S	40
2	3	60	r	6	30
3	4	40	6	11	20
4	5	30	11	16	40
6	7	6	2	7	60
7	8	70	7	12	30
8	9	30	12	17	60
9	10	20	3	8	20
11	12	50	8	13	30
12	13	70	4	9	20
13	14	60	9	14	40
13	18	30	14	19	30
14	15	30	5	10	50
16	17	70	10	15	60
17	18	60	15	S	70
18	19	30			

Table 3: Turning delays at intersections.

Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)	Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)
R	2	3	65	6	11	12	65
R	2	7	0	6	11	16	65
3	2	*r*	65	12	11	6	0
3	2	7	65	16	11	6	65
7	2	3	0	12	11	16	65
7	2	*r*	65	16	11	12	0
2	3	4	65	11	12	13	90
2	3	8	0	11	12	17	0
8	3	2	65	11	12	7	100
8	3	4	0	7	12	13	100
4	3	2	65	7	12	17	90
4	3	8	65	7	12	11	0
3	4	5	65	13	12	11	90
3	4	9	0	13	12	7	0
9	4	3	65	13	12	17	100
9	4	5	0	17	12	7	90
5	4	3	65	17	12	11	100
5	4	9	65	17	12	13	0
4	5	10	0	12	13	14	90
10	5	4	0	12	13	18	0
R	6	7	65	12	13	8	100
R	6	11	65	8	13	14	100
7	6	*r*	0	8	13	18	90
11	6	*r*	65	8	13	12	0
7	6	11	65	14	13	12	90
11	6	7	0	14	13	8	0
6	7	8	90	14	13	18	100
6	7	12	0	18	13	8	90
6	7	2	100	18	13	12	100
2	7	8	100	18	13	14	0
2	7	12	90	13	14	15	90
2	7	6	0	13	14	19	0
8	7	6	90	13	14	9	100
8	7	2	0	9	14	15	100
8	7	12	100	9	14	19	90
12	7	2	90	9	14	13	0
12	7	6	100	15	14	13	90
12	7	8	0	15	14	9	0
3	8	13	90	15	14	19	100
3	8	9	100	19	14	9	90
3	8	7	0	19	14	13	100
9	8	7	90	19	14	15	0
9	8	3	0	14	15	*s*	0
9	8	13	100	14	15	10	65
7	8	9	90	10	15	*s*	65
7	8	13	0	10	15	10	0
7	8	3	100	*s*	15	10	65
13	8	3	90	*s*	15	14	65
13	8	7	100	11	16	17	0
13	8	9	0	17	16	11	0
4	9	14	90	16	17	18	65
4	9	10	100	16	17	12	65
4	9	8	0	12	17	16	0
10	9	8	90	12	17	18	65
10	9	4	0	18	17	16	65
10	9	14	100	18	17	12	0
8	9	10	90	17	18	19	65
8	9	14	0	17	18	13	65

Table 3: Continued.

Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)	Forward intersection number	Intersection number	Backward intersection number	Turning delays (s)
8	9	4	100	13	18	17	0
14	9	4	90	13	18	19	65
14	9	8	100	19	18	17	65
14	9	10	0	19	18	13	0
9	10	15	0	18	19	*s*	65
9	10	5	65	18	19	14	65
5	10	15	65	14	19	18	0
5	10	9	0	14	19	*s*	65
15	10	5	65	*s*	19	18	65
15	10	9	65	*s*	19	14	0

driver with the algorithm output. The probability is also the traffic distribution, as shown in Figure 8.

According to the link probability, there are nine possible shortest paths. The path choice probability is shown in Table 4.

5. The Cooperative Vehicle-Infrastructure System

5.1. Work Process of CVIs. Based on the hyperpath, this section introduces the CVIs to realize route guidance. The CVIs consist of four parts: intelligent traffic management system, intelligent communication system, intelligent vehicle system, and intelligent roadside system. The intelligent traffic management system processes the information obtained by both the intelligent vehicle system and the intelligent roadside system and then calculates the turning delay included in the hyperpath set, which is specifically divided into the following two situations:

Case 1. According to the transmitted information, when the turn encounters the green light, the intelligent traffic management system obtains the remaining green light time t_1 and the vehicle's distance l from the current position to the turn-stop line. After that, the intelligent traffic management system calculates the number of vehicles Q_2 that can pass during the remaining green time. If $Q_2 \geq Q_1$, the time when the vehicle arrives at the stop line is $t = l/v_1$, if $Q_2 < Q_1$. The signal cycle time is T, which is obtained by the database. The number of vehicles is Q_3 that can be passed during one signal cycle. The dissipation speed of the vehicles is v_2. The number of vehicles that need to wait is $n = \lfloor (Q_1 - Q_2)/Q_3 \rfloor$ during a signal cycle. The time to arrive at the stop line is $t = t_1 + nT + (Q_1 - Q_2 - nQ_3)/v_2$. The number of vehicles in front of the induced vehicle is Q_1, the symbol $\lfloor \rfloor$ indicates rounding down and $Q_2 = q \cdot t_1$.

Case 2. When the turn encounters the red light, the intelligent traffic management system obtains the remaining red time t_3 from the current phase. If the turn encounters the yellow light, the intelligent traffic management system should obtain the remaining yellow time plus red time. According to the database, the signal cycle time is T. The number of vehicles that can pass during one signal cycle is Q_3. The dissipation speed of the vehicles is v_2. The number of the vehicles that need to wait is $n = \lfloor Q_1/Q_3 \rfloor$. The time to arrive at the stop line is $t = t_3 + nT + (Q_1 - nQ_3)/v_2$.

The intelligent traffic management system determines the turning movement with the shortest time and recommends it to the driver. That ensures that vehicles always choose the passable priority phase at the intersection to

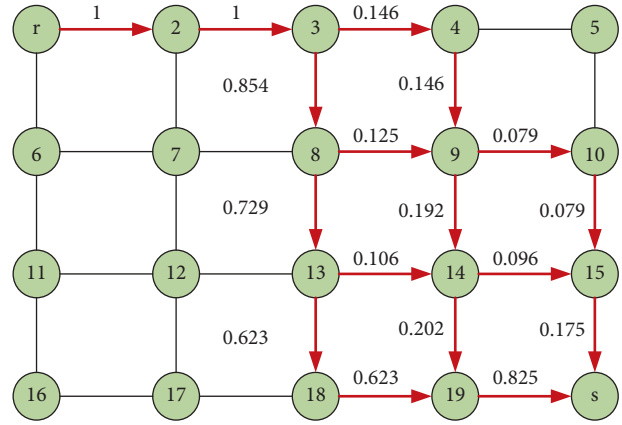


Figure 8: Link choice probability.

Table 4: Path choice probability.

Route	Choice probability
r-2-3-4-9-14-19-s	0.050
r-2-3-4-9-14-15-s	0.036
r-2-3-4-9-10-15-s	0.061
r-2-3-8-13-18-19-s	0.290
r-2-3-8-13-14-19-s	0.177
r-2-3-8-13-14-15-s	0.030
r-2-3-8-9-14-19-s	0.177
r-2-3-8-9-14-15-s	0.127
r-2-3-8-9-10-15-s	0.052

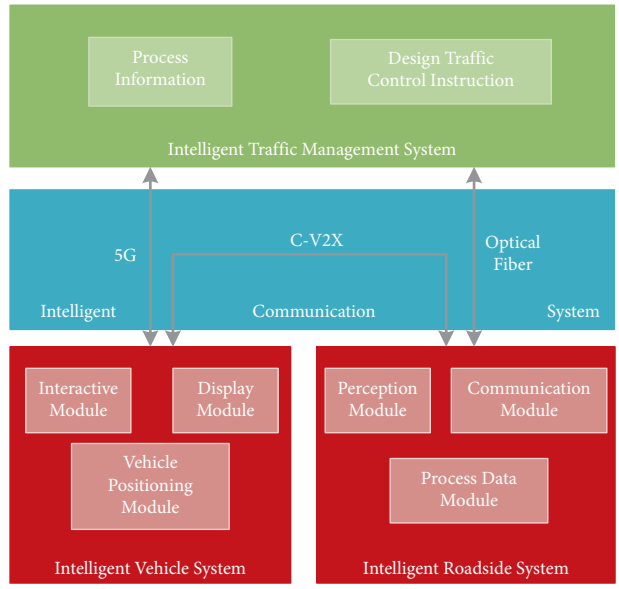


Figure 9: The cooperative vehicle-infrastructure system.

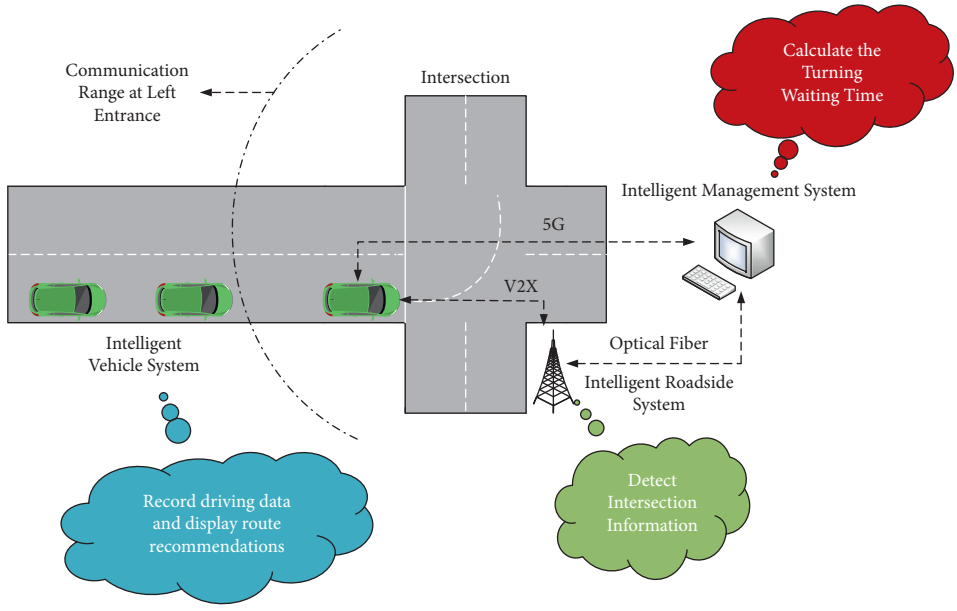


Figure 10: The information interaction on the cooperative vehicle-infrastructure system.

minimize waiting delays. Figure 9 shows the CVIs, and Figure 10 represents the information interaction principle of the vehicle route guidance system under the CVIs.

5.2. Numerical Experiment. Suppose a total of 1,000 drivers travel from the start point r to the end point s in a certain period. Travel along the nine paths was selected by the

Table 5: Hyperpath and single-path delay time comparison.

Hyperpath guidance under independence assumption						
Label	Route	Number of vehicles allocated	Expected vehicle delay time (s/pcu)	Total delay time (s/pcu)	Expected vehicle travel time (s/pcu)	Total travel time (s/pcu)
(1)	*r*-2-3-4-9-14-19-*s*	50				
(2)	*r*-2-3-4-9-14-15-*s*	36				
(3)	*r*-2-3-4-9-10-15-*s*	61				
(4)	*r*-2-3-8-13-18-19-*s*	290				
(5)	*r*-2-3-8-13-14-19-*s*	177	193.144	193,144	476.534	476,534
(6)	*r*-2-3-8-13-14-15-*s*	30				
(7)	*r*-2-3-8-9-14-19-*s*	177				
(8)	*r*-2-3-8-9-14-15-*s*	127				
(9)	*r*-2-3-8-9-10-15-*s*	52				
Shortest path guidance						
Label	Route	Number of vehicles allocated	Simple vehicle delay time (s/pcu)	Total delay time (s/pcu)	Simple vehicle travel time (s/pcu)	Total travel time (s/pcu)
(9)	*r*-2-3-8-9-10-15-*s*	1,000	205	205,000	525	525,000
Delays reduced by 5.8% Travel times reduced by 9.2%						

hyperpath algorithm (shown in Figure 9) and the only path with the shortest time. Calculate the delays of vehicles at the intersection, as shown in Table 5.

The results in Table 5 show that compared with the shortest path, applying the hyperpath guidance system in the road network can reduce the delays by 5.8% and travel times by 9.2%. It can be seen that the adoption of the hyperpath guidance strategy can realize the reasonable distribution of traffic in the network, reduce the delay at intersections, and achieve the shortest travel time in the network.

6. The Hyperpath in Nanjing Local Road Network

In order to validate our solution approach in a realistic context, we built our hyperpath network based on the local road network obtained from the center of Nanjing. By taking an investigation, there are 49 intersections and 37 out of them are controlled by traffic signals. Also, 29 road segments are involved, and some of them are one-way roads. Three road types (i.e., the arterial street, subarterial road, and branch road) are associated with different design speeds. The network is illustrated in Figure 11.

According to the output of the algorithm in Section 3.3, the optimal hyperpath is marked in Figure 12. Note that node 37 is the origin, and node 1 is the destination.

If we employ the Dijkstra algorithm on this network, the shortest path is 37-32-26-25-47-43-18-17-14-10-9-7-38-1, and the shortest travel time is 1,142 s. However, from the optimal hyperpath algorithm, the expected travel time is 1,093.88 s. Compared with the shortest path, the expected travel time of the optimal hyperpath is reduced by 4.2%. This result clearly showed the application of the hyperpath algorithm compared to the Dijkstra algorithm in terms of delays and travel time.

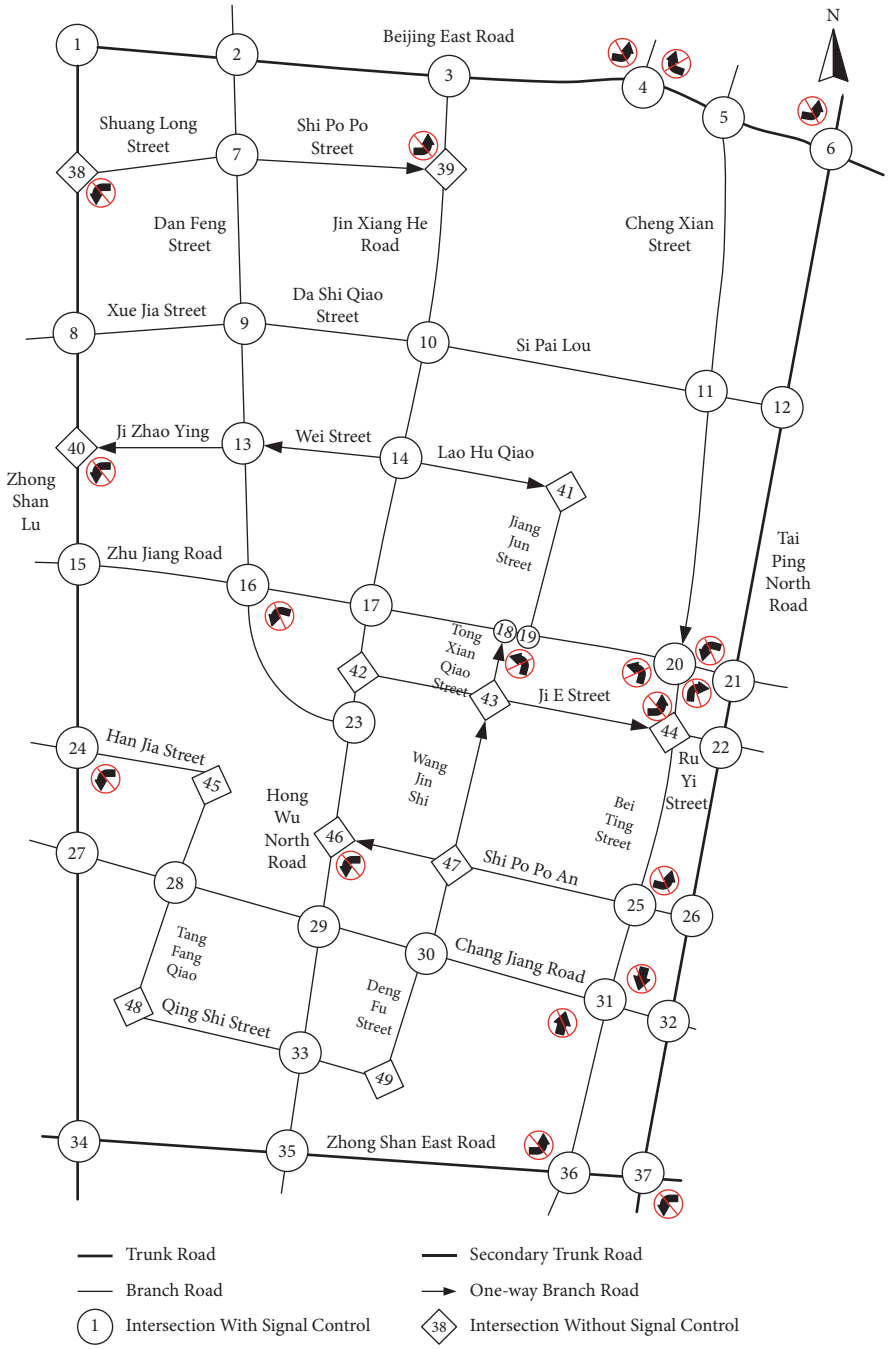


Figure 11: Local road network in Nanjing.

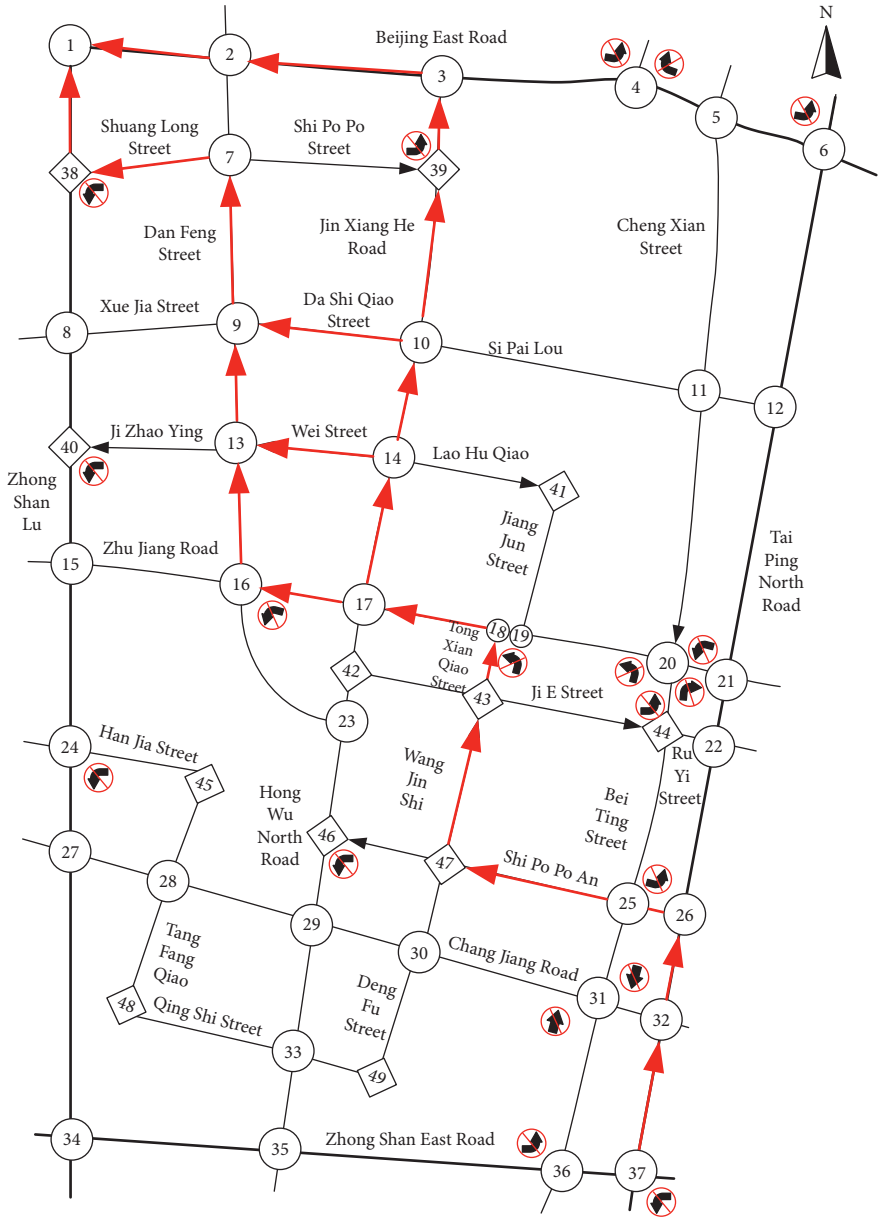


Figure 12: The optimal hyperpath from origin node 37 to destination node 1.

7. Conclusions

This article offers an efficient way to generate the optimal hyperpath set based on the signal timing and the CVIs for use in vehicle route guidance systems. Under the maximum pessimistic expectation assumption, this article generates the maximum waiting time for each turn at the intersection. Further, we find out the expected waiting time for linear combination, which avoids processing a large amount of data and accelerates the algorithm speed in the route guidance system. There is a "recommendation priority passable phase" strategy for the cooperative vehicle-infrastructure system, leading to a real-time and exact vehicle route guidance system.

For individuals, the driver can shift the driving route in the hyperpath subnet for the use of the route guidance system to avoid delays at intersections. As far as the overall network, this can be proven that using a hyperpath algorithm in the network will give a reasonable traffic distribution. Thus, the total travel time is reduced, and the transportation network's performance is improved, which will produce substantial economic benefits. In order to meet the increased demand for personalized travel, the hyperpath guidance system can introduce corresponding demand factors to change the objective function to improve network performance and further achieve personalized guidance to drivers. Meanwhile, it is vital to take into account the cooperative vehicle-infrastructure system when vehicles are guided. The tests concluded that compared with the shortest path, applying the hyperpath guidance system in the road network can reduce the delays by 5.8% and travel times by 9.2%. Such properties make the hyperpath algorithm a

promising solution for use in intelligent transportation systems, which was confirmed by an experimental study comparing the travel time and delay time with the shortest path algorithms.

There are two assumptions about calculating the waiting time of vehicles at intersections based on signal timing. One is the assumption of maximum pessimistic expectations. The waiting time for each turn of the vehicle at the intersection is red time plus yellow time. The other is the independent hypothesis, assuming that the vehicle's phases are independent of each other when calculating the expected waiting time at the intersection. These two assumptions are ideal, leading to a specific deviation from the actual situation. Future research will further improve the mentioned assumptions to make the hyperpath algorithm more suitable for actual road network conditions.

Data Availability

The network data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was funded by the Natural Science Foundation of China, Grant no. 71801079, and the Fundamental Research Funds for the Central Universities, Grant no. B200202079.

References

[1] M. P. Dixon and L. R. Rilett, "Population origin-destination estimation using automatic vehicle identification and volume data," *Journal of Transportation Engineering*, vol. 131, no. 2, pp. 75–82, 2005.

[2] S. U. Shariff and M. Ganeshan, "A path finding visualization using A star algorithm and dijkstra's algorithm," *Journal of Trend in Scientific Research and Development*, vol. 5, no. 1, pp. 588-589, 2020.

[3] D. Mansoor and G. Mohsen, ""Shortest path problem on uncertain networks: an efficient two phases approach," *Computers & Industrial Engineering*, vol. 157, 2021.

[4] L. Chi-Kang, "A multiple-path routing strategy for vehicle route guidance systems," *Pergamon*, vol. 2, no. 3, pp. 185–195, 1994.

[5] Y. Chen, M. G. H. Bell, and K. Bogenberger, "Reliable pretrip multipath planning and dynamic adaptation for a centralized road navigation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 14–20, 2007.

[6] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1998.

[7] E. De Queirós Vieira Martins, M. M. B. Pascoal, and J. L. E. D. Santos, "Deviation algorithms for ranking shortest paths," *International Journal of Foundations of Computer Science*, vol. 10, no. 03, pp. 247–261, 1999.

[8] V. M. Jiménez and A. Marzal, "Computing the K Shortest paths: a new algorithm and an experimental comparison," *Algorithm Engineering*, vol. 1688, pp. 15–29, 1999.

[9] E. A. Dinic, "Algorithm for solution of a problem of maximum flow in a network with power estimation," *Soviet mathematics - Doklady*, vol. 11, pp. 248–264, 1970.

[10] D. Torrieri, "Algorithms for finding an optimal set of short disjoint paths in a communication network," *IEEE Transactions on Communications*, vol. 40, no. 11, pp. 1698–1702, 1992.

[11] H. Yu, Z. Fang, F. Lu et al., "Impact of oil price fluctuations on tanker maritime network structure and traffic flow changes," *Applied Energy*, vol. 237, pp. 390–403, 2019.

[12] S. Vanhove and V. Fack, "An effective heuristic for computing many shortest path alternatives in road networks," *International Journal of Geographical Information Science*, vol. 26, no. 6, pp. 1031–1050, 2012.

[13] B. Y. Chen, Q. Li, and W. H. K. Lam, "Finding the k reliable shortest paths under travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 94, pp. 189–203, 2016.

[14] X. Hu and Y.-C. Chiu, "A constrained time-dependent K shortest paths algorithm addressing overlap and travel time deviation," *International Journal of Transportation Science and Technology*, vol. 4, no. 4, pp. 371–394, 2015.

[15] W. Xu, S. He, R. Song, and S. S. Chaudhry, "Finding the K shortest paths in a schedule-based transit network," *Computers & Operations Research*, vol. 39, no. 8, pp. 1812–1826, 2012.

[16] M. Niksirat, M. Ghatee, and S. Mehdi Hashemi, "Multimodal K-shortest viable path problem in Tehran public transportation network and its solution applying ant colony and simulated annealing algorithms," *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5709–5726, 2012.

[17] B. Y. Chen, X.-W. Chen, H.-P. Chen, and W. H. K. Lam, "Efficient algorithm for finding k shortest paths based on re-optimization technique," *Transportation Research Part E: Logistics and Transportation Review*, vol. 133, Article ID 101819, 2020.

[18] J. Yallouz, O. Rottenstreich, P. Babarczi, A. Mendelson, and A. Orda, "Minimum-weight link-disjoint Node-"Somewhat disjoint" paths," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1110–1122, 2018.

[19] T. Gomes and J. Craveirinha, "Efficient calculation of the most reliable pair of link disjoint paths in telecommunication networks," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1055–1064, 2007.

[20] T. Gomes, L. Martins, S. Ferreira, M. Pascoal, and D. Tipper, "Algorithms for determining a node-disjoint path pair visiting specified nodes," *Optical Switching and Networking*, vol. 23, pp. 189–204, 2017.

[21] C.-N. Lai, "Optimal node-disjoint paths in folded hypercubes," *Journal of Parallel and Distributed Computing*, vol. 147, pp. 100–107, 2021.

[22] L. Wang, S. Gao, K. Wang, T. Li, L. Li, and Z. Chen, "Time-dependent electric vehicle routing problem with time windows and path flexibility," *Journal of Advanced Transportation*, vol. 2020, pp. 1–19, 2020.

[23] R. Guo, W. Zhang, W. Guan, and B. Ran, "Time-dependent urban customized bus routing with path flexibility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2381–2390, 2021.

[24] X. Liu, M. Qi, and C. Cheng, "Green vehicle routing problem with path flexibility," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 10-13 Dec. 2017.

[25] L. Shen, H. Shao, T. Wu, E. Z. Fainman, and W. H. K. Lam, "Finding the reliable shortest path with correlated link travel times in signalized traffic networks under uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 144, Article ID 102159, 2020.

[26] H. Spiess and M. Florian, "Optimal strategies: a new assignment model for transit networks," *Transportation Research Part B: Methodological*, vol. 23, no. 2, pp. 83–102, 1989.

[27] M. G. H. Bell, "Hyperstar: a multi-path Astar algorithm for risk averse vehicle navigation," *Transportation Research Part B*, vol. 43, no. 1, pp. 97–107, 2008.

[28] J. Ma, D. Fukuda, and J.-D. Schmöcker, "Faster hyperpath generating algorithms for vehicle navigation," *Transportmetrica: Transportation Science*, vol. 9, no. 10, pp. 925–948, 2013.

[29] L. Hu, J. Ou, J. Huang et al., "Safety evaluation of pedestrian-vehicle interaction at signalized intersections in Changsha, China," *Journal of Transportation Safety & Security*, p. 26, 2021.

[30] L. Hu, X. Q. Zhou, X. Zhang, F. Wang, Q. Li, and W. Wu, "A review on key challenges in intelligent vehicles: safety and driver-oriented features," *IET Intelligent Transport Systems*, vol. 15, no. 9, pp. 1093–1105, 2021.

[31] L. Hu, X. Bao, M. Lin, C. Yu, and F. Wang, "Research on risky driving behavior evaluation model based on CIDAS real data," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 8, pp. 2176–2187, 2021.

[32] F. Zheng and H. Van Zuylen, "Uncertainty and predictability of urban link travel time," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2192, no. 1, pp. 136–146, 2010.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 6653598, 14 pages
https://doi.org/10.1155/2022/6653598

Research Article

A Multiobjective Cooperative Driving Framework Based on Evolutionary Algorithm and Multitask Learning

Xia Jiang,[1] **Jian Zhang**,[1,2] **Qing-yang Li**,[1] **and Tian-yi Chen**[3]

[1]*Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies and Jiangsu Province Collaborative Innovation Center for Technology and Application of Internet of Things, School of Transportation, Southeast University, Nanjing 210096, China*
[2]*School of Engineering, Tibet University, Lhasa, Tibet 850000, China*
[3]*Department of Civil & Environmental Engineering, University of Wisconsin-Madison, 1221 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706, USA*

Correspondence should be addressed to Jian Zhang; jiangzhang@seu.edu.cn

Received 23 July 2021; Revised 18 December 2021; Accepted 30 December 2021; Published 21 January 2022

Academic Editor: Peng Hang

The development of connected and automated vehicle (CAV) techniques brings an upcoming revolution to traffic management. The control of CAVs in potential conflict areas such as on-ramps and intersections will be complex to traffic management when considering their deployment. There is still a lack of a general framework for dispatching CAVs in these bottlenecks, which is expected to ensure safety, traffic efficiency, and energy consumption in real time. This study aimed to fill the technique gap, and a comprehensive cooperative intelligent driving framework is put forward to study the problem, which can be used in both on-ramp and intersection scenarios. Based on a multi-objective evolutionary algorithm, CAVs are denoted as a sequence to be searched in solution space, while a multitask learning neural network with adaptive loss function is implemented for optimization target feedback to surrogate the simulation test procedure. The simulation results show that the proposed framework can get satisfying performance with low time and energy consumption. It can reduce time consumption by up to 16.51% for the on-ramp scenario and 9.8% for the intersection scenario, while reducing energy consumption by up to 16.39% and 11.39% for the two scenarios. Meanwhile, an analysis of computation time is carried out, illuminating the flexibility and controllability of the new strategy.

1. Introduction

Connected and automated vehicles are considered to play an important role in improving traffic efficiency and saving energy [1]. The fickle driving behaviors can easily lead to a series of problems, including traffic congestion, energy consumption, and accident [2–4], but transport systems consisting of intelligent vehicles can make a difference using vehicle-to-everything (V2X) communication and advanced control techniques [5–7].

The development of connected and automated vehicles (CAVs) brings both opportunities and challenges to traffic management. As the bottlenecks in traffic organization, intersection and on-ramp become the research hot spots in the domain [8–10]. Conventionally, the vehicles must adhere to the traffic signals in urban scenarios, and corresponding studies are proposed to optimize the trajectory of vehicles in this case [1, 11, 12]. Considering the traffic environment composed of CAVs, traffic signals can be eliminated because the information on the road can be fully obtained [13], while the vehicles on the road can be fully controlled. It is possible to implement cooperative control for CAVs through V2X communication [14]. Thus, the design of a cooperative driving strategy through the use of real-time traffic information becomes particularly important. Ann and Colombo [15] pointed out that an effective cooperative driving framework can work in different traffic scenarios such as intersections, merging roadways, and roundabouts. On account of the significance of cooperative driving, the researchers proposed many theoretical

methods to solve the problem for different scenarios. Grand cooperative driving challenges were also organized to promote its development in practice [16].

Some of the existing studies belong to the optimization-based method. Yan et al. [17] proposed a dynamic programming algorithm to evacuate vehicles at the intersection as soon as possible. Zhu and Ukkusuri [18] put forward a linear programming model to dispatch vehicles at autonomous intersections in order to minimize total travel time. Besides, mixed-integer linear programming (MILP) is widely used to obtain solutions [19–21]. However, Li and Wang [13] proposed a framework based on the optimization principle, which utilized a tree search algorithm to achieve the same purpose. All of the listed studies focus on searching for optimal solutions based on different prior hypotheses.

Relevant studies pointed out that the key to solving the problem is determining the right-of-way for CAVs approaching the merging area [22–24]. In other words, the vehicles can be formulated as a passing sequence in the form of arrays, and the performance of the schedule strategy hinges on the way to generate the best passing order among a large number of possible solutions.

In terms of generating passing orders, the existing studies can be classified into two categories. One is the rule-based strategy, which uses some heuristic rules to determine the passing order of vehicles. Dresner and Stone [25, 26] proposed a reservation-based system and assigned right-of-way to vehicles on a first-come-first-served (FCFS) basis. Although the effectiveness of the FCFS method can be proved [26, 27], its rule-based nature always leads to feasible but not optimal solutions. Moreover, the reservation-based strategy cannot outperform traditional signal control in some cases [28, 29]. While the rule-based strategies cannot always perform very well, the other approach to generate passing orders is introduced, called "planning-based strategy" [13]. Meng et al. demonstrated that the planning-based strategy could consistently outperform the FCFS method in intersection scenarios by comprehensive simulations [30]. Actually, the planning-based strategy is a framework that can search for optimal solutions in a huge solution space. The strategy is essentially a traversal problem with intolerable computational complexity. Therefore, consequent studies focused on the reduction in computing time. Xu et al. [31] proposed a grouping-based strategy, which groups CAVs to reduce the count of possible solutions. In their other study, a Monte Carlo tree is built to keep the trade-off between coordination performance and computation time [32]. Meanwhile, Zhang et al. [33] reported a framework that utilized a neural network to surrogate the simulation test process with the intent to reduce computation time. However, the only optimization target they considered is about traffic efficiency indexes such as passing time or total delay, while the value of other targets such as energy consumption or queue length is difficult to acquire. This is caused by the weakness of the trajectory interpretation algorithm in their studies.

Therefore, there is still a lack of a real-time, multi-objective cooperative driving strategy that can be maneuverable and reliable. To this end, we design a multi-objective discrete evolutionary algorithm (MODEA) to search for (near) optimal passing orders, which combines the non-dominated sorting method [34] and state transition algorithm [35]. A multitask learning model is proposed to be a regressor, which can give feedback of objective values to MODEA. The scenario is simulated by Simulation of Urban MObility (SUMO) [36]. The simulation results indicate that the framework can be applied to different scenarios, performing well even under a high concurrency environment.

The rest of the study is arranged as follows: Section "Problem Statement" gives the general form of cooperative driving problems and traffic scenarios the paper studied. Section "Methodology" presents the framework we proposed, including the MODEA and multitask learning method in detail. Section "Simulation and Analysis" provides the simulation results of a series of experiments. Finally, conclusions are given in Section "Conclusion."

2. Problem Statement

Highway on-ramps and urban unsignalized intersections are two typical scenarios for cooperative driving (see Figure 1). Rios-Torres and Malikopoulos [37] pointed out that the two mainstream frameworks in the cooperative driving field are centralized coordination and decentralized coordination, respectively, while the method proposed in this study belongs to the former. The centralized frameworks rely on a central controller responsible for computing and sending control commands. The controller has a communication range (CR) that defines the boundary of communication and control. This article denotes the CR as a circle, which is widely adopted in previous studies [30, 33, 38]. Only vehicles within the CR will communicate with the controller and be controlled.

Some followed assumptions are listed to make the analysis and implementation easier:

(i) Lane-changing behaviors are prohibited in CR for safety consideration.

(ii) The system has no interference from pedestrians and non-motor vehicles.

(iii) All CAVs can transmit id, position, speed, and other precise information to the controller spontaneously.

(iv) The vehicles are homogeneous pure electric CAVs for estimating energy consumption. The energy model can be found in [39].

The general form of the objective function in cooperative driving can be defined as follows:

$$\min_{x} F(x), \tag{1}$$

where F is the function that represents queue length, energy consumption, or traffic delay, and x is the independent variable that will give rise to the optimization target. In this study, two objects are considered: (a) the minimization of time consumption to evacuate all CAVs in CR and (b) electricity consumption for CAVs in the process of a scheduling scheme.

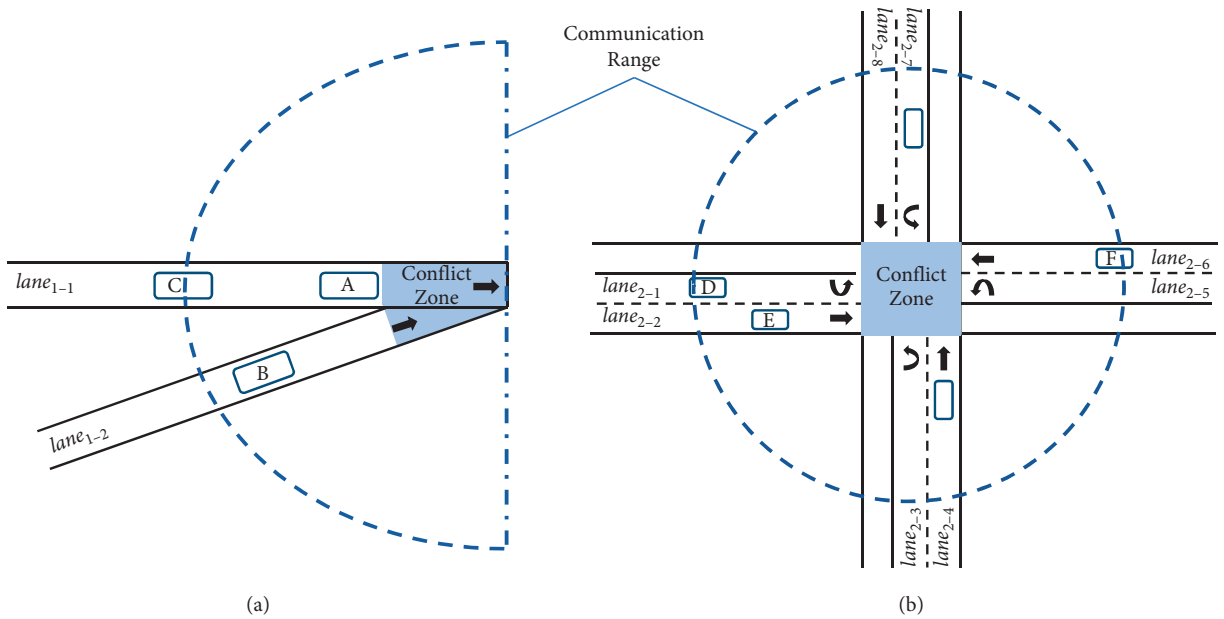


FIGURE 1: Two typical cooperative driving scenarios in general road networks. (a) The on-ramp in highway and (b) the unsignalized intersection in urban road.

The input of the function x denotes a passing sequence, which can be denoted as follows:

$$x = [\mathrm{CAV}_1, \mathrm{CAV}_2, \ldots, \mathrm{CAV}_n], \tag{2}$$

where n is the number of vehicles in CR. Let f_1 be the time consumption to evacuate all CAVs in CR, and f_2 be the corresponding electricity consumption, and (1) can be transformed to as follows:

$$\min_x [f_1(x), f_2(x)], \tag{3}$$

Here, $f_1(x)$ can be denoted as follows:

$$f_1(x) = t_e(\mathrm{CAV}_n), \tag{4}$$

where t_e represents the time when the vehicle CAV_n exits from CR. $f_2(x)$ can be denoted as follows:

$$f_2(x) = \sum_{i=1}^{n} \int_0^{t_e(\mathrm{CAV}_i)} e(\mathrm{CAV}_i), \quad i \in (1, n), \tag{5}$$

where e represents the energy consumption of CAV_i in discrete time, and readers can refer to [39] for the stepwise energy consumption model.

3. Methodology

Figure 2 illustrates the procedure of the framework this study proposed. The framework uses MODEA with non-dominated sorting and multitask neural network to reduce computation time and implement multi-objective optimization. A population-based evolutionary algorithm is used to search solutions in solution space, while the fitness value of every individual can be obtained from a neural network, which plays the role of target regressor. Then, the framework will be introduced in detail.

3.1. Multitask Learning Model. It is found that carrying out learning for tasks jointly can improve the performance compared with conducting them individually [40]. Thus, in this study, a multitask deep learning model is trained to target the evolutionary algorithm's feedback. Therefore, the task of the model is learning for target yield in each traffic state. Here, we consider the time consumption and electricity consumption as the targets defined in equations (4) and (5).

For performing the regression task, the input should be appropriately expressed. As in equation (2), a passing sequence can be denoted as an array including CAV ids. We define the encoding of a single CAV as follows:

$$E_i = (p_i, v_i, a_i, \mathrm{encode}(\mathrm{lane}_i)), \tag{6}$$

where p_i is the position of CAV_i from the beginning of the lane, and v_i is the speed, while a_i represents the acceleration of CAV_i, and p_i, v_i, a_i will be normalized for input into the model.

In addition, $\mathrm{encode}(\mathrm{lane}_i)$ is the encoding of the lane that the driving vehicle belongs to. The encoding method is different according to the different traffic scenarios. For the on-ramp scenario shown in Figure 1(a), the one-hot encoding is applied. However, in the intersection scenario, considering the spatial relationship, we combine with approach direction and driving direction. Figure 3 shows the encoding process that takes the scenarios in Figure 1 as an example. For instance, vehicle D is coming from the west approach, and it will turn left at the intersection, so the encoding of its lane is $(1, 0, 0, 0, 0, 1)$. Finally, a passing sequence can be formulated as the concatenation of encodings of CAVs.

When the vectorized representations of passing sequences are constructed, a neural network model can be built to take the vectors as input. Similar to TextCNN [41], we also use the convolutional neural network (CNN)

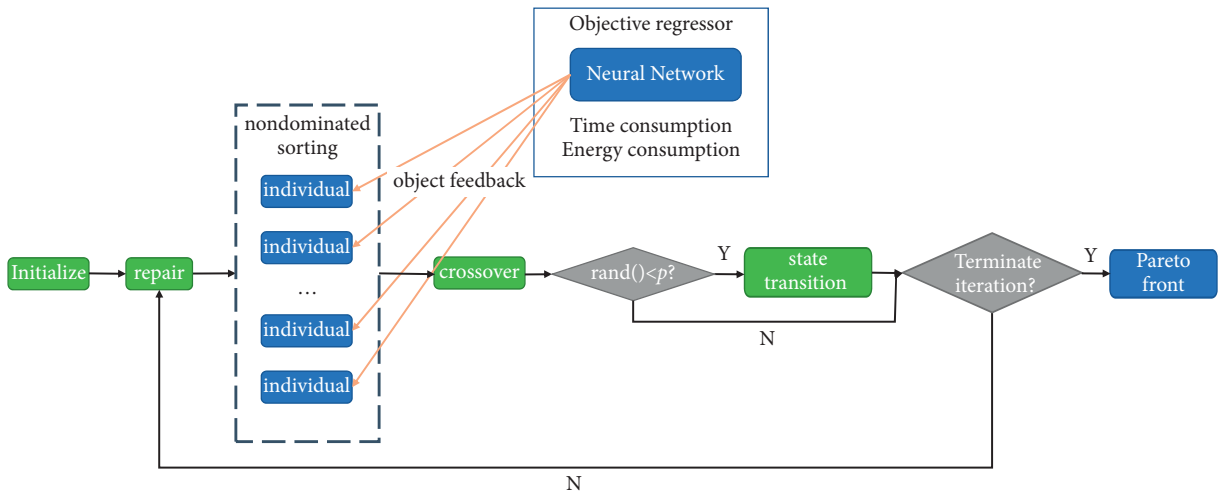


Figure 2: Flowchart of the proposed framework.

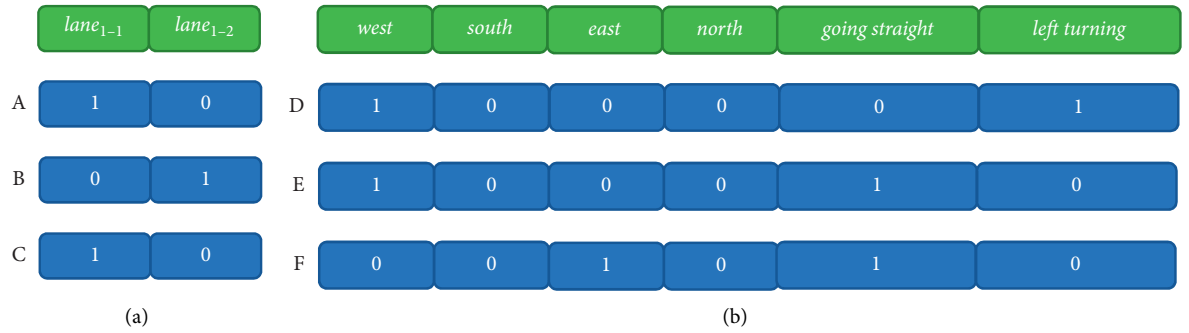


Figure 3: Encoding of lane of vehicles, which takes the state in Figure 1 as an example. (a) Example of Figure 1(a) and (b) example of Figure 1(b).

to carry out the learning process, whereas CNN can extract the features from original data automatically [42]. The structure of the CNN-based multitask learning model is shown in Figure 4. The backbone part takes sequence vectors consisting of several CAV encodings as inputs and extracts latent feature representations for them; then, the specific task part takes the feature representations as input and output time consumption and energy consumption of the sequences in a specific traffic scenario. In the backbone part, one-dimensional convolution layers with different scales of kernel size are applied to extract features.

After determining the basic structure of the neural network, the loss function should be specified to train the learning model towards the optimization goals. Here, considering the training process of two targets in two single-task models, the loss functions are considered as mean squared error (MSE), which is as follows:

$$\begin{aligned} \mathrm{MSE}_1 &= \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{f}_1(x_i) - f_1(x_i)\right)^2, \\ \mathrm{MSE}_2 &= \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{f}_2(x_i) - f_2(x_i)\right)^2, \end{aligned} \tag{7}$$

where n is the count of test samples. $\widehat{f}_1$ and $\widehat{f}_2$ are predicting values, while f_1 and f_2 are ground truth. Generally, the loss function in the multitask learning model can be defined as the naive weighted sum of losses, which is as follows:

$$L = \omega_1 \mathrm{MSE}_1 + \omega_2 \mathrm{MSE}_2, \tag{8}$$

where the loss weights ω_1 and ω_2 are uniform or manually tuned. The performance of the model highly depends on the settings of the weight parameters. Cipolla et al. pointed out that the loss function can be calculated based on maximizing the Gaussian likelihood with homoscedastic uncertainty

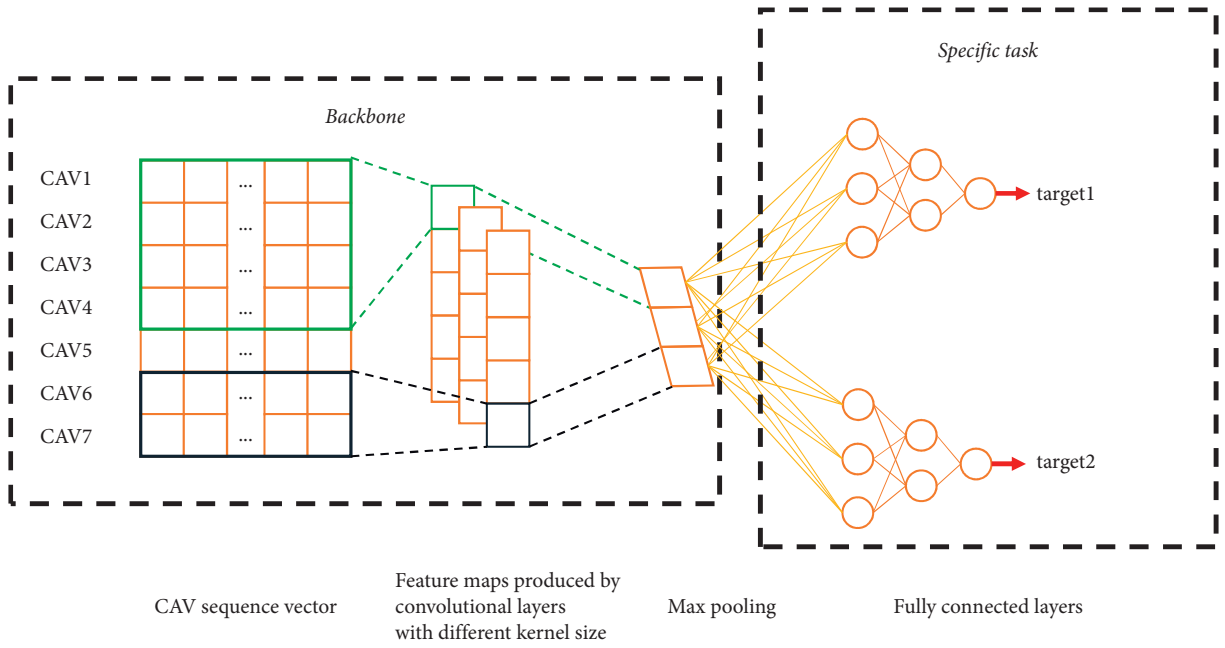


FIGURE 4: Structure of the proposed neural network. The backbone layer is for feature extracting, and the specific task layer is for different regression tasks.

[43]. As a result, let $f^W(x)$ be the outputs of neural network with weights W, and the likelihood as a Gaussian can be defined as follows:

$$p\left(f_j(x)|f^W(x)\right) = \mathrm{N}\left(f^W(x), \sigma^2\right)(j = 1, 2), \tag{9}$$

where σ is a scalar that represents observation noise. Let $\widehat{f}_j(x)$ be the sufficient statistics; then, the multitask likelihood can be derived from the following:

$$\begin{aligned} p\left(f_1(x), f_2(x)|f^W(x)\right) &= p\left(f_1(x)|f^W(x)\right) \cdot p\left(f_2(x)|f^W(x)\right) \\ &= \mathrm{N}\left(f_1(x); f^W(x), \sigma_1^2\right)\mathrm{N} \\ &\quad \cdot \left(f_2(x); f^W(x), \sigma_2^2\right). \end{aligned} \tag{10}$$

Taking logarithmic form, the new loss function can be defined as follows:

$$-\log\, p\left(f_1(x), f_2(x)|f^W(x)\right) \propto \frac{1}{2\sigma_1^2}f_1(x) - f^W(x)^2 + \frac{1}{2\sigma_2^2}f_2(x) - f^W(x)^2 + \log\, \sigma_1\sigma_2 = \frac{1}{2\sigma_1^2}\mathrm{MSE}_1 + \frac{1}{2\sigma_2^2}\mathrm{MSE}_2 + \log\, \sigma_1\sigma_2. \tag{11}$$

Notice that σ_1 and σ_2 are the denominators in equation (11). To avoid division by zero errors, the logarithmic form is used for the actual training process:

$$\varsigma_j = log\sigma_j^2 (j = 1, 2). \tag{12}$$

Finally, the loss function is given in equation (13), which can be adaptive during the training process.

$$L = \frac{1}{2}\left(e^{-\varsigma_1}\mathrm{MSE}_1 + e^{-\varsigma_2}\mathrm{MSE}_2 + \varsigma_1 + \varsigma_2\right). \tag{13}$$

4. Multi-Objective Discrete Evolutionary Algorithm

Generally, the average count of possible passing sequences in cooperative driving grows almost exponentially with the increase in numbers of CAVs in CR [30]. Thus, searching for the best solution is hard when the number of CAVs is large, so this study proposes a population-based evolutionary algorithm to obtain (near) optimal passing order from this perspective.

In multi-objective optimization problems, the Pareto optimal solution is used to select according to the practical problem [44]. The conception of the Pareto optimal solution set is introduced as below. First, in this minimization problem, solution x^0 Pareto dominates x^1 only if:

$$\begin{aligned} f_j\left(x^0\right) &\le f_j\left(x^1\right), \quad \forall j = 1, 2 \\ f_j\left(x^0\right) &< f_j\left(x^1\right), \quad \exists j = 1, 2. \end{aligned} \tag{14}$$

We use the corresponding symbol to denote the domination relationship:

$$x^0 \succ x^1, \tag{15}$$

which represents that x^0 dominates x^1. If there is not any solution that dominates x^0, then x^0 will be called the non-

dominated solution. Accordingly, the Pareto optimal solution set P_s can be defined as the set consisting of all the non-dominated solutions. Therefore, the primary purpose of the algorithm is to search corresponding Pareto optimal solutions. If there is more than one element in the Pareto optimal solution set, two kinds of heuristic strategies can be used:

(i) Delay-first strategy (DFS): always choose the solution with minimal time consumption from P_s.

(ii) Energy-first strategy (EFS): always choose the solution with minimal energy consumption from P_s.

The form of the candidate solutions in the algorithm is denoted as equation (2), while the initialization operation is generating n different integers with ranges from 1 to n. The feasible solutions make up a population in the evolutionary algorithm. Considering that lane-changing behavior is prohibited in CR, some solutions will be illegal. For example, in Figure 1(a), the passing order [C, A, B] cannot be accepted as candidate solution because A is supposed to be in front of C. Hence, a repair operation is applied to repair illegal sequence, which is defined as follows:

$$\dot{x} = M_r x, \tag{16}$$

where $\dot{x}$ represents a passing sequence that can be a candidate, and M_r is a matrix that carries out the repair operation. The matrix is constructed according to the order of vehicles on the lanes. For unfeasible sequence [3, 1, 2], which represents "C-A-B" in Figure 1(a), M_r is as follows:

$$M_r = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{17}$$

Then, [3, 1, 2] will be transformed to [1, 3, 2], which represents "A-C-B," and it will be legal.

The proposed algorithm uses selection operation, crossover operation, state transition with swap operation, shift operation, and symmetry operation for population evolution. Corresponding operations can be described as follows.

4.1. Selection Operation. Non-dominated sorting technique is used for layering individuals. Algorithm 1 shows the process of non-dominated sorting. In the algorithm, c is the non-dominated level, and X^N is the set of all the non-dominated solutions in P; fitness represents the virtual value of individuals, which is used for selection operation. Eventually, the roulette wheel method is applied to choose individuals in the population; then, the crossover operation can be carried out. In the roulette wheel method, the selection probability of individual i is defined as follows:

$$\Pr(i) = \frac{c_f - \text{fitness}(i)}{\sum_{j=1}^{c_f - 1} j}, \tag{18}$$

where c_f is the value of c after iterations in Algorithm 1.

4.2. Crossover Operation. Tie-breaking crossover is introduced in this study [45]. This operation can prevent two identical orders from appearing in a sequence, and the procedure is indicated in Figure 5. The start positions and length of subsequences are generated randomly, so the results after crossover could be with duplicated items. A crossover map will also be generated, and the crossover map is actually a random order of integers $0, 1, \ldots, n-1$. Accordingly, the new sequences after exchange can be transformed by multiplying the length of the sequence and adding the crossover map. Finally, as shown in Figure 5, offspring can be produced by sorting operation according to phase 3.

4.3. State Transition. The state transition procedure is probabilistic in the light of predefined probability value p. In this study, the value of p is set to 0.2 to keep the trade-off between exploration and exploitation. The state transition operations include swap, shift, and symmetry [35]. Swap transformation is used for randomly exchanging subsequences in passing sequences; shift transformation is used for subsequence translation, and symmetry transformation means two subsequences symmetrical about a selected central point exchange their values. These operations can be implemented by several matrixes, which can be denoted as follows:

$$x_{k+1} = M_{\text{symmetry}}\left(M_{\text{shift}}\left(M_{\text{swap}} x_k\right)\right), \tag{19}$$

where x_{k+1} is a passing sequence after k iterations. M_{symmetry}, M_{shift}, and M_{swap} represent the matrix, which implements symmetry operation, shift operation, and swap operation, respectively. Figure 6 illustrates the three transformations. The length of subsequences is a hyperparameter for swap transformation and shift transformation. The values of these two operations are generated randomly according to the number of CAVs. While for symmetry operation, the length of subsequences and the position of the symmetry center can be generated randomly. Note that the boundary condition will be processed here when the indexes of elements may be out of bounds.

5. Vehicle Control

When a passing order is determined, CAVs can move in the light of the sequence. First of all, the motion of vehicles needs to be constrained by the speed limit and acceleration ability:

$$\begin{aligned} 0 \le v_i \le v_{\max}, \\ d_{\max} \le a_i \le a_{\max}, \end{aligned} \tag{20}$$

where $v_{\max}$ denotes the maximum speed limit on the road, and $d_{\max}$ is the maximum deceleration, while $a_{\max}$ is the maximum acceleration constraint by vehicle dynamics.

The virtual vehicle mapping method is used in the framework to ensure safety [46, 47]. Taking the case in Figure 1(a) as an example, if the passing order is "A-C-B," then C will be mapped into lane_{1-1}. CAV B will then follow

Input: Init population P
(1) : $c \longleftarrow 1$
(2) : while length (P) > 0
(3) : $B \longleftarrow X^N$
(4) : **for** each $b \in B$
(5) : fitness $(b) \longleftarrow c$
(6) : end
(7) : $c \longleftarrow c + 1$
(8) : Delete B from P
(9) : end

Algorithm 1: Non-dominated sorting.

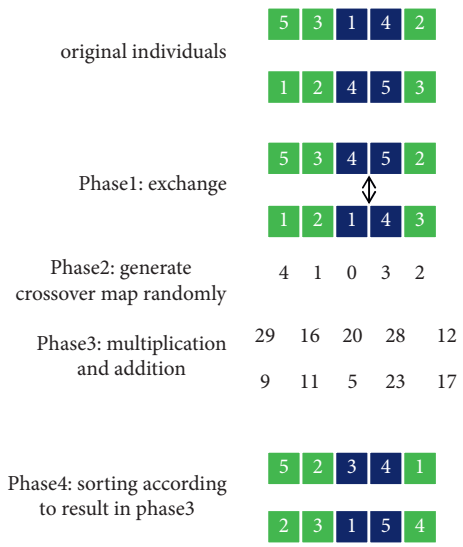


Figure 5: Procedure of tie-breaking crossover.

a virtual vehicle mapped by CAV C, which means the mode of motion of CAVs will be divided into two cases: free driving and car following, respectively. The control process of the CAVs in sequences can be given by Algorithm 2. Accordingly, Conflict is a function to judge whether there are potential conflicts between $x[i]$ and $x[j]$. Carfollowing is a function to guide vehicle $x[i]$ to follow vehicle $x[k]$. The equation of Carfollowing can be denoted as follows:

$$\min \int_{t=t_0}^{t_1} \left| p_k - p_i - \Delta s \right| \mathrm{d}t, \tag{21}$$

where t_0 is the start time and t_1 is the time when $x[i]$ arrives at the conflict zone or stop line. In addition, Δs is the value of the safe gap between two consecutive CAVs. The gap here represents the distance from the front of the following vehicle to the rear of the leading vehicle. If $x[k]$ is a real CAV, the value is set to $\ddot{\Delta} s$, and if $x[k]$ is a virtual vehicle, a correction factor should be added to it, which can be denoted as follows:

$$\Delta s = \ddot{\Delta} s + b \times L_i^M, \tag{22}$$

where b is a bool variable, if $x[k]$ is virtual, the value of b will be 1, and L_M will be the distance for $x[i]$ to cross the conflict zone.

Using Algorithm 2, the first CAV in sequence drives freely, and a CAV with a minimal relative distance with the first CAV in the rest of the sequence is chosen as car following target.

Finally, if a passing sequence is determined, it will not be altered unless the set of CAVs in CR changes.

6. Simulation and Analysis

6.1. Simulation Platform and CNN Training. This study uses the microscopic traffic simulation software SUMO to study the cooperative driving strategy in two traffic scenarios in Figure 1. Under the premise of comprehensive consideration of reality, the simulation settings are given in Table 1. The simulation step is set to 0.2 s for smoother time-continuous control. The radius of CR in the on-ramp scenario is set to 1000 m by considering the communication capability [38]. Meanwhile, we set the radius parameter to 200 m in the urban intersection scenario because the speed of vehicles in this case is slow, while 200 m is enough for vehicle braking.

First of all, more than 50000 records were collected in SUMO for each traffic scenario to serve as the training data. The records include encoding of passing sequence and the combination of two regression targets. We use message-digest algorithm 5 (MD5) to delete duplicated data to ensure the uniqueness of the records. Because the length of CAV encoding in the two scenarios is 5 and 9, respectively, the convolution kernel sizes are set to [2, 3, 4] and [2, 5, 7] to extract different scales of features. The Adam optimizer is used to optimize the weights and biases for the network, and a step decay schedule for learning rate is implemented in the training process for better performance. Accordingly, the rest of the hyperparameters (*e.g., batch size, the initial learning rate, and the scales of dense layers*) were tuned automatically by applying tree-structured Parzen estimator (TPE), which can search significantly better results compared with random search methods [48].

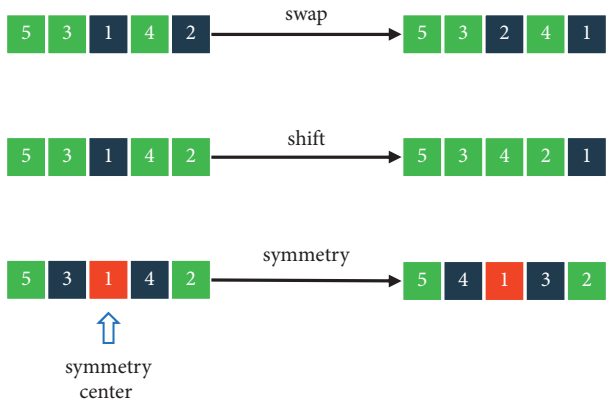


Figure 6: Sketch of swap operation, shift operation, and symmetry operation. The length of subsequences in this figure is 1.

Input: The passing sequence x
(1) : for $i \in 1$: length(x)
(2) : if $i = 1$
(3) : $a_i = (v_i - v_{\max})\&\&a_{\max}$
(4) else
(5) : $k \longleftarrow 1, rp_1 = p_1 - p_j - \Delta s$
(6) : for $j \in 2: i$
(7) : $rp_j = p_i - p_j - \Delta s$
(8) : if Conflict$(x[i], x[j])\&\&\ rp_j < rp_1$
(9) : $k \longleftarrow j$
(10) : end
(11) : Carfollowing$(x[i], x[k])$
(12) : end
(13) : end
(14) : end

Algorithm 2: Simple sequence control.

6.2. Simulation Results. To evaluate the proposed strategy comprehensively, we carried out two kinds of simulations based on the pre-trained CNN model. One is a **discrete simulation**, which is used for observing the performance of the framework under different static numbers of vehicles to be scheduled. The other is a **continuous simulation**, which is served to evaluate the framework in different traffic demand levels using the trace data exported from SUMO.

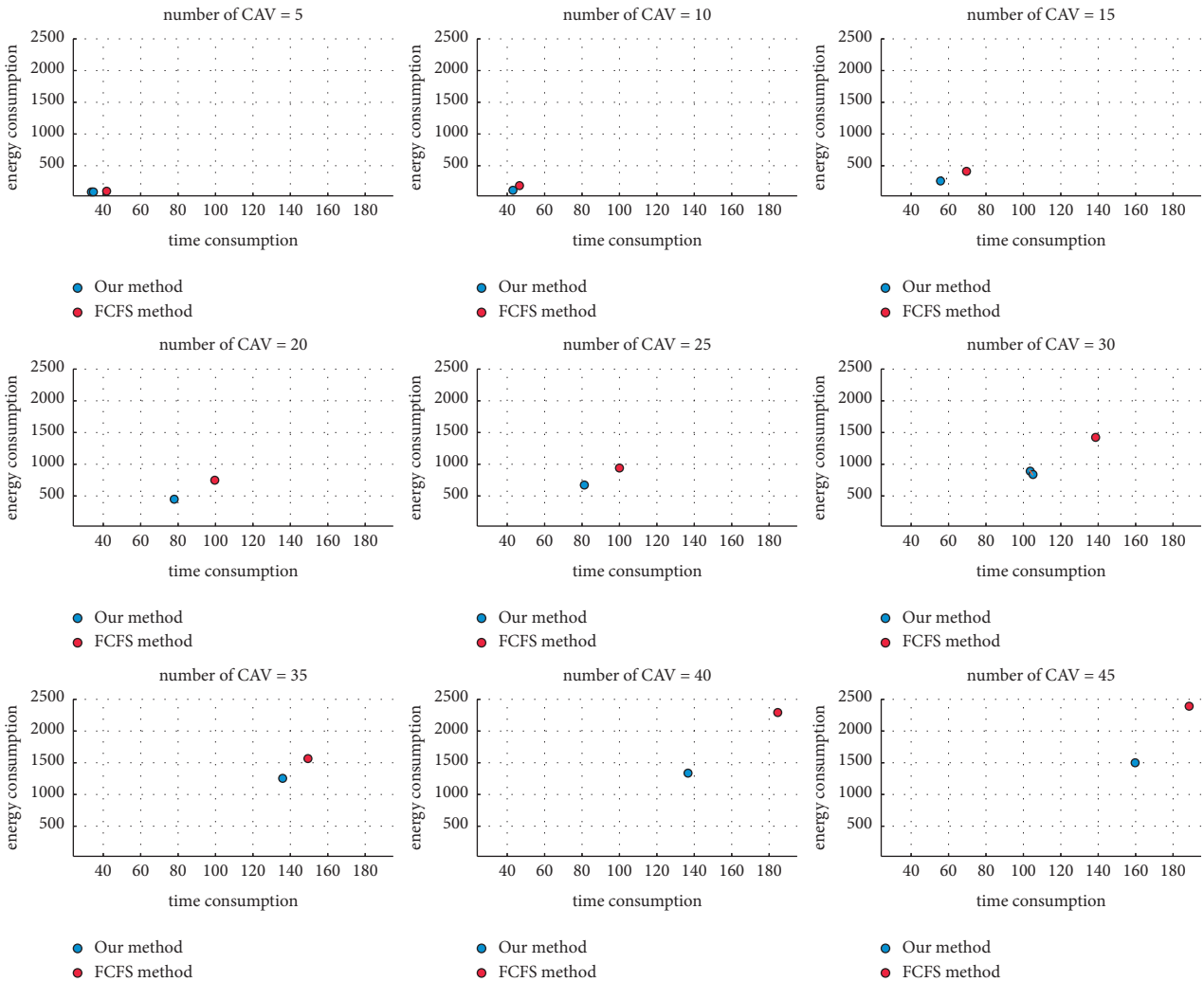
We choose the FCFS strategy as a baseline, whereas it is generally used in the domain. The iteration step and population size in MODEA are set to 30 and 40, respectively. We generate different numbers of CAVs distributed in lanes randomly for the two scenarios, and the results of the discrete simulation are shown in Figure 7. Obviously, the proposed method always has a better performance than the FCFS method. While in the on-ramp scenario, the gap between the two methods becomes more significant with the increase in CAVs. Thus, the capability of global optimization of MODEA can be verified, while the rule-based FCFS method is regarded as weak to get satisfying solutions. Meanwhile, when there is more than one solution in the Pareto front, the final sequence can be chosen manually according to specific requirements.

As for continuous simulation, different arrival rates of CAVs are deployed for 2000 simulation steps, and the trace data are exported per 4 times steps. The trace datasets include the information of CAVs such as position, speed, and acceleration, and then, we reload these data in SUMO and carry out simulations. In other words, the same trace data are used for result comparison so that the randomness can be eliminated.

All results presented are averaged over 10 independent runs, when the best results are shown in bold in Table 2. According to Table 2, there is no significant difference between DFS and EFS, which may be caused by the regression error of the neural network. However, with the increase in CAV arrival rate, the difference in results between FCFS and the proposed framework gets more remarkable. It demonstrates that the MODEA can optimize the two objectives jointly.

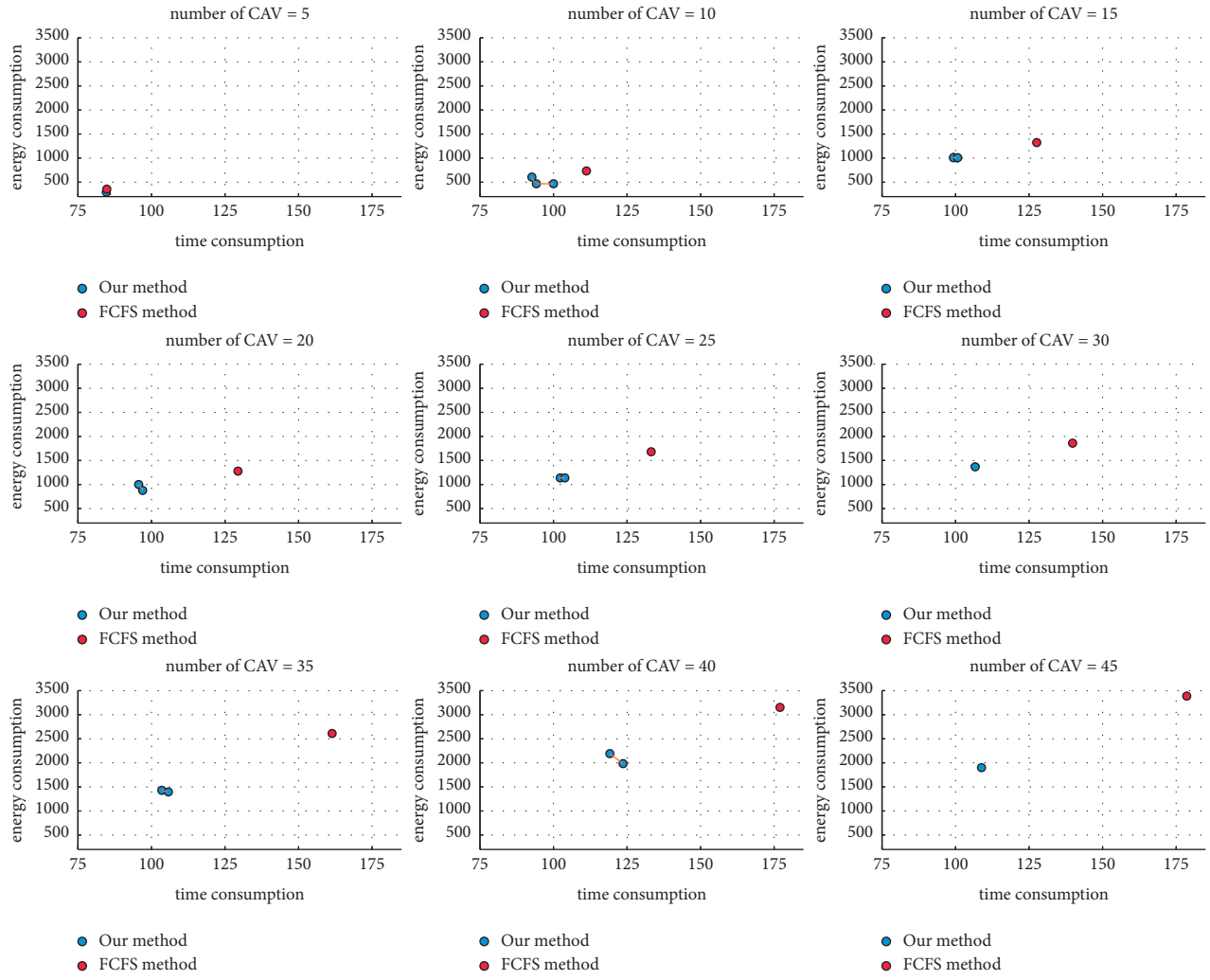
Table 1: Parameter configuration in simulation.

Item	Value	Unit
Generic settings		
Maximum acceleration of CAVs	2.6	m/s^2
Maximum deceleration of CAVs	4.5	m/s^2
Length of single lane	1000.0	m
Size of single simulation step	0.2	s
Safe gap Δs	2.5	m
For on-ramp scenario		
Road speed limit	120.0	km/h
Radius of CR	1000.0	m
For intersection scenario		
Road speed limit	60.0	km/h
Radius of CR	200.0	m



(a)

Figure 7: Continued.



(b)

Figure 7: Performance of the proposed framework compared with FCFS strategy under different CAV number circumstances. (a) For on-ramp scenario and (b) for intersection scenario.

7. Discussion about Computation Time

In cooperative driving tasks, the computation time of algorithms is vital to ensure safety and efficiency. We focus on the time performance of the proposed framework in this part, and we only consider the on-ramp scenario for evaluating computation time because the time complexity of the algorithm in the two scenarios is equal. All experiments were conducted using Julia programming language on Windows 10 operating system with Intel CORE i7-10750H CPU. Meanwhile, BenchmarkTools.jl package is used to precisely evaluate the computation time performance [49].

As Figure 8 shows, the computation time of the proposed method mainly depends on the population size of MODEA, while the number of CAVs in CR has little effect on the computation complexity, which means that we can control the computation time flexibly by setting the population size of the algorithm manually.

Meanwhile, the influence of computation time on the traffic system should be discussed. First, safety is always the most primary goal to be achieved. The impact of computing time on safety considerations will be reflected in the safe gap $\ddot{\Delta}s$. The $\ddot{\Delta}s$ can be roughly revised with the time consumption t_d:

Table 2: Simulation result comparison.

	FCFS		MODEA-DFS		MODEA-EFS		Optimization Rate (%)	
	Time cost (s)	Energy cost (Wh)	Time cost (s)	Energy cost (Wh)	Time cost (s)	Energy cost (Wh)	Time	Energy
On-ramp scenario								
400 veh/(lane · h)	71.5 ± 9.2	302.9 ± 88.0	71.1 ± 9.1	302.2 ± 88.8	71.1 ± 9.2	298.8 ± 86.8	0.56%	1.35%
600 veh/(lane · h)	75.4 ± 7.7	488.8 ± 132.4	74.6 ± 7.9	479.0 ± 129.5	75.0 ± 7.6	475.3 ± 128.0	1.06%	2.76%
800 veh/(lane · h)	80.5 ± 6.8	679.8 ± 172.8	79.0 ± 7.0	658.9 ± 168.7	79.9 ± 7.2	653.0 ± 167.2	1.86%	3.94%
1000 veh/(lane · h)	90.1 ± 7.9	933.5 ± 239.2	82.8 ± 6.6	833.3 ± 220.0	83.2 ± 6.9	831.7 ± 210.5	8.10%	10.91%
1200 veh/(lane · h)	106.6 ± 10.5	1276.3 ± 324.1	89.0 ± 8.4	1070.9 ± 277.6	90.0 ± 8.1	1067.1 ± 274.9	16.51%	16.39%
Intersection scenario								
400 veh/(lane · h)	41.4 ± 12.7	140.2 ± 49.5	41.0 ± 12.9	137.8 ± 48.3	41.7 ± 12.8	140.0 ± 49.4	0.10%	1.71%
600 veh/(lane · h)	58.6 ± 15.6	293.5 ± 150.5	56.7 ± 14.4	287.3 ± 141.3	57.1 ± 14.5	288.9 ± 142.9	3.24%	2.11%
800 veh/(lane · h)	106.5 ± 44.2	951.5 ± 661.3	98.2 ± 37.4	861.9 ± 574.0	98.6 ± 38.5	861.4 ± 572.1	7.79%	9.47%
1000 veh/(lane · h)	141.8 ± 60.9	1468.5 ± 829.0	127.9 ± 35.9	1323.5 ± 639.4	130.4 ± 38.5	1301.3 ± 610.2	9.80%	11.39%
1200 veh/(lane · h)	197.4 ± 44.9	2572.5 ± 812.1	182.3 ± 42.9	2384.1 ± 780.9	186.7 ± 57.8	2352.6 ± 779.8	7.65%	8.55%

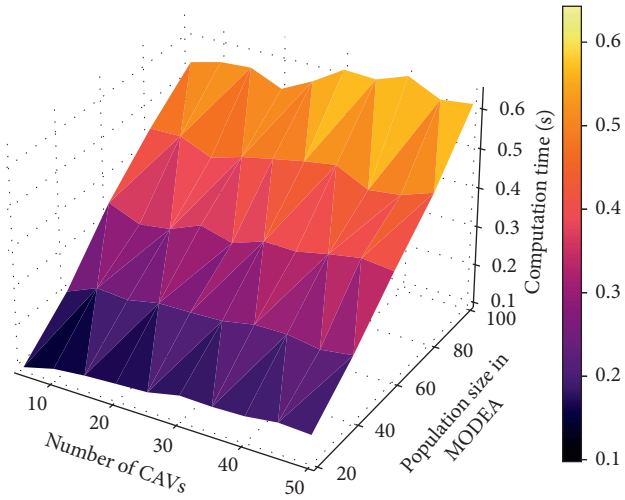


Figure 8: Average computation time of the proposed framework based on the different number of CAVs and population size in MODEA.

$$\Delta s^{'} = \ddot{\Delta s} + t_d \times v_{\max}, \tag{23}$$

where $v_{\max}$ is used for ensuring safety under any circumstance, so that Δs will be changed in simulations in terms of equation (22).

Then, we carry out a series of simulations using the same trace data exported from SUMO to compare the performance of the control framework under different computation delays. In the test, the delay caused by computation varies from 0.1 s to 0.4 s, while DFS is chosen to get solutions. Figures 9(a) and 9(b) show the time consumption and energy consumption under different circumstances. On average, the FCFS rule will outperform the proposed framework in the time consumption aspect when the computation delay reaches 0.3 s, and it will have almost identical performance in the electricity consumption aspect when the computation delay reaches 0.4 s.

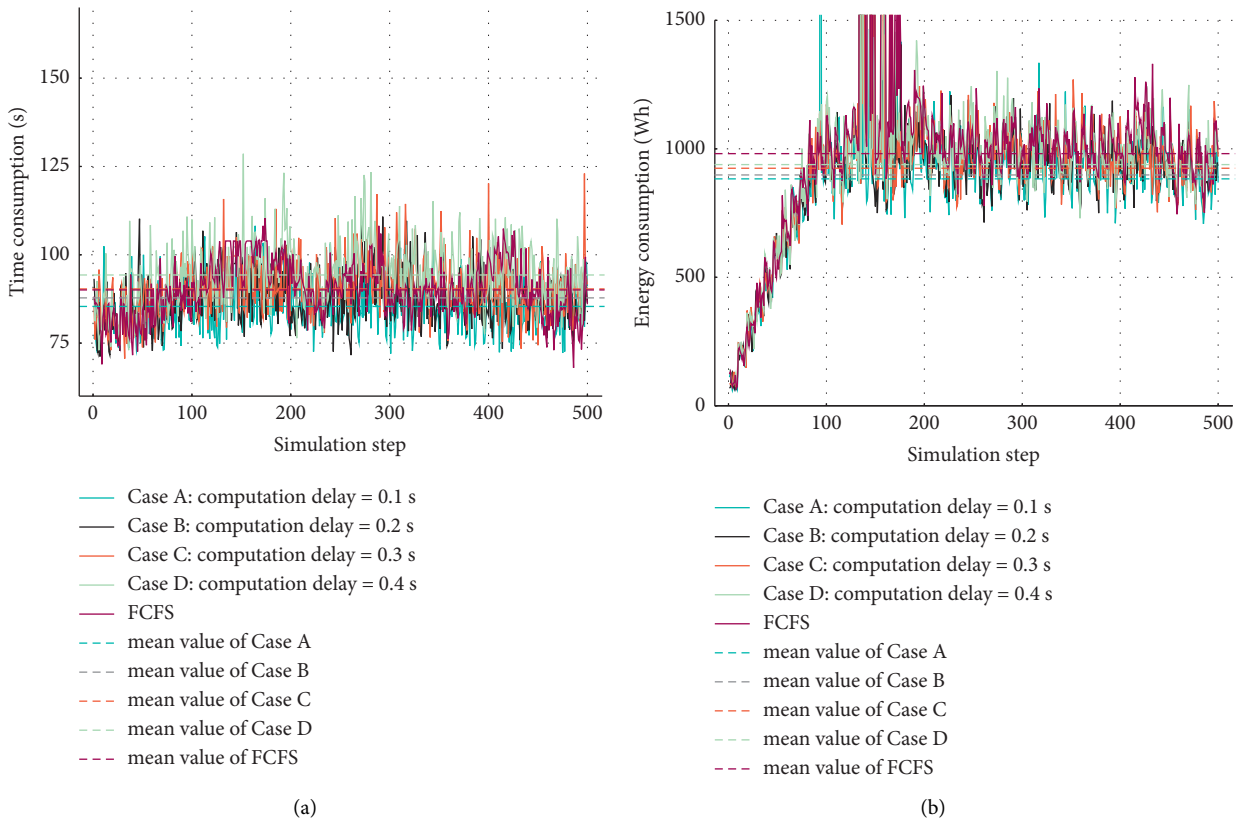


Figure 9: Comparison for values of objective function under different computation delays. (a) Time consumption and (b) electricity consumption.

8. Conclusions

Over the last few years, many methods have been put forward in the cooperative driving field, but the controllability of optimization objectives and the efficiency of algorithms are still difficult to deal with. Based on the combination of evolutionary algorithm and machine learning technique, this study proposes an intelligent framework that considers both the delay and the energy consumption of vehicles. An encoding approach of CAVs is implemented, and a passing sequence of CAVs is approximately regarded as a sentence in natural language so that the TextCNN can be applied to extract features. Compared with other frameworks, it has some significant advantages:

(i) Controllability and flexibility: the optimization objectives and computation time can be adjusted manually, and it can be instrumental under different design requirements.

(ii) General applicability: similar to FCFS protocol, the framework can be applied in different cooperative driving scenarios such as intersection and on-ramps.

In future research, a more concrete vehicle control method is supposed to be studied for practicability. Moreover, the neural network this study implements can only deal with a finite number of cases because the input length for the network is fixed. Therefore, the maximum number of CAVs must be assigned, and the zero paddings will be used if the number of CAVs is less than the predefined maximum length. Hence, the form of the neural network and CAV encodings can be further studied for better performance; for example, the encoder-decoder structure can be applied to study the cases of different numbers of CAVs. Finally, the lane-changing behavior of vehicles and pedestrian crossing rules can be considered in the system. However, a more complex but more realistic system will be put before us to study.

Data Availability

The data used to support the findings of this study are produced by simulations.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported in part by the Key R&D Program of Jiangsu Province in China (No. BE2020013), the Opening Project of Key Laboratory of Intelligent Transportation Systems Technologies, Ministry of Communications, P.R.

China (Grant No. 2020-8501), and the R&D Project of China Communications Construction Company (Grant 2019-ZJKJ-ZDZX02-2). The work of the first author was supported in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. SJCX21_0062).

References

[1] J. Zhang, S. Dong, Z. Li, B. Ran, R. Li, and H. Wang, "An eco-driving signal control model for divisible electric platoons in cooperative vehicle-infrastructure systems," *IEEE Access*, vol. 7, pp. 83277–83285, 2019.

[2] A. A. Malikopoulos and J. P. Aguilar, "An optimization framework for driver feedback systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 955–964, 2013.

[3] Y. Ma, K. Tang, S. Chen, A. J. Khattak, and Y. Pan, "On-line aggressive driving identification based on in-vehicle kinematic parameters under naturalistic driving conditions," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 554–571, 2020.

[4] B. Peng, M. F. Keskin, B. Kulcsár, and H. Wymeersch, "Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning," *Communications in Transportation Research*, vol. 1, Article ID 100017, 2021.

[5] C. Dong, H. Wang, Y. Li, W. Wang, and Z. Zhang, "Route control strategies for autonomous vehicles exiting to off-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 3104–3116, 2020.

[6] Y. Zheng, B. Ran, X. Qu, J. Zhang, and Y. Lin, "Cooperative lane changing strategies to improve traffic operation and safety nearby freeway off-ramps in a connected and automated vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4605–4614, 2020.

[7] J. Larsson, M. F. Keskin, B. Peng, B. Kulcsár, and H. Wymeersch, "Pro-social control of connected automated vehicles in mixed-autonomy multi-lane highway traffic," *Communications in Transportation Research*, vol. 1, Article ID 100019, 2021.

[8] Y. Han, M. Ramezani, A. Hegyi, Y. Yuan, and S. Hoogendoorn, "Hierarchical ramp metering in freeways: an aggregated modeling and control approach," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 1–19, 2020.

[9] H. Yu, P. Liu, Y. Fan, and G. Zhang, "Developing a decentralized signal control strategy considering link storage capacity," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102971, 2021.

[10] Y. Guo, T. Sayed, and M. Essa, "Real-time conflict-based Bayesian Tobit models for safety evaluation of signalized intersections," *Accident Analysis & Prevention*, vol. 144, Article ID 105660, 2020.

[11] B. Varga, T. Tettamanti, B. Kulcsár, and X. Qu, "Public transport trajectory planning with probabilistic guarantees," *Transportation Research Part B: Methodological*, vol. 139, pp. 81–101, 2020.

[12] M. Li, X. Wu, X. He, G. Yu, and Y. Wang, "An eco-driving system for electric vehicles with signal control under V2X environment," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 335–350, 2018.

[13] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 6, pp. 1712–1724, 2006.

[14] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 1, pp. 4–24, 2020.

[15] H. Ahn and A. Colombo, "Abstraction-based safety verification and control of cooperative vehicles at road intersections," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4061–4074, 2020.

[16] V. Dolk, J. d. Ouden, S. Steeghs et al., "Cooperative automated driving for various traffic scenarios: experimental validation in the GCDC 2016," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1308–1321, 2018.

[17] F. Yan, M. Dridi, and A. El Moudni, "Autonomous vehicle sequencing algorithm at isolated intersections," in *Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 1–6, St. Louis, MO, USA, October 2009.

[18] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.

[19] M. W. Levin and D. Rey, "Conflict-point formulation of intersection control for autonomous vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 528–547, 2017.

[20] E. R. Müller, R. C. Carlson, and W. K. Junior, "Intersection control for automated vehicles with MILP," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37–42, 2016.

[21] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: from reservation- to optimization-based methods," *Transportation Research Part B: Methodological*, vol. 122, pp. 416–435, 2019.

[22] S. Ilgin Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 2014.

[23] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, 2012.

[24] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, "A bi-level cooperative driving strategy allowing lane changes," *Transportation Research Part C: Emerging Technologies*, vol. 120, Article ID 102773, 2020.

[25] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 530–537, New York, NY, USA, July 2004.

[26] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of Artificial Intelligence Research*, vol. 31, pp. 591–656, 2008.

[27] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2259, no. 1, pp. 223–232, 2011.

[28] M. W. Levin, S. D. Boyles, and R. Patel, "Paradoxes of reservation-based intersection controls in traffic networks,"

Transportation Research Part A: Policy and Practice, vol. 90, pp. 14–25, 2016.

[29] E. Lukose, M. W. Levin, and S. D. Boyles, "Incorporating insights from signal optimization into reservation-based intersection controls," *Journal of Intelligent Transportation Systems*, vol. 23, no. 3, pp. 250–264, 2019.

[30] Y. Meng, L. Li, F.-Y. Wang, K. Li, and Z. Li, "Analysis of cooperative driving strategies for nonsignalized intersections," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 2900–2911, 2017.

[31] H. Xu, S. Feng, Y. Zhang, and L. Li, "A grouping-based cooperative driving strategy for CAVs merging problems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6125–6136, 2019.

[32] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, 2019.

[33] J. Zhang, X. Jiang, Z. Liu, L. Zheng, and B. Ran, "A study on autonomous intersection management: planning-based strategy improved by convolutional neural network," *KSCE Journal of Civil Engineering*, vol. 25, no. 10, pp. 3995–4004, 2021.

[34] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[35] X. Zhou, D. Y. Gao, C. Yang, and W. Gui, "Discrete state transition algorithm for unconstrained integer optimization problems," *Neurocomputing*, vol. 173, pp. 864–874, 2016.

[36] P. A. Lopez, E. Wiessner, M. Behrisch et al., "Microscopic traffic simulation using SUMO," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC*, pp. 2575–2582, Maui, HI, USA, November 2018.

[37] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[38] Z. Li, M. Pourmehrab, L. Elefteriadou, and S. Ranka, "Intersection control optimization for automated vehicles using genetic algorithm," *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 12, Article ID 04018074, 2018.

[39] T. Kurczveil, P. Á. López, and E. Schnieder, "Implementation of an energy model and a charging infrastructure in SUMO," in *Simulation of Urban MObility User Conference*, pp. 33–43, Springer, Berlin, Heidelberg, 2013.

[40] K. Zhang, L. Zheng, Z. Liu, and N. Jia, "A deep learning based multitask model for network-wide traffic speed prediction," *Neurocomputing*, vol. 396, pp. 438–450, 2020.

[41] Y. Kim, "Convolutional Neural Networks for Sentence Classification," arXiv preprint arXiv:1408.5882, 2014.

[42] Y. Liu, C. Lyu, X. Liu, and Z. Liu, "Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2349–2358, 2021.

[43] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, Salt Lake City, UT, USA, June 2018.

[44] J. Mou, "Intersection traffic control based on multi-objective optimization," *IEEE Access*, vol. 8, pp. 61615–61620, 2020.

[45] P. W. Poon and J. N. Carter, "Genetic algorithm crossover operators for ordering applications," *Computers & Operations Research*, vol. 22, no. 1, pp. 135–147, 1995.

[46] E. G. Debada and D. Gillet, "Virtual vehicle-based cooperative maneuver planning for connected automated vehicles at single-lane roundabouts," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 4, pp. 35–46, 2018.

[47] A. Uno, T. Sakaguchi, and S. Tsugawa, "A merging control algorithm based on inter-vehicle communication," in *Proceedings of the 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, vol. 5, pp. 783–787, Tokyo, Japan, October 1999.

[48] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2546–2554, Granada, Spain, Decemeber 2011.

[49] J. Chen and J. Revels, "Robust benchmarking in noisy environments," arXiv preprint arXiv:1608.04295, 2016.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9513170, 10 pages
https://doi.org/10.1155/2021/9513170

Research Article

The Prediction of Multistep Traffic Flow Based on AST-GCN-LSTM

Fan Hou,[1] Yue Zhang,[1] Xinli Fu,[3] Lele Jiao,[1] and Wen Zheng [1,2]

[1]*Institute of Public-safety and Big Data, College of Data Science, Taiyuan University of Technology, Taiyuan 030060, China*
[2]*Center for Big Data Research in Health, Changzhi Medical College, Changzhi 046000, China*
[3]*College of Foreign Languages, Taiyuan University of Technology, Taiyuan 030060, China*

Correspondence should be addressed to Wen Zheng; zhengwen@tyut.edu.cn

Received 30 August 2021; Revised 11 October 2021; Accepted 4 December 2021; Published 24 December 2021

Academic Editor: Peng Hang

Aiming at the traffic flow prediction problem of the traffic network, this paper proposes a multistep traffic flow prediction model based on attention-based spatial-temporal-graph neural network-long short-term memory neural network (AST-GCN-LSTM). The model can capture the complex spatial dependence of road nodes on the road network and use LSGC (local spectrogram convolution) to capture spatial correlation features from the K-order local neighbors of the road segment nodes in the road network. It is more accurate to extract the information of neighbor nodes by replacing the single-hop neighborhood matrix with K-order local neighborhoods to expand the receptive field of graph convolution. The high-order neighborhood of road nodes is also fully considered instead of only extracting features from first-order neighbor nodes. In addition, an external attribute enhancement unit is designed to extract external factors (weather, point of interest, time, etc.) that affect traffic flow in order to improve the accuracy of the model's traffic flow prediction. The experimental results show that when considering the static, dynamic, and static and dynamic combination, the model has excellent performance: RMSE (4.0406, 4.0362, 4.0234), MAE (2.7184, 2.7044, 2.7030), accuracy (0.7132, 0.7190, 0.7223).

1. Introduction

Traffic forecasting is an important field in the research of intelligent transportation [1], and effective traffic flow forecasting can alleviate traffic congestion, travel planning, and traffic management for individual drivers and decision-makers [2, 3]. The complex temporal and spatial correlations between traffic flows will show huge differences affected by external emergencies [4], dynamic factors, and static factors. Ahmed [5] and others proposed an autoregressive integrated moving average model (ARIMA) model that can only deal with nonstationary time series data. It is difficult to explore connections between dynamic data and is no longer suitable for current application scenarios. In addition, though traditional linear methods such as a series of Kalman filtering methods proposed and improved by Stephanedes [6], Xie (2007) [7], Ojeda (2013) [8], Guo [9] have improved the accuracy of traffic prediction in some aspects, its ability to fit nonlinear traffic flow data is still poor, and it increases the prediction time [10–12].

With the development of computer capabilities, typical machine learning methods, such as support vector regression (SVR) [13, 14], k-nearest neighbor algorithm [15, 16] K-NN (K-NearestNeighbor), and decision tree models [17–19], can dig out the essential laws and rich information hidden in traffic flow from massive data [20], and better promote the development process of traffic flow forecasting.

The emergence of deep neural network models has enabled the development of the potential of artificial intelligence in traffic prediction. Although some simple network structures can improve the accuracy of model traffic prediction [21], there are problems such as slow convergence, prone to over-fitting, and prone to error values [22]. Compared with the traditional neural network model, recurrent neural network (RNN) [23], long short-term memory network (LSTM) [24], and gate recurrent unit

(GRU) [25] can effectively use the self-loop system and learn time series features to improve the effectiveness of prediction. Therefore, it is used as a component of each model to predict traffic speed, travel time, traffic flow, etc.

In order to capture the spatial dependencies in the traffic road network, researchers [26] extract spatial features combined with convolutional neural networks (CNN) from two-dimensional spatiotemporal traffic data. The description of the traffic structure using two-dimensional spatiotemporal data is not accurate and does not conform to the complex road network conditions in real life so some scholars [27] have begun to try to convert the structure of the traffic network into images and use CNN to learn the traffic images in order to capture the spatial characteristics. However, there is more or less noise in the images converted by the traffic network structure, and the existence of noise will inevitably cause CNN to capture false spatial relationships. Traditional methods based on CNN cannot essentially deal with the topological structure and physical properties of the traffic network. Recent studies [28, 29] also tried to convert the traffic state data into a three-dimensional (3D) matrix and use 3D convolutional networks to extract characteristics in deeper levels. Researchers [30] learned the traffic network as a graph and extract features from the graph structure of the traffic network using convolution operators based on graphs, which effectively learns the changes in traffic flow under the temporal and spatial attributes and achieves great forecast results.

After considering the dynamic change characteristics of the traffic network, this paper proposes the AST-GCN-LSTM model, which can predict the future traffic state according to historical traffic flow information on roads and external auxiliary information. Since traffic flow is affected by a variety of external factors (actual factors) such as weather, holiday, and time, in this article, we predict the traffic speed in the future based on the traffic speed in the past period of time and the external factors that affect the traffic flow. This is of great significance for realizing dynamic traffic signal optimization, dynamic traffic management planning, and traffic management decision [31].

2. Method

2.1. The Introduction of Basic Algorithms

2.1.1. Graph Neural Network (GCN). The transportation network can be regarded as a graph composed of nodes and edges, so the transportation network as a graph structure has been used for dynamic shortest path routing [32], traffic congestion analysis [33], and dynamic traffic allocation [34].

The most commonly used method for our research on graph networks is to introduce a spectrum frame in the spectrum domain [35] and obtain the spectrogram convolution model by designing the spectrum convolution based on the graph Laplacian matrix. In order to reduce the number of parameters and save the amount of calculation, we use the local spectrogram convolution with polynomial filter, but the Laplacian matrix power operation still requires a lot of calculation and high complexity, and to reduce the complexity, the Chebyshev polynomial is introduced to calculate the K-order local convolution, which can reduce the computational time complexity from the square level to the linear level.

As shown in Figure 1, the spectrogram convolution model using Chebyshev polynomial approximation can capture features from the K-order local neighbors of the vertices in the graph, fully taking into account the high-order neighborhood of the node instead of extracting features from the single-hop neighborhood only. This chapter expands the receptive field of graph convolution by replacing the single-hop neighborhood matrix with the K-order local neighborhood, which can extract the information of neighbor nodes more accurately.

2.1.2. GCN-LSTM Structure. To capture the complex spatial correlation and dynamic time correlation of traffic data in the real world, we have added a long- and short-term memory neural network LSTM. LSTM is an improved recurrent neural network (RNN), and LSTM has better performance than ARIMA when the training time series is long enough [36, 37]. The basic unit of the hidden layer of LSTM is a special cell unit, not a traditional neuron node. It is this special memory unit that enables LSTM to successfully solve the defect of RNN gradient explosion and also capture the temporal correlation of traffic flow. The overall structure of the GCN-LSTM structure is shown in Figure 2. In order to capture the complex spatial correlation and dynamic time correlation of traffic data in the real world, we combine the GCN with LSTM models. The function of the GCN model is to generate a graph of the traffic information of the road segment based on a given graph structure. It learns the representation of the road segment by integrating the characteristics of the local neighbors of the node and captures the spatial dependence of these road segments in the road network at each timestamp. Then, these time-varying feature representations are input into the LSTM model to capture the time dependence [38, 39].

2.2. AST-GCN-LSTM Spatiotemporal Graph Convolution Model

2.2.1. Attribute Augmentation Unit. On the basis of the GCN-LSTM traffic flow model introduced in Section 2.1.2, we have added an attribute augmentation unit. As shown in Figure 3, static external attribute features and dynamic external attribute features expand the dimensions of the original traffic feature matrix through attribute augmentation units.

At time t, traffic information matrix X_t is extracted from the historical feature matrix $\mathcal{X} = (X_1, \ldots, X_t)$, and $\{L^{t-w}, L^t\}$ is the set of dynamic attribute features of $w+1$ time windows. In different timestamps, the static attribute feature set of H is always unchanged. L and H are merged to generate an augmented matrix P^t. The problem of multistep traffic flow prediction can be expressed as

$$\widehat{y} = f(A, X, P). \tag{1}$$

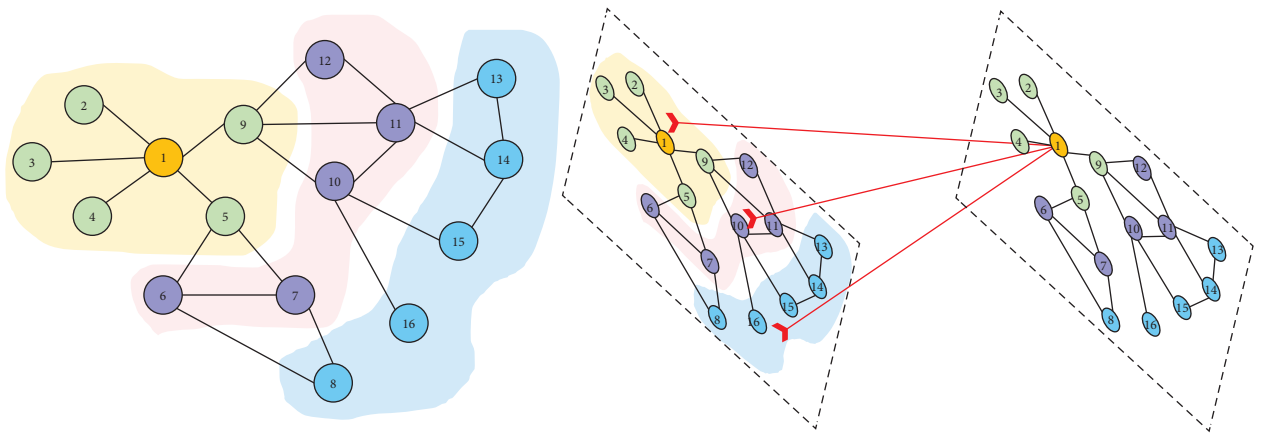


Figure 1: K-hop neighbors of graph convolution.

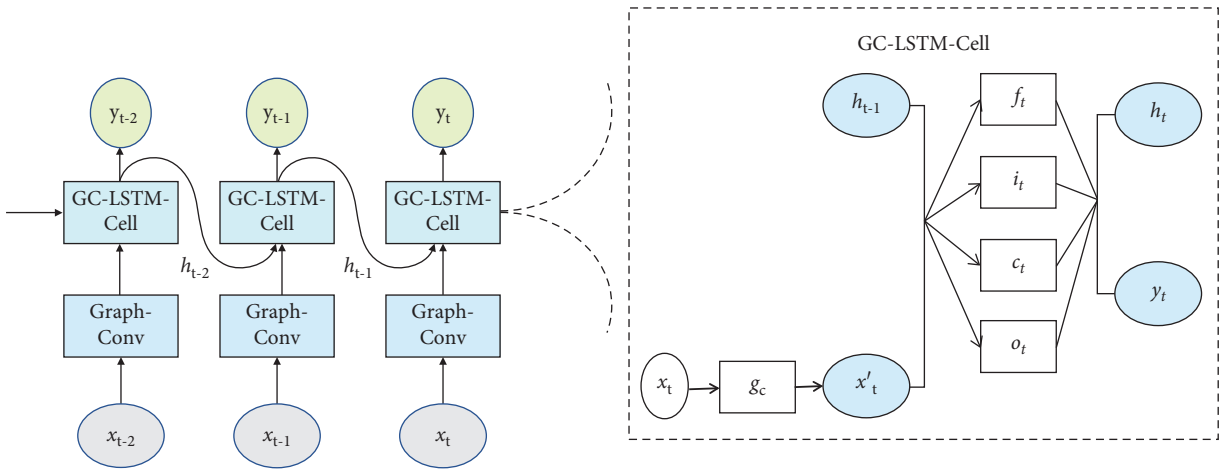


Figure 2: The architecture of the GCN-LSTM model. i_t is the input gate, o_t is the output gate, f_t is the forget gate, and c_t is the input unit state.

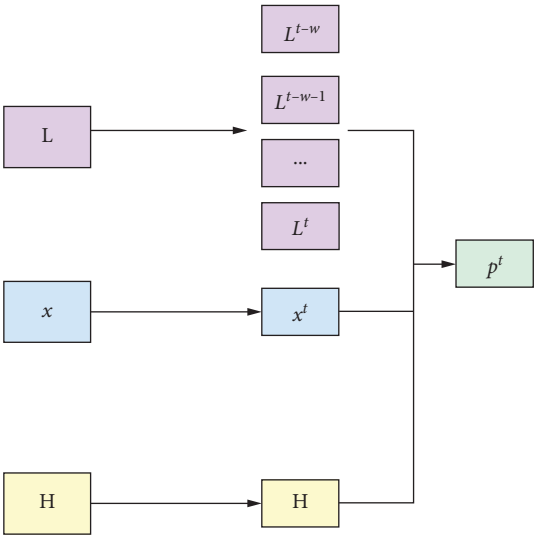


Figure 3: The Attribute augmentation unit.

The model learns the complex spatial dependence, dynamic time dependence, and external dependence in traffic data. In this model, the gate structure and hidden state in LSTM are unchanged, but the input is replaced by the graph convolution feature. At time t, the input gate, output gate, forget gate, and input unit are defined as formulas (2) to (7):

$$f_t = \sigma\left(W_f \cdot \left[g_c\left(p_t, A\right), h_{t-1}\right] + b_f\right), \tag{2}$$

$$i_t = \sigma\left(W_i \cdot \left[gc\left(P_t, A\right), h_{t-1}\right] + b_i\right), \tag{3}$$

$$\tilde{c}_t = \tanh\left(W_c \cdot \left[gc\left(P_t, A\right), h_{t-1}\right] + b_c\right), \tag{4}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}, \tag{5}$$

$$o_t = \sigma\left(W_\circ \cdot \left[gc\left(P_t, A\right), h_{t-1}\right] + b_\circ\right), \tag{6}$$

$$h_t = o_t \circ \tanh\left(c_t\right). \tag{7}$$

The sign "•" is a matrix multiplication operator, W_f, W_i, W_c, and W_o are weight matrices that map the input to the states of three gates and input units, and b_f, b_i, b_c, and b_o are four deviation vectors. s is the activation function of the gate, which is usually the sigmoid function. Tanh is the hyperbolic tangent function, and $gc(\cdot)$ represents the graph convolution operation (Chebyshev polynomial approximation).

2.2.2. Loss Function. In the process of model training, loss is chosen as the training target to optimize the error of multistep prediction and make the prediction result close to the real traffic state. Therefore, the loss function used in multistep traffic prediction AST-GCN-LSTM can be expressed as follows:

$$\text{Loss} = L\left(y_T, \hat{y}_T\right) = \left\| y_T - \hat{y}_T \right\| + \rho L_{\text{reg}}. \tag{8}$$

$L(\bullet)$ is a function to calculate the error between the predicted $\hat{y}_T$ value and the true value y_T. Here, L_{reg} represents a regular term avoiding over-fitting of the model, and ρ is a hyperparameter that is learnable in the network.

2.2.3. AST-GCN-LSTM Spatiotemporal Graph Convolution Model. The GCN-LSTM traffic flow model introduced in Section 2.1.2 is combined with the attribute expansion unit in Section 2.2.1. A multistep traffic flow prediction model (AST-GCN-LSTM) that considers external factors is also proposed. This model fully takes the external attribute characteristics that affect the traffic flow into account. Figure 4 shows the overall framework of the model, which is mainly composed of data preprocessing, attribute expansion, and spatiotemporal graph convolutional layers. In the model, we set the number of neural units in all hidden layers to 64, the batch size to 64, the learning rate to 0.001, the order of the Chebyshev polynomial to 3, and the maximum number of training iterations to 3000. The Adam optimizer is used to train the model. The data set is divided into two parts, 80% of the data are used for training, and 20% of the data are used for testing. After dividing the data set into two parts, we generate sequence samples through a time window whose width is $T + T'$.

2.3. Data Set. The traffic speed data set of the real-world road network is used in this article to evaluate the model performance. This public data set contains the taxi trajectory data of every 15 minutes setting on 156 roads from January 1 to January 31, 2015. The data sampling location is Luohu District, Shenzhen, Guangdong Province. The data mainly include the following 4 parts:

(1) Adjacency matrix: the data set selects 156 roads, so the size of the adjacency matrix A is $156 * 156$. The adjacency matrix represents the connectivity between segments. Each row of the matrix represents a road. If there are links connecting nodes i and j, then the element in the adjacency matrix $A_{i,j} = 1$. If there are no links connecting nodes i and j, the element in the adjacency matrix $A_{i,j} = 0$.

(2) Feature matrix: the feature matrix size of the data set is $2976 * 156$. The feature matrix is the speed value of 156 roads in 31 days. Each column represents a road, and each row represents the traffic speed value of 156 roads at a certain time τ. Speed information is collected every 15 minutes.

(3) Static attribute characteristic matrix: the point-of-interest (POI) information on 156 roads is provided in the data set. POI categories include the following nine types: catering, business, shopping, transportation, education, life, medical care, accommodation, and others. When determining the POI category of each road, the POI distribution on each road is calculated firstly, and then, proportions of the various categories of POIs are calculated. After comparing the proportions of the various categories of POIs, the POI with the largest proportion is used as the static feature of the road. The size of the static attribute feature matrix is $156 * 1$.

(4) Dynamic attribute feature matrix: weather conditions of every 15 minutes in January are provided by the data set, which can be divided into five categories: light rain, heavy rain, cloudy, foggy, and sunny. Time information includes the time of day, weekdays, and weekends. Because they will have a significant impact on the traffic state, this section also takes it into consideration. The size of the three types of external attribute feature matrices is $2976 * 156$.

2.4. Evaluation Index. Three commonly used traffic forecasting indicators are as follows: mean absolute error (MAE), accuracy, and root mean square error (RMSE) [31] are used to evaluate the performance of the proposed model and the comparison model. The formula is from (9) to (11):

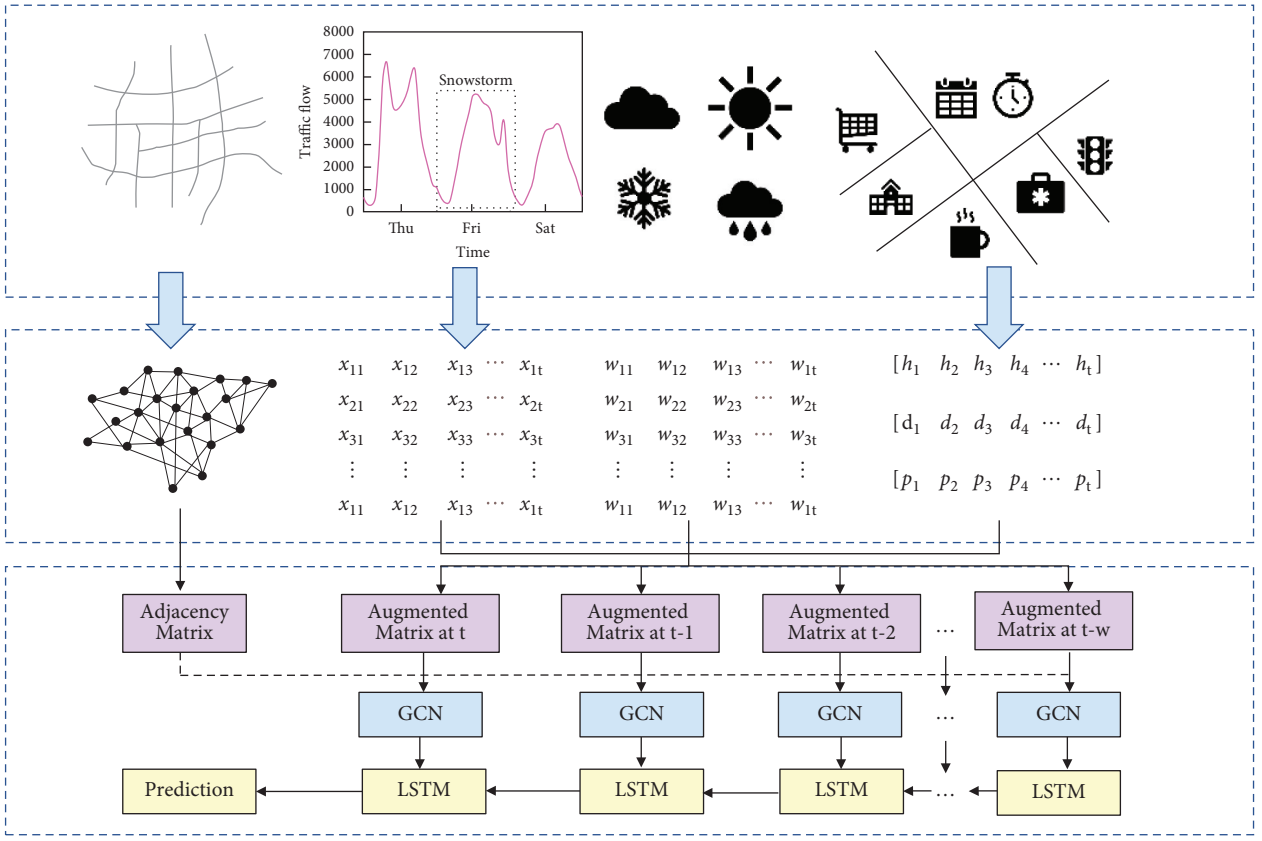


Figure 4: The architecture of the AST-GCN-LSTM model.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_T - \hat{y}_T\right|, \tag{9}$$

$$\mathrm{accuracy} = 1 - \frac{\left\|y_T - \hat{y}_T\right\|_F}{\left\|y_T\right\|_F}, \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_T - \hat{y}_T\right)^2}. \tag{11}$$

where n is the total number of test sets and y_T and $\hat{y}_T$ represent the true and predicted values of the flow.

3. Analysis of Experimental Results

3.1. Analysis of Static Attribute. In order to evaluate the overall performance of our proposed AST-GCN-LSTM model, we compare it with other traditional and common models. These models are as follows:

(1) Historical average model (HA): HA models the traffic flow as a seasonal cyclical process and uses the average value of the previous seasons (for example, the flow value of the same time period in the previous days) as the predictive value

(2) Autoregressive integrated moving average model (ARIMA): the autoregressive integrated moving average model (ARIMA) with Kalman filter is widely used in time series forecasting. It predicts the series by fitting time series data.

(3) Support vector regression (SVR): linear support vector machine is used to predict the regression task of traffic flow sequence

(4) Diffusion convolution recurrent neural network (DCRNN): diffusion convolution recurrent neural network formulates the diffusion process in graph convolution and uses a two-way random walk to capture the spatial correlation of the traffic flow in graphs. An encoder-decoder is used to capture the temporal correlation of the traffic flow, and the diffusion convolution GCN is combined with the recursive model in prediction.

(5) GCN-LSTM: the combination of LSGC and LSTM model using Chebyshev polynomial approximation is introduced in Section 2.1.1.

Among them, HA, ARIMA, and SVR are traditional nonneural network models, DCRNN is a deep learning model that can capture spatial features, and GCN-LSTM is a deep learning model that comprehensively considers the spatial features and dynamic correlation of traffic data.

Table 1 shows the overall prediction performance of the AST-GCN-LSTM model and five representative methods. Three indicators, root mean square error (RMSE), mean

Table 1: Performance comparison of different methods.

T (min)	Metrics	HA	ARIMA	SVR	DCRNN	GCN-LSTM	AST-GCN-LSTM
15	RMSE	4.2951	7.2406	4.1455	4.5000	4.1193	4.0234
	MAE	2.7815	4.9824	2.6233	3.1700	2.7701	2.7030
	Accuracy	0.7008	0.4463	0.7112	0.2913	0.7129	0.7223
30	RMSE	4.2951	6.7899	4.1628	4.5600	4.1207	4.0508
	MAE	2.7815	4.6765	2.6875	3.2300	2.7739	2.7244
	Accuracy	0.7008	0.3845	0.7100	0.2970	0.7126	0.7196
45	RMSE	4.2951	6.7852	4.1885	4.6000	4.1252	4.0587
	MAE	2.7815	4.6734	2.7359	3.2700	2.7753	2.7346
	Accuracy	0.7008	0.3847	0.7082	0.3021	0.7123	0.7172
60	RMSE	4.2951	6.7708	4.2156	4.6400	4.1262	4.0689
	MAE	2.7815	4.6655	2.7751	3.3100	2.7811	2.7403
	Accuracy	0.7008	0.3851	0.7063	0.3069	0.7119	0.7165

absolute error (MAE), and accuracy (accuracy) evaluation, are used in the comparison of performances.

According to Table 1, it can be concluded that, from the results of the 15-minute prediction window, compared with the traditional models, HA, ARIMA, and SVR, the RMSE value of the AST-GCN-LSTM model decreases by 3.07%, 44.43%, and 2.95%. Compared with the HA model and the SVR model, the accuracy value is increased by 14.69% and 1.56%, respectively. This shows that HA, ARIMA, and SVR cannot compete with other methods because the data have complex spatiotemporal correlation and high-dimensional features, and nonneural network methods are not suitable for such network-wide prediction tasks Accounting for external attribute features, the RMSE value of the AST-GCN-LSTM model that takes all external attributes into account is 10.59% and 2.33% lower than that of the DCRNN model and GCN-LSTM model. The value of MAE is lower than that of the DCRNN model and GCN-LSTM model and reduces by 14.73% and 2.42%. According to Table 1, compared with traditional methods and other methods based on deep learning, the model proposed by this article has achieved significant improvements proving the effectiveness of the model.

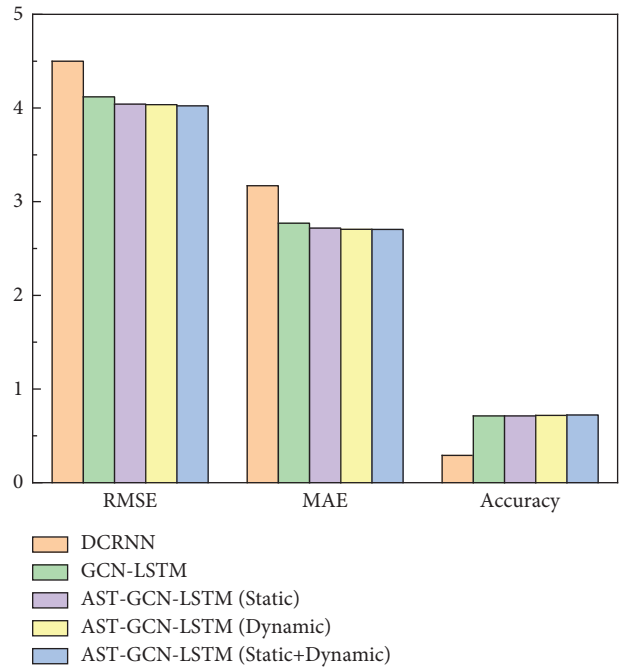


Figure 5: Experiments under different conditions.

3.2. Analysis of the External Attribute. In order to verify the influence of external attribute characteristics in traffic flow prediction, corresponding comparative experiments are done. The experimental settings are divided into four kinds as follows: adding static attribute characteristics only, adding dynamic attribute characteristics only, adding dynamic and static external attribute characteristics at the same time, and not adding external attributes characteristics. The results are shown in Figure 5. Yellow is the result of adding static attribute characteristics. Gray is the result of adding dynamic attribute features. Blue is the result of adding dynamic and static external attributes at the same time.

It can be seen from Figure 5 that when only dynamic attribute features are considered, the value of AST-GCN-LSTM (dynamic) RMSE is 10.31% and 2.02% lower than that of DCRNN and GCN-LSTM models. The value of MAE is lower than that of DCRNN and GCN-LSTM models and reduced by 14.69% and 2.37%. When only static attributes are considered, the value of AST-GCN-LSTM (static) RMSE is reduced by 10.21% and 1.91% compared with DCRNN and GCN-LSTM models, and the value of MAE is reduced by 14.25% and 1.87% compared with DCRNN and GCN-LSTM models. When static factors and dynamic factors are considered at the same time, the RMSE value of the AST-GCN-LSTM model is reduced by 10.59% and 2.33% compared with the DCRNN model and the GCN-LSTM model, and the value of MAE is reduced by 14.73% and 2.42% compared with the DCRNN model and the GCN-LSTM model.

It can be seen from Figure 5 that the model performance when only dynamic attribute features are considered is better than the model performance when only static attribute features are considered. This also indirectly illustrates the importance of considering dynamic external attribute features, and we also observed that when static and dynamic factors are considered at the same time, the performance of the model is optimal.

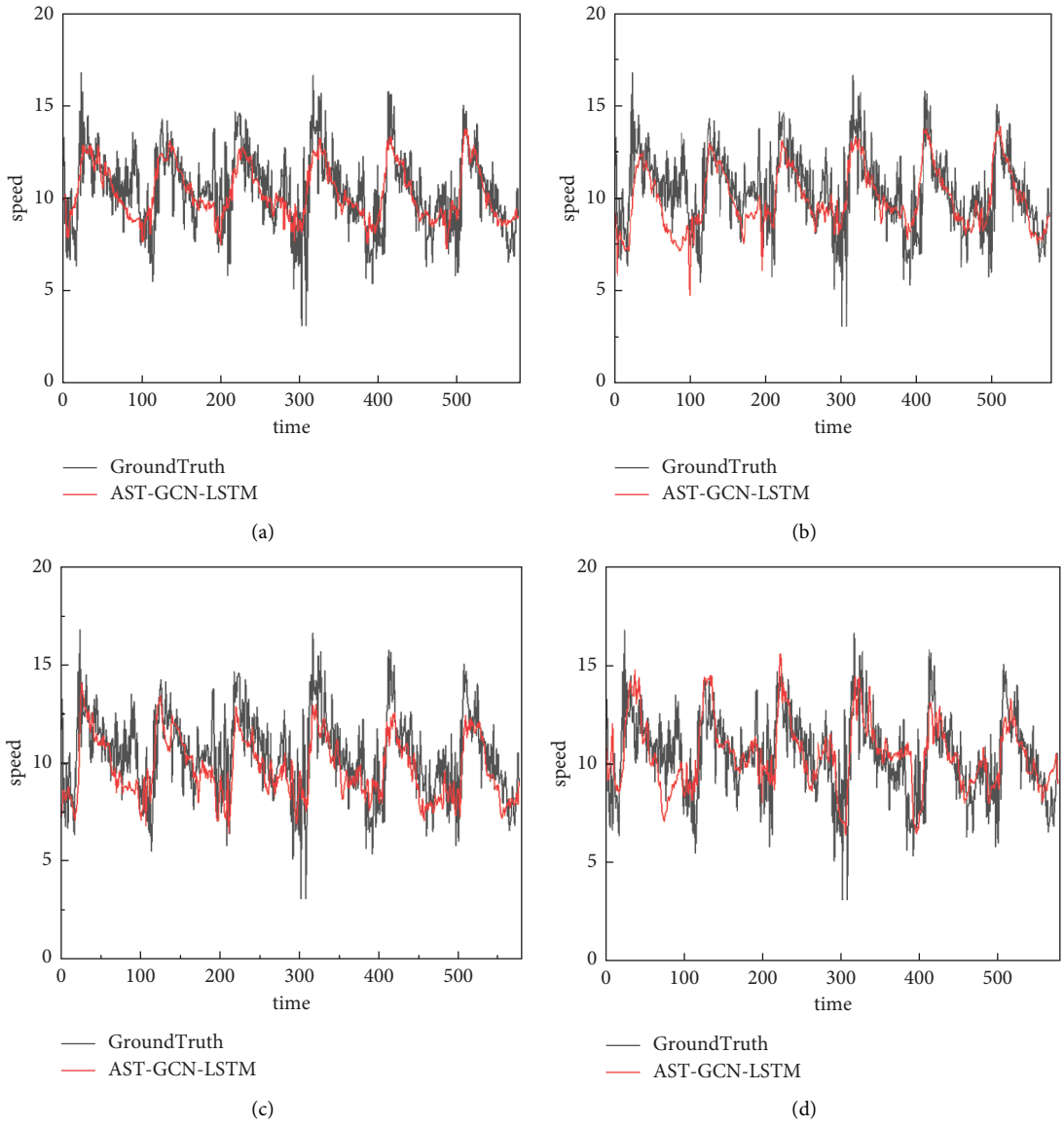


Figure 6: Visualization of the prediction window. (a–d) The prediction effect of 15, 30, 45, and 60 minutes.

In summary, considering the external information has a good effect on the prediction of the model under actual conditions.

3.3. Performance in Different Forecast Periods. All tests of this model use 60 minutes as the historical time window, which means four observation data points are used to predict the traffic conditions in the future 15, 30, 45, and 60 minutes ($H = 1, 2, 3, 4$). Figure 6 shows the visualization results of 15-, 30-, 45-, and 60-minute forecast windows. Each graph is the prediction result from January 26, 2015, to January 31, 2015.

It can be seen from Table 2 that when the traffic flow prediction window is 15 minutes, the RMSE value of the AST-GCN-LSTM model is reduced by 10.59% and 2.33% compared with the DCRNN model and the GCN-LSTM

Table 2: Performance comparison of different prediction durations.

T (min)	Metric	DCRNN	GCN-LSTM	AST-GCN-LSTM
15	RMSE	4.5000	4.1193	4.0234
	MAE	3.1700	2.7701	2.7030
	Accuracy	0.2913	0.7129	0.7223
30	RMSE	4.5600	4.1207	4.0508
	MAE	3.2300	2.7739	2.7244
	Accuracy	0.2970	0.7126	0.7196
45	RMSE	4.6000	4.1252	4.0587
	MAE	3.2700	2.7753	2.7346
	Accuracy	0.3021	0.7123	0.7172
60	RMSE	4.6400	4.1262	4.0689
	MAE	3.3100	2.7811	2.7403
	Accuracy	0.3069	0.7119	0.7165

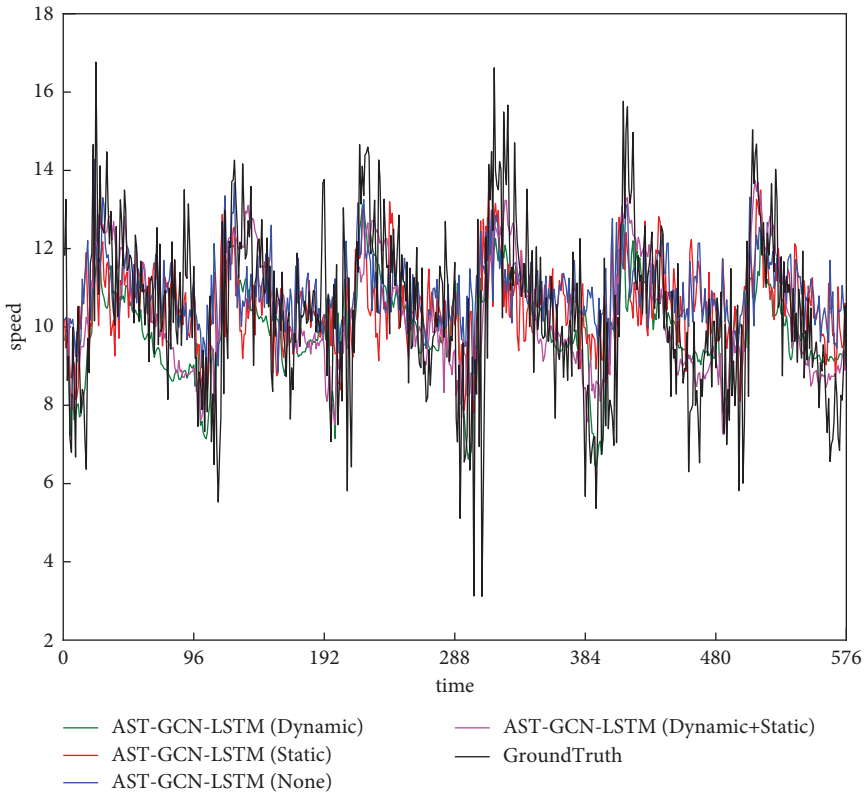


Figure 7: Comparison among forecasting enhanced by different external information. The green line is the dynamic attribute prediction of the model, the red line is the static attribute prediction of the model, the blue line is the model without external attribute prediction, the pink line is the model dynamic and static attribute prediction, and the black line is the real traffic flow.

model. Compared with the DCRNN model and the GCN-LSTM model, the value of MAE is reduced by 14.73% and 2.42%. When the traffic flow prediction window is 30 minutes, the RMSE value of the AST-GCN-LSTM model is decreased by 11.17% and 1.70% compared with the DCRNN model and the GCN-LSTM model. The MAE value is decreased by 15.65% and 1.78% compared with the DCRNN model and the GCN-LSTM model. When the traffic flow prediction window is 45 minutes, the RMSE value of the AST-GCN-LSTM model is 11.77% and 1.61% lower than the DCRNN model and the GCN-LSTM model, and the MAE value is 16.37% and 1.47% lower than the DCRNN model and the GCN-LSTM model.

When the traffic flow prediction window is 60 minutes, the RMSE value of the AST-GCN-LSTM model is decreased by 12.31% and 1.39% compared with the DCRNN model and the GCN-LSTM model. The MAE value is decreased by 17.21% and 1.47% compared with the DCRNN model and the GCN-LSTM model. The above conclusions show the robustness and stability of our proposed model in long-term prediction.

For different prediction times, the AST-GCN-LSTM model proposed in this paper can predict traffic speed well. The results show that this model can capture the changing trend of traffic speed very well, which also verifies the effectiveness of our model in multistep traffic flow prediction.

Comparing the prediction values of the 15-minute and 60-minute prediction windows, we can see that the prediction effect of the 15-minute window in the short-term prediction is closer to the true value, which also shows that the model can better capture short-term dependence.

In order to test the effectiveness of adding static and dynamic external attribute features, we visualized the model prediction results. Figure 7 shows a comparison of prediction results between models with static external attributes, dynamic external attributes, and models without external attributes.

From the visualization results in Figure 7, it can be found that the deviation between the predicted result of AST-GCN-LSTM and the real speed value is smaller than that of AST-GCN-LSTM (static attribute) and AST-GCN-LSTM (dynamic attribute), which indicates that the diversity of external information can better promote prediction.

4. Conclusions

This paper uses the proposed AST-GCN-LSTM model to obtain dynamic attribute features by adding the attribute augmentation unit structure of external factors. After the feature matrix is augmented, the Chebyshev polynomial approximation spectrogram convolution model is used for feature extraction. This model can capture the spatial characteristics of traffic flow from the K-order local neighbors of the vertices in the graph. The K-order local neighborhood can replace the single-hop neighborhood matrix to expand the receptive field of the graph convolution, which can more accurately extract the information of

neighbor nodes. After the information is extracted, the characteristic representation of the information that changes over time is input into the LSTM model to capture the time dependence. By analyzing the performance of the proposed model, including the performance analysis of external attribute characteristics and the performance analysis of different prediction windows, and comparing with different baseline models to verify the effectiveness of the proposed model, it solves the inability of the previous traffic prediction models. The external factors affecting traffic flow are fully considered.

Results show that the AST-GCN-LSTM model can not only fully consider the spatial relationship of road nodes but also capture the time dependence of traffic flow and effectively improve the accuracy of traffic prediction. In addition, the AST-GCN-LSTM model is suitable for both road network traffic flow prediction and midterm and long-term traffic flow prediction and multistep prediction.

Data Availability

The data set in this article can be obtained by contacting the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China 340 (no. 11702289), Key Core Technology and Generic Technology Research and Development Project 341 of Shanxi Province (no. 2020XXX013), and National Key Research and Development Project.

References

[1] J. Li, F. Guo, A. Sivakumar, Y. Dong, and R. Krishnan, "Transferability improvement in short-term traffic prediction using stacked LSTM network," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102977, 2021.

[2] M. Cao, V. Li, and V. Chan, "A CNN-LSTM Model for Traffic Speed Prediction," in *Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, May 2020.

[3] L. Li, Y. Yang, and Z. Yuan, "A spatial-temporal approach for traffic status analysis and prediction based on bi-lstm structure," *Modern Physics Letters B*, vol. 35, no. 31, 2021.

[4] S. Wang, Q. Zhuo, H. Yan, Q. M. Li, and Y. Qi, "A network traffic prediction method based on LSTM," *ZTE Communications*, vol. 17, no. 2, 2019.

[5] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Traffic Time-Series Data by Using Box-Jenkins Techniques*, pp. 1–9, 1979.

[6] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[7] Y. Zhang, Z. Ye, and Y. Xie, "Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 5, pp. 326–334, 2007.

[8] L. L. Ojeda, A. Y. Kibangou, and C. C. D. Wit, "Adaptive Kalman Filtering for Multi-step Ahead Traffic Flow Prediction," in *Proceedings of the 2013 American Control Conference*, pp. 4724–4729, Washington, DC, USA, June 2013.

[9] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, no. 2, pp. 50–64, 2014.

[10] M. Y. Aghdam, S. K. Tabbakh, S. M. Chabok, and M. khairabadi, "Optimization of air traffic management efficiency based on deep learning enriched by the long short-term memory (LSTM) and extreme learning machine (ELM)," *Journal of Big Data*, vol. 8, no. 1, pp. 1–26, 2021.

[11] Y. Li, S. Chai, Z. Ma, and G. Wang, "A hybrid deep learning framework for long-term traffic flow prediction," *IEEE Access*, vol. 9, p. 1, 2021.

[12] T. Li, A. Ni, C. Zhang, and G. Xiao, "Short-term traffic congestion prediction with Conv-BiLSTM considering spatio-temporal features," *IET Intelligent Transport Systems*, vol. 14, no. 1, 2021.

[13] M. Zhang, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 12, no. 6, pp. 1–8, 2013.

[14] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 99, pp. 1–13, 2018.

[15] D. Fan and X. Zhang, "Short-term traffic flow prediction method based on balanced binary tree and K-nearest neighbor nonparametric regression," in *Proceedings of the International Conference on Modelling, Simulation and Applied Mathematics*, Bangkok, Thailand, March 2017.

[16] S. Wu, Z. Yang, X. Zhu, and B. Yu, "Improved k-nn for short-term traffic forecasting using temporal and spatial information," *Journal of Transportation Engineering*, vol. 140, no. 7, Article ID 04014026, 2014.

[17] D. Xu, Y. Wang, P. Peng, S. Beilun, Z. Deng, and H. Guo, "Real-time road traffic state prediction based on kernel-KNN," *Transportmetrica: Transportation Science*, vol. 16, no. 1, pp. 104–118, 2020.

[18] R. V. Leshem, "Traffic flow prediction using adaboost algorithm with random forests as a weak learner," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 19, no. -, pp. 193–198, 2007.

[19] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 58, no. SEP.B, pp. 308–324, 2015.

[20] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *Journal of Sensors*, vol. 2017, Article ID 7074143, 15 pages, 2017.

[21] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[22] Y. Lv, Y. Duan, W. Kang, and Z. Li, "Traffic flow prediction with big data: a deep learning approach," *Intelligent transportation systems*, vol. 16, no. 2, pp. 865–873, 2015.

[23] A. Graves, *Long Short-Term Memory*Springer, Heidelberg, Germany, 2012.

[24] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU Neural Network Methods for Traffic Flow prediction," in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, Hubei Province China, November 2016.

[25] S. Hart, E. J. Hultink, N. Tzokas, and H. R. Commandeur, "Industrial companies' evaluation criteria in new product development gates," *Journal of Product Innovation Management*, vol. 20, no. 1, pp. 22–36, 2003.

[26] Y. Li, R. Yu, C. Shahabi, and L. Yan, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *Proceedings of the Sixth International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[27] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, 2017.

[28] N. Hoang, L. K. Minh, W. Tao, and C. Chen, "Deep learning methods in transportation domain: a review," *Iet Intelligent Transport Systems*, vol. 12, no. 9, 2018.

[29] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, 2017.

[30] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Chennai, India, July 2017.

[31] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, no. 1, 2014.

[32] et al., "Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction," in *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, vol. 60, pp. 893–898, Singapore, November 2018.

[33] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–14, 2019.

[34] Y. Sun, X. Yu, R. Bie, and H. Song, "Discovering time-dependent shortest path on traffic graph for drivers towards green driving," *Journal of Network and Computer Applications*, vol. 83, pp. 204–212, 2016.

[35] H. Sun, J. Wu, D. Ma, and J. Long, "Spatial distribution complexities of traffic congestion and bottlenecks in different network topologies," *Applied Mathematical Modelling*, vol. 38, no. 2, pp. 496–505, 2014.

[36] A. Azari, P. Papapetrou, S. Denic, and G. Peters, "Cellular traffic prediction and classification: a comparative evaluation of lstm and arima," in *Proceedings of the International Conference on Discovery Science*, Split, Croatia, October 2019.

[37] T. Nguyen, "Spatiotemporal tile-based attention-guided lstms for traffic video prediction," 2019, https://arxiv.org/abs/1910.11030.

[38] G. Kalafatas and S. Peeta, *An Exact Graph Structure for Dynamic Traffic Assignment: Formulation, Properties, and Computational Experience*, Transportation Research Board Meeting, Washington, DC USA, 2007.

[39] Z. Ding, *Diversified Ensemble Classifiers for Highly Imbalanced Data Learning and its Application in bioinformatics*, Georgia State University, Atlanta, GA USA, 2011.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9562560, 14 pages
https://doi.org/10.1155/2021/9562560

Research Article

An Eco-Cruise Control for Electric Vehicles Moving on Slope Road with Constant Speed

Ying Zhang[1], **Yingjie Zhang**[2], **Zhaohua Liu**[3], **Jinchao Chen**[1], **Tao You**[1], **and Chenglie Du**[1]

[1]*School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China*
[2]*College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China*
[3]*School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China*

Correspondence should be addressed to Ying Zhang; ying_zhang@nwpu.edu.cn

Received 10 July 2021; Revised 8 September 2021; Accepted 29 September 2021; Published 18 October 2021

Academic Editor: Peng Hang

The unreasonable actuation of electric vehicle's motor drive system usually results in a lot of unwanted energy consumption on a slope road. This paper proposes an eco-cruise control (ECC) scheme based on the driving condition estimation to decrease electric vehicle's energy consumption in the constant-speed cruise control mode. The eco-cruise control scheme is realized by reducing the unreasonable actuation of the motor drive system. The vehicle's total mass and pitch angle are estimated in real time by using an improved base-vector-based cross iteration estimator (BVCIE). Based on the estimated results, the required torque is predicted. Combining the speed deviation between the desired speed and the real speed, and the torque deviation between the required torque and the real torque, a three-power nonlinear controller of the ECC scheme is designed. The ECC scheme is validated on a slope road with different cruise speeds on a cosimulation platform, and the results indicate that the proposed strategy enjoys a better speed maintenance ability and energy efficiency compared with the benchmarked cruise control.

1. Introduction

In the last decades, the lane changing/lane keeping system [1, 2], antilock braking system [3], cruise control system (CCS) [4], advanced vehicle motion control systems [5–7], etc. have been widely implemented on the intelligent vehicles. Cruise control system is one of the first intelligent systems implemented on a vehicle, and it has been vastly applied to different types of ground vehicles. In the constant-speed cruise control mode, the driver can be disengaged from the control tasks of accelerator pedal and brake pedal, which helps to decrease driver's driving load so as to reduce traffic accidents. However, in the constant-speed cruise control mode, when a vehicle is moving on a complex road, such as on a slope road, the driving torque provided by the motor does not often meet the required torque, which is the unreasonable actuation phenomenon. The unreasonable actuation of the motor drive system usually causes the vehicle to consume a lot of unwanted energy. The unwanted energy consumption will undeniably decrease the vehicle's driving range, and this is one of the main factors hindering the spread and popularization of electric vehicles [8]. Therefore, improving energy efficiency is an urgent issue in the field of electric vehicles.

The past research on CCS mainly involves collision avoidance, constant-speed control, improvement of cruise system's fault tolerance, and other intelligent functions.

To address the collision problem, an intelligent cruise system is designed to force the vehicle acceleration to converge to the desired acceleration by using the on-board radar information [9]. In addition, [10, 11] propose advanced algorithms to address the problems of control strategy, scheduling, and real-time constraints simultaneously. Literature [12] initiates the longitudinal stop-and-go cruise control system of heavy-duty trucks, and the test results show that the method not only meets the desired dynamic response, but also enjoys good robustness. In the parameters optimization aspect, [13, 14]

propose different methods to select PID controller's parameters to improve the controller's performance with constant-speed control. The simulation results of the methods in [13, 14] show a good performance in vehicle's stability. To maintain the constant time headway with respect to the front vehicle, a neuro-fuzzy controller is proposed for intelligent cruise control of semiautonomous vehicles, and this method demonstrates better performance compared with the conventional PID controller [15]. In order to improve the fault tolerance of CCS, a prototype tool is designed in [16], and a fault recovery technique is proposed in [17] to improve the fault tolerance and robustness of CCS; these methods demonstrate a good performance in guaranteeing the normal work of CCS. In addition, to enable the driver to fully disengage himself/herself from the driving task, the new research on CCS is advancing in more intelligent directions, such as adaptive cruise control [18, 19] and cooperative adaptive cruise control [20–25].

Obviously, the aforementioned intelligent technologies and methods contribute to the development of the autonomous vehicles. However, most of the research has not considered the energy efficiency problem. In the last decades, energy consumption in the transportation field has sharply increased and has become one of the most serious challenges. In reality, the driving conditions greatly affect the vehicle's energy consumption [26].

Usually, the vehicle moving under traffic signs condition with cruise mode will increase energy consumption level due to the unreasonable start, stop, acceleration, and deceleration. To reduce the idle time at a red light and decrease energy consumption, a predictive cruise control [27] that uses the information of upcoming traffic lights is proposed to judge a predictive manner. By this means, it contributes to the improvement of the energy efficiency of the vehicle's powertrain. According to the traffic information, the energy-efficient cruise control systems [28, 29] are designed to optimize the acceleration profile so as to minimize energy consumption. Similarly, [30] designs an adaptive cruise control system based on the upcoming traffic signal information to reduce the idle time at stop lights and fuel consumption. To decrease the energy consumption due to the queue effects at a signalized intersection, [31] proposes an eco-cooperative adaptive cruise control to improve the vehicle's fuel efficiency, and the simulation result demonstrates that the overall fuel consumption could be saved up to 40%.

The latest research indicates that driving conditions such as the road slope, the vehicle's pitch angle, and the vehicle's mass are seriously affecting the energy consumption and proposes many excellent strategies to improve the energy efficiency [32–36]. However, these methods obtain the road profile from digital maps or from global positioning system (GPS) [32–34], and this will have poor performance in case of shelter conditions. In addition, most of the research mainly focuses on the fuel vehicles [34–36]. Compared with the fuel vehicles, the electric vehicles call for more urgent need to address the energy efficiency problem because of the limits of the driving range.

Therefore, in this paper, we specifically concentrate on the improvement of the electric vehicle's energy efficiency in constant-speed cruise control mode on slope road. Different from the prevailing methods, in this paper, a novel ECC scheme is proposed without using the digital map's information or the GPS information. To guarantee that the electric vehicle has good speed maintenance ability and energy efficiency performance in constant-speed cruise control mode, the ECC scheme is realized by considering the deviation in vehicle speed and the deviation in the required torque. With the ECC scheme, the driving torque and the braking torque are optimized so as to reduce the unreasonable actuation of the motor drive system. In summary, the major innovations and significance of this paper are as follows:

(1) There is methodological innovation in decreasing electric vehicle's energy consumption in the constant-speed cruise control mode on slope road. In this paper, an ECC strategy is proposed to improve the speed maintenance ability and energy efficiency performance. With the ECC strategy, the average driving efficiency and the average regenerative braking efficiency of the motor drive system can be improved.

(2) An improved BVCIE is proposed to estimate the vehicle's driving load factors (total mass and pitch angle). In the framework of the improved BVCIE, a new sliding-model-based strategy is designed to improve the robustness of the estimator. With the sliding-model-based strategy, the phenomenon of signal distortion and data loss can be compensated for.

(3) A new three-power nonlinear controller is designed based on the estimated driving load factors to avoid the unreasonable actuation of the motor's torque caused by the slope road condition. The controller enables the motor drive system to provide reasonable driving torque and braking torque according to the deviations in the vehicle speed and the desired torque. Therefore, the unwanted actuation can be avoided.

The rest of this paper is organized as follows. In Section 2, the vehicle dynamics is analyzed, and the required torque is presented. In Section 3, the vehicle's total mass and the pitch angle are estimated by an improved BVCIE. In Section 4, a novel ECC scheme is proposed. In the ECC scheme, a new controller is designed, and its stability is proved. In Section 5, the simulation platform is introduced, and the estimation results of vehicle's mass and pinch angle are presented. In Section 6, the cruise control performances by the proposed ECC scheme and by the benchmarked cruise control are analyzed in detail. In Section 7, the energy consumption and the energy saving performance of the proposed ECC and the benchmarked cruise control are analyzed at different cruise speeds. Finally, we conclude this paper with its main novelties and contributions in Section 8.

2. Vehicle Dynamics

When an electric vehicle is moving on a slope road in the constant-speed cruise control mode, the required torque in the longitudinal motion is a real-time variable due to the variation of the driving conditions, such as the vehicle's pitch angle and the longitudinal speed. Therefore, to maintain the vehicle at the desired cruise speed, the CCS needs to control the driving torque and the braking torque according to the driving conditions. In vehicle's moving process, the factors affecting the vehicle's performance include the rolling resistance, the equivalent aerodynamic drag resistance, the slope resistance, and the acceleration resistance, as shown in Figure 1.

Assuming the vehicle's total mass is m and the longitudinal acceleration is $\dot{v}$, the longitudinal dynamics model can be expressed as

$$F = F_{\text{rol}} + F_{\text{aero}} + F_{\text{slope}} + m\dot{v}, \tag{1}$$

where F represents a resultant force of the driving force and the braking force, F_{rol} refers to the rolling resistance, F_{aero} refers to the equivalent aerodynamic drag resistance, and F_{slope} refers to the slope resistance. In (1), the rolling resistance depends on the road condition and the vehicle's total mass, which can be expressed as follows:

$$F_{\text{rol}} = mg\mu\cos\theta, \tag{2}$$

where g is the acceleration due to gravity, μ refers to the rolling resistance coefficient, and θ refers to the vehicle's pitch angle.

The slope resistance in (1) is affected by the factors of the vehicle's total mass and the pitch angle, and this relationship can be expressed as follows:

$$F_{\text{slope}} = mg\sin\theta. \tag{3}$$

The equivalent aerodynamic drag resistance in (1) can be expressed as

$$F_{\text{aero}} = \frac{1}{2}AC_d\rho v^2, \tag{4}$$

where A is the frontal area of the vehicle, C_d is the aerodynamic drag coefficient, ρ is the density of air, and v is the vehicle speed. Assuming the wheel's radius is r, the resultant torque T in the longitudinal motion can be expressed as

$$T = Fr. \tag{5}$$

In (5), the resultant torque is from the electric motor (EM). The relation between the resultant torque and the motor's torque can be depicted as

$$T_m = \frac{T}{i_g\eta_g}, \tag{6}$$

where i_g refers to the gear ratio, η_g refers to the efficiency of the mechanical system, and T_m is the motor's torque.

When a vehicle is moving in the constant-speed cruise control mode, the ideal condition is the vehicle moving at the desired cruise speed and the acceleration equal to zero. In addition, in the real condition, the vehicle's total mass is affected by the passenger number and the payload state, and the pitch angle is determined by the real-time road condition. Therefore, the vehicle's total mass is an unknown parameter in the moving process, and the pitch angle is a variable in the moving process. Assuming the total mass can be estimated with $\hat{m}$ in real time and the vehicle's pitch angle can be estimated with $\hat{\theta}$ in real time, the required resultant force at the sampling instance t_k can be expressed as

$$\begin{aligned} F_r(t_k) &= F_{\text{rol}}(t_k) + F_{\text{aero}}(t_k) + F_{\text{slope}}(t_k) \\ &= \hat{m}(t_k)g\mu\cos\left(\hat{\theta}(t_k)\right) + \hat{m}(t_k)g\sin\left(\hat{\theta}(t_k)\right) + 0.5AC_d\rho v_r^2 \\ &= \hat{m}(t_k)g\sqrt{1+\mu^2}\sin\left(\hat{\theta}(t_k) + \theta_\mu\right) + 0.5AC_d\rho v_r^2. \end{aligned} \tag{7}$$

In (7), F_r represents the desired resultant force, v_r represents the desired cruise speed in the constant-speed cruise control mode, and θ_μ can be calculated by $\theta_\mu = \arctan\mu$.

According to (5) and (7), the required torque T_r under the condition that the acceleration is equal to zero can be expressed as

$$T_r(t_k) = F_r(t_k)r. \tag{8}$$

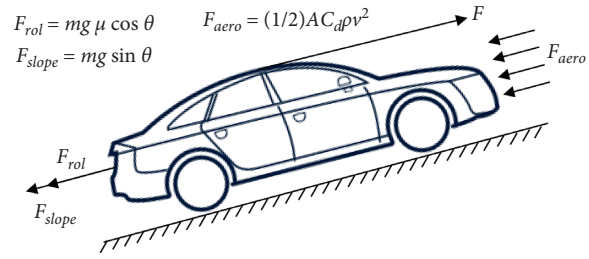


Figure 1: The diagram of vehicle longitudinal dynamics.

3. Estimation of the Total Mass and Pitch Angle

3.1. Estimator Design. The real-time vehicle speed in constant-speed cruise control mode is affected by the driving condition, and consequently when calculating the real-time required torque, it needs to obtain the real-time variables of the vehicle's total mass and the vehicle's pitch angle. To address this problem, in this paper, we adopt a BVCIE because it does not need to reconstruct a state space equation. More importantly, compared with Kalman filter (KF), recursive least square (RLS) method, etc., the BVCIE enjoys a higher estimation accuracy. In addition, compared with the neural network method, the BVCIE does not need to train the model through a large amount of data. To construct the BVCIE, we define $y = \dot{v}$, $\varphi_1 = T/r - 0.5\rho AC\text{d}v^2$, $\varphi_2 = -g(1+\mu^2)^{1/2}$; the vehicle longitudinal dynamics (1) can be rewritten with a linear formula:

$$y = \phi x. \tag{9}$$

In (9), x is the vector which needs be estimated, $x = [x_1, x_2]^T \in \mathbb{R}$, which can be written as follows:

$$\begin{cases} x_1 = \dfrac{1}{m}, \\ x_2 = \sin\left(\theta + \theta_\mu\right). \end{cases} \tag{10}$$

Assuming the vector of x can be estimated with $\hat{x}(k/k-1)$, where k and $k-1$ are the indexes of the discrete sampling instants, we can easily obtain the system's estimated output according to (9), and it can be expressed as follows:

$$\hat{y}(k) = \phi(k)\hat{x}\left(\frac{k}{k-1}\right), \tag{11}$$

where $\hat{x}(k/k-1)$ represents the estimated vector by a cross iteration strategy and $\hat{y}$ is the estimated output. Based on (9) and (11), we can design the estimator with the following formulation:

$$\hat{x}(k) = \hat{x}\left(\frac{k}{k-1}\right) + \phi^{-1}(k)\left(y(k) - \hat{y}(k)\right). \tag{12}$$

According to (9) and (10), the $\hat{x}_1(k/k-1)$ can be formulated by a cross iteration strategy:

$$\hat{x}_1\left(\frac{k}{k-1}\right) = f_1(k) + g_1(k)\hat{x}_2\left(\frac{k}{k-1}\right), \tag{13}$$

where $f_1(k) = \dot{v}(k)/((T(k)/r) - ((1/2)\rho A C_d v^2(k)))$, $g_1(k) = (g\sqrt{1+\mu^2})/((T(k)/r) - ((1/2)\rho A C_d v^2(k)))$.

In vehicle's moving process, the total mass of the vehicle can be deemed as a constant; thus, we can formulate the following equation:

$$\hat{x}_1\left(\frac{k}{k-1}\right) = \hat{x}_1\left(\frac{k-1}{k-2}\right) = \hat{x}_1(k-1). \tag{14}$$

Assuming $\hat{x}_1(k-1)$ can be known in advance, $\hat{x}_2(k/k-1)$ can be formulated using a cross iteration strategy:

$$\begin{aligned} \hat{x}_2\left(\frac{k}{k-1}\right) &= f_2(k) + g_2(k)\hat{x}_1\left(\frac{k}{k-1}\right) \\ &= f_2(k) + g_2(k)\hat{x}_1\left(\frac{k-1}{k-2}\right) \\ &= f_2(k) + g_2(k)\hat{x}_1(k-1), \end{aligned} \tag{15}$$

where $f_2(k) = -(\dot{v}(k)/(g\sqrt{1+\mu^2}))$, $g_2(k) = ((T(k)/r) - ((1/2)\rho A C_d v^2(k)))/(g\sqrt{1+\mu^2})$.

For the proof of its convergence, please refer to [37] for the details. Now, the vehicle's total mass and the vehicle's pitch angle can be estimated by the above estimator. In the above estimator, the inputs of the estimator are the resultant torque, the vehicle speed, and the acceleration. The outputs of the estimator are the estimated vehicle's total mass and the estimated vehicle's pitch angle, as shown in Figure 2.

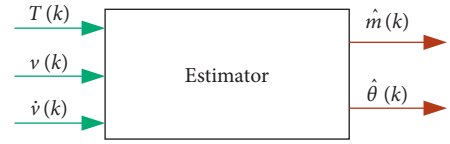


Figure 2: The estimator's diagram for estimating vehicle's total mass and pitch angle.

3.2. Sliding-Mode-Based Strategy. Actually, the input signals of the estimator are bounded by a physical constraint, and this constraint can be described as

$$\begin{cases} T \in [T_{\min}, T_{\max}], \\ v \in [v_{\min}, v_{\max}], \\ \dot{v} \in [a_{\min}, a_{\max}], \end{cases} \tag{16}$$

where *min* and *max* refer to the minimum value and maximum value, respectively. In the vehicle system, the disturbances due to the internal and external factors and the vehicle vibration could result in the input signals' distortion and even the measurement data's loss [38], shown as region 1 in Figures 3(a) and 3(b) and region 2 in Figure 3(c), respectively. These cases will seriously affect the estimation results and even result in the wind-up of the estimator. To fix this problem, a sliding-mode-based strategy for processing the input signals is proposed.

The sliding-mode-based strategy includes a look-backward method and a double-threshold method. When data get lost in an input signal, the look-backward method is put into action to fill in the lost data, as shown in the flow diagrams in Figures 3(e) and 3(g). In the look-backward method, a batch of the input data sequence can be defined as

$$\ell = [\, \ell(t_{k-2}) \;\; \ell(t_{k-1}) \;\; \ell(t_k) \,]. \tag{17}$$

When the input data get lost at the sampling instance t_k, the filling value with the look-backward method can be expressed as

$$\begin{aligned} \ell(t_k) &= \ell(t_{k-1}) + \Delta\ell(t_{k-1}) \\ &= \ell(t_{k-1}) + (\ell(t_{k-1}) - \ell(t_{k-2})) \\ &= 2\ell(t_{k-1}) - \ell(t_{k-2}). \end{aligned} \tag{18}$$

Furthermore, when the input signals are greater than its maximum thresholds or smaller than its minimum thresholds, the double-threshold method is applied to process the input signals, as in the flow diagrams shown in Figures 3(d) and 3(f). The double-threshold method can be expressed as follows:

$$\ell = \begin{cases} \ell_{\max}, & \ell \geq \ell_{\max}, \\ \ell_{\min}, & \ell \leq \ell_{\min}. \end{cases} \tag{19}$$

With the above signal processing strategy, the robustness of the estimator can be enhanced.

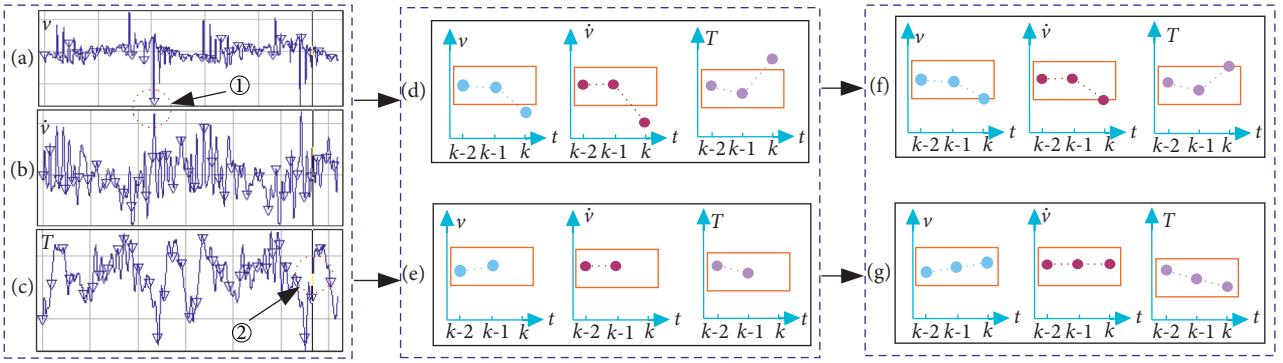


Figure 3: The flow diagram of the sliding-mode-based strategy.

4. Eco-Cruise Control

4.1. Control Scheme. The required torque in (8) in Section 2 represents the torque in the condition that the vehicle maintains the desired constant speed. However, when the vehicle is moving on a slope road, the vehicle speed usually shows fluctuations. In this condition, the CCS needs to regulate the driving torque and the braking torque so as to guarantee that the speed maintains the required value. On the sloping road, the driving load is a variable because the road slope is a variable. Thus, in the vehicle speed regulation process, unreasonable actuation conditions in the driving torque and braking torque may occur due to the slope road condition, which will result in a lot of unwanted energy consumption. For this problem, this section proposes an ECC scheme to reduce the unreasonable actuations of the driving torque and braking torque. In order to clearly describe the ECC scheme, the traditional cruise control scheme and the ECC scheme are shown in Figure 4.

Figure 4(a) shows the traditional cruise control scheme. This scheme only considers the deviation in the speed (e) and does not consider the desired driving torque. For the cruise control scheme in Figure 4(a), a state-of-the-art method is model predictive control (MPC). For the MPC-based cruise control, please refer to [23, 28, 39, 40] for details. Figure 4(b) shows the ECC-based cruise control scheme. As shown in Figure 4(b), the controller has two inputs: one is the deviation in the speed (e), and the other is the deviation in the required torque (ΔT). Hence, the ECC scheme can not only guarantee that the speed maintains the desired value, but also keep the driving torque and the braking torque away from the unreasonable actuation.

4.2. Controller Design and Stability Analysis. In order to design a controller for the ECC scheme, the control system of the electric vehicle in Section 2 can be depicted as

$$\begin{cases} \dot{\mathbf{x}} = \boldsymbol{\alpha}\mathbf{x}^2 + \mathbf{u} - \mathbf{b}, \\ \mathbf{y} = \mathbf{x}, \end{cases} \tag{20}$$

where $\mathbf{x} = v$, $\mathbf{u} = T/(m{\cdot}r)$, $\alpha = -0.5\rho AC_d/m$, $\mathbf{b} = g(\sin\theta + \mu\cos\theta)$. To depict the output deviation of the control system (20), the tracking error in the output ($\mathbf{e}_o$) can be defined as

$$\mathbf{e}_o = \mathbf{y}_r - \mathbf{y}. \tag{21}$$

In addition, for the control system (20), the tracking error in the input ($\mathbf{e}_i$) can be depicted as

$$\mathbf{e}_i = \mathbf{u}_r - \mathbf{u}. \tag{22}$$

In (22), $\mathbf{ur} = T_r/(m{\cdot}r)$, and T_r can be obtained by (8). According to (21) and (22), when considering the errors in the input and output, a three-power nonlinear controller can be designed as in the following formulation:

$$\mathbf{u} = \mathbf{K}_o\mathbf{e}_o^3 + \mathbf{K}_i\mathbf{e}_i, \tag{23}$$

where $\mathbf{K}_o$ refers to the gain of the output $\mathbf{y}$, and $\mathbf{K}_i$ refers to the gain of the required input $\mathbf{u}$.

Theorem 1. *For the control system (20), if there exist $\mathbf{K}_o \in [0, \xi_1]$ and $\mathbf{K}_i \in [0, \xi_2]$, and they satisfy the following condition:*

$$3\mathbf{K}_o\mathbf{e}_o^2\dot{\mathbf{e}}_o + \mathbf{K}_i\dot{\mathbf{e}}_i - \dot{\mathbf{b}} = \ddot{\mathbf{y}} - 2\boldsymbol{\alpha}\dot{\mathbf{x}}\mathbf{y}_r, \tag{24}$$

then the output $\mathbf{y}$ will converge to $\mathbf{y}_r$ by the control strategy (23).

Proof. If the output $\mathbf{y}$ converges to $\mathbf{y}_r$, (21) should satisfy $\mathbf{e}_o \longrightarrow 0$ by the control strategy (23). Equation (20) can be differentiated as

$$\begin{aligned} \ddot{\mathbf{x}} &= 2\boldsymbol{\alpha}\mathbf{x}\dot{\mathbf{x}} + \dot{\mathbf{u}} - \dot{\mathbf{b}} \\ &= 2\boldsymbol{\alpha}\mathbf{x}\dot{\mathbf{x}} + 3\mathbf{K}_o\mathbf{e}_o^2\dot{\mathbf{e}}_o + \mathbf{K}_i\dot{\mathbf{e}}_i - \dot{\mathbf{b}}. \end{aligned} \tag{25}$$

Substituting (25) into (21) yields

$$\mathbf{e}_o = \mathbf{y}_r - \frac{(\ddot{\mathbf{x}} + \dot{\mathbf{b}})}{2\boldsymbol{\alpha}\dot{\mathbf{x}}} - \frac{\left(3\mathbf{K}_o\mathbf{e}_o^2\dot{\mathbf{e}} + \mathbf{K}_i\dot{\mathbf{e}}_i\right)}{2\boldsymbol{\alpha}\dot{\mathbf{x}}}. \tag{26}$$

According to condition (24), $\ddot{\mathbf{y}}$ can be expressed as

$$\ddot{\mathbf{y}} = 3\mathbf{K}_o\mathbf{e}_o^2\dot{\mathbf{e}}_o + \mathbf{K}_i\dot{\mathbf{e}}_i - \dot{\mathbf{b}} + 2\boldsymbol{\alpha}\dot{\mathbf{x}}\mathbf{y}_r. \tag{27}$$

Following the output equation in (20), one can obtain

$$\ddot{\mathbf{y}} = \ddot{\mathbf{x}}. \tag{28}$$

Thus, (26) can be rewritten as

$$\ddot{\mathbf{x}} = 3\mathbf{K}_o\mathbf{e}_o^2\dot{\mathbf{e}} + \mathbf{K}_i\dot{\mathbf{e}}_i - \dot{\mathbf{b}} + 2\boldsymbol{\alpha}\dot{\mathbf{x}}\mathbf{y}_r. \tag{29}$$

Substituting (29) into (26) yields $\mathbf{e}_o = 0$. □

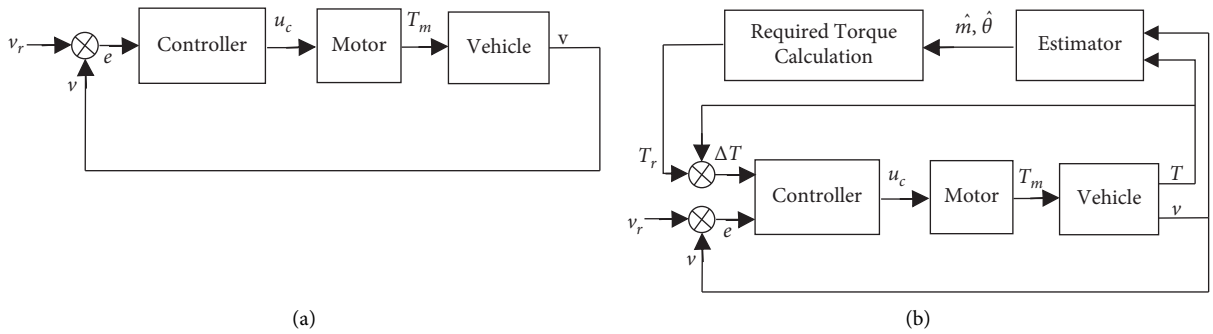


Figure 4: Cruise control scheme. (a) Traditional cruise control scheme. (b) Eco-cruise control scheme.

5. Simulation Environment and Estimation Results

This section first presents the test vehicle and the driving environment; then, the state estimation results and the control performance are presented with detailed analysis.

5.1. The Test Vehicle and the Test Road. In this work, the ECC is validated on a cosimulation platform with CarSim and Matlab/Simulink. The vehicle simulator combining CarSim with Matlab is a high-fidelity platform for validating the vehicle dynamics and control methods. The vehicle's model in CarSim is close to the real-world vehicle. More importantly, the validation through CarSim and Matlab can effectively avoid the influence from the traffic condition and driver's behavior. For the cosimulation platform in this paper, the version of the Matlab is R2007a, and the version of the CarSim is 2016.0. The parameters of the vehicle body and road information are set in CarSim, and the electric powertrain, the proposed ECC, and the benchmarked method are built in Matlab/Simulink. The cosimulation platform is validated on personal computer with Windows 10 64-bit operating system, 16 G memory, i7-10510U CPU, 2.30 GHz.

The simulated vehicle is a front wheel drive electric vehicle, as shown in Figure 5(a). The motor drive system of this vehicle is shown in Figure 5(b). In driving mode, the battery provides power for the EM, and the EM drives the front wheels. The EM is a direct-current motor, and the gearbox has single variable speed ratio. In the driving mode, the battery provides energy to the electric powertrain, and in the braking mode, the regenerative braking energy is fed back to the battery. An asphalt road is selected as the test road, which is shown in Figure 5(c). The profile of the road is from the real world. The length of the test road is about 275 meters, the maximum height of the test road is about 4.173 meters, and the test road includes uphill parts and downhill parts. The profile of the test road is shown in Figure 5(d). The driving environment, the designed parameters, and the specification of the test vehicle are shown in Table 1.

In order to analyze the efficiency of the motor, the efficiency map of the motor should be built. The accurate motor's efficiency map is related to motor's torque and speed. However, as the efficiencies are similar within a relative large range of motor rotational speeds, a simplified efficiency map can be fitted by only considering the motor's torque, and this simplified efficiency map has been successfully used in [41]. For this reason, in this paper, we regard the motor efficiency as a function of torque. The driving efficiency and regenerative braking efficiency of the EM can be fitted by the least square method. Assume that the curve function of the driving efficiency and regenerative braking efficiency can be expressed as

$$y = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x^1 + a_0. \tag{30}$$

To identify the parameters a_0, a_1, a_2, a_3, a_4 based on the least square method, we need to measure several groups' data of the motor's torque and motor efficiency. Since the motor's torque, the motor's rotational speed, and the input/output power of the battery in driving mode and braking mode can be easily collected, the motor efficiency (including the driving efficiency and the regenerative braking efficiency) can be calculated with the following equation: where η_o refers to the driving efficiency, η_i refers to the regenerative braking efficiency, ω is the motor's rotational speed, and P_i/P_o refers to the input/output power of battery.

In order to derive the motor efficiency map, we calculated 11 groups of the driving efficiencies and 11 groups of the regenerative braking efficiencies based on (31). Thus, the curves of the driving efficiency and the regenerative braking efficiency can be fitted. For the fitted driving efficiency curve, $a_0 = 0.1311$, $a_1 = 0.0224$, $a_2 = -2.3647 \times 10^{-4}$, $a_3 = 9.8593 \times 10^{-7}$, $a_4 = -1.4736 \times 10^{-9}$. For the fitted braking efficiency curve, $a_0 = -8.697 \times 10^{-4}$, $a_1 = 0.0125$, $a_2 = -8.685 \times 10^{-5}$, $a_3 = 2.5372 \times 10^{-7}$, $a_4 = -2.6187 \times 10^{-10}$. The calculated driving efficiency, the regenerative braking efficiency, and the fitted curves are shown in Figure 6. The discrete red points are the calculated driving efficiency, and the dashed green curve refers to the fitted driving efficiency. The discrete blue points refer to the calculated regenerative braking efficiency, and the solid magenta curve refers to the fitted braking regenerative efficiency. In the braking mode, part of the mechanical energy will be converted to the thermal energy and get lost. In addition, the efficiency of the power conversion circuit is unlikely to reach 100%. Moreover, during the braking process, part of the regenerative electric energy will get lost. For these reasons, in Figure 6, the regenerative braking efficiency is lower than the driving efficiency under the same torque condition.

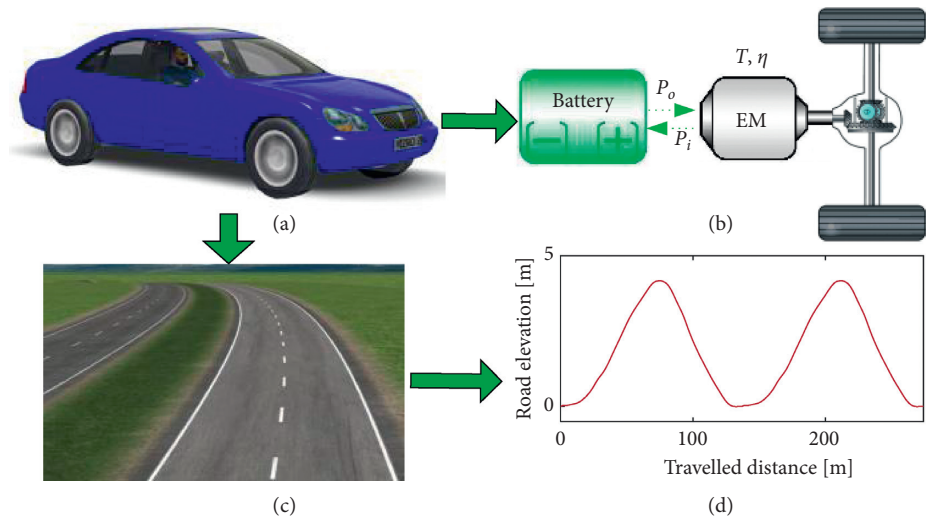


Figure 5: Diagram of the test vehicle and the test road.

Table 1: Parameters and specification of the test vehicle.

Name	Symbol	Values	Unit
Vehicle mass	m	1450	kg
Air density	ρ	1.206	$N{\cdot}s^2/m^4$
Maximum road elevation	h	4.173	m
Vehicle front area	A	2.2	m^2
Wheel radius	r	0.353	m
Roll inertia	I_{xx}	536.6	$kg{\cdot}m^2$
Pitch inertia	I_{yy}	1536.7	$kg{\cdot}m^2$
Yaw inertia	I_{zz}	1536.7	$kg{\cdot}m^2$
Road distance	l	275	m
Battery voltage	V_{bat}	335	V
Battery capacity	E_{bat}	61	kwh
EM's maximum torque	$T_{m\text{-max}}$	320	Nm
EM's maximum power	$P_{m\text{-max}}$	160	kw
Gear ratio	i_g	7.6	—
Drag coefficient	C_d	0.28	—
Rolling friction coefficient	μ	0.012	—

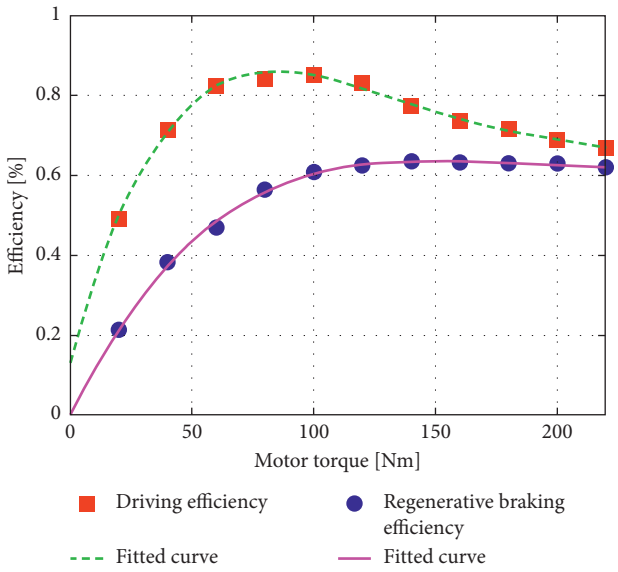


Figure 6: Driving efficiency and regenerative braking efficiency of EM.

$$\eta_o = \frac{T_m \omega}{P_o},$$
$$\eta_i = \frac{P_i}{T_m \omega}, \tag{31}$$

To analyze the proposed ECC scheme, the state-of-the-art method of MPC-based cruise control is selected as a benchmark. The scheme of the benchmarked method is introduced in Figure 4(a) in Section 4.1. For the MPC in Figure 4(a), assuming P is the predictive horizon, M is the control horizon, and $M \le P$, the predictive output sequence is $\widehat{Y}(k+1/k)$ and the future control sequence $\Delta U(k)$ at sample time k can be formulated as follows:

$$\begin{cases} \widehat{Y}\left(\dfrac{k+1}{k}\right) = \left[\, \widehat{y}(k+1) \quad \widehat{y}(k+2) \quad \cdots \quad \widehat{y}(k+P) \,\right]^T_{1\times P}, \\ \Delta U(k) = \left[\, \Delta u(k) \quad \Delta u(k+1) \quad \cdots \quad \Delta u(k+m-1) \,\right]^T_{1\times M}. \end{cases} \tag{32}$$

In addition, the cost function J can be constructed as

$$J = \sum_{j=1}^{P} [\widehat{y}(k+j) - y(k+j)]^2 + \sum_{j=1}^{M} \lambda(j)[\Delta u(k+j-1)]^2, \tag{33}$$

where λ is the weighting coefficient. The controller of MPC is used to make the cost function minimal. To obtain the optimum control sequence, we can make the derivative of the cost function equal to zero:

$$\frac{\partial J}{\partial \Delta U} = 0. \tag{34}$$

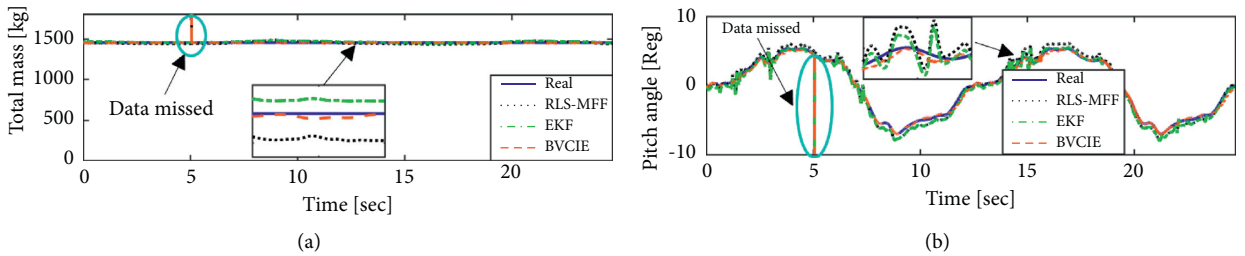
Then, the solution of $\Delta U(k)$ can be obtained. Moreover, by referring to the method in [23] and using the trial-and-error method, the prediction horizon length M and the control horizon length P are specified to be 10. The benchmarked cruise control and the proposed ECC are validated on the cosimulation platform of CarSim and Matlab/Simulink. With the cosimulation environment, the ECC and the benchmarked method are tested with different cruise speeds.

5.2. Results of Vehicle State Estimation. To analyze the performance of the improved BVCIE in Section 3, the test on the vehicle moving at 40 km/h is presented. In addition, the recursive least square method with multiple forgetting factors (RLS-MFF), the extended Kalman filter (EKF), and the original method, i.e., BVCIE, are adopted for comparison.

For the original BVCIE, RLS-MFF, and EKF and the improved BVCIE, the initial values of the estimated vehicle total mass and pinch angle are set to 0. The estimated total mass and the estimated pitch angle by the RLS-MFF, the EKF, the original BVCIE, and the improved BVCIE are shown in Figures 7 and 8, respectively. In Figure 7, the blue curves represent the real value, the dotted black curves represent the estimated value by RLS-MFF, the dash-dotted green curves represent the estimated value by EKF, and the dashed orange curves represent the estimated value by the original BVCIE. From the results in Figure 7, especially the zoom-in curves in Figure 7, it can be found that the original BVCIE outperforms RLS-MFF and EKF in estimation accuracy. However, in case of the input data loss, the estimation results by RLS-MFF, EKF, and original BVCIE have a sharp wind-up. Figure 8 depicts the estimated results by the improved BVCIE. The blue curves represent the real value, and the dashed magenta curves represent the estimated value. From the results in Figure 8, we can find that the estimated values have good performances and can accurately reflect the real values. In addition, the wind-up phenomenon can be eliminated when the input data are lost. From the local zoom-in curves in Figures 8(a) and 8(b), we know that there exists a little error between the estimated value and the real value. To analyze the deviation, the estimation errors in total mass and in pitch angle by the improved BVCIE are shown in Figures 9(a) and 9(b). Compared with the estimated values in Figures 8(a) and 8(b), the estimation errors are very small. In this condition, the vehicle's state estimated by the method in Section 3 can be applied to the proposed ECC scheme in Section 4.

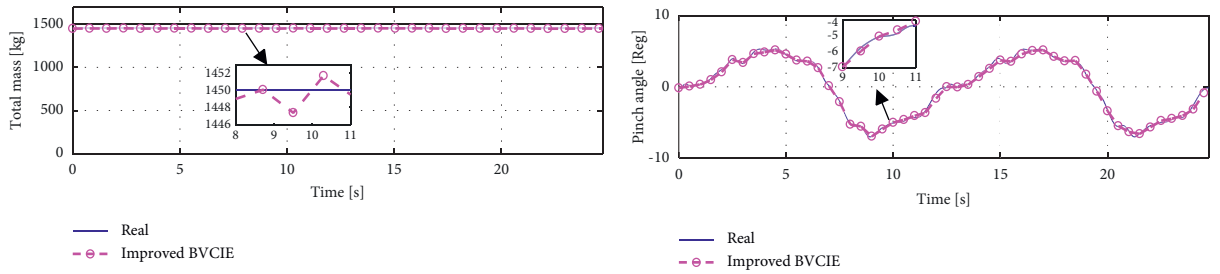
6. Control Performances of ECC Scheme

6.1. Control Performance at 40 km/h. To analyze the control performances of the proposed ECC and the benchmarked cruise control, the vehicle speed, the motor's torque (including the driving torque and the braking torque), and the motor's efficiency are analyzed in detail at 40 km/h when moving on the test road. The results are shown in Figure 10. The blue curve and the red curve in Figure 10(a), respectively, reflect the vehicle speed by the benchmarked cruise control and ECC. The vehicle speed has a little perturbation when moving on the test road. On uphill part, the vehicle speed decreases less by ECC than by the benchmarked cruise control, and on downhill part, the vehicle speed increases less by ECC than by the benchmarked cruise control. Therefore, it can be obviously observed that ECC is better than the benchmarked cruise control in speed maintenance ability. From Figure 10(b), one can know that the motor's torques vary by different methods. Combining the speed profiles in Figure 10(a), we can find that the cruise speed by ECC has less fluctuation compared with that by the benchmarked cruise control, and most of the working points of the motor's torque by ECC are in the small value interval. Therefore, the motor's torque by ECC has less unreasonable actuation condition compared with that by the benchmarked cruise control. Figure 10(c) demonstrates the driving efficiency and the regenerative braking efficiency. According to the results in Figure 10(c), we can observe that at some sampling instances the motor's efficiency by ECC is greater than that by the benchmarked cruise control, but at other sampling instances, the motor's efficiency by ECC is lower than that



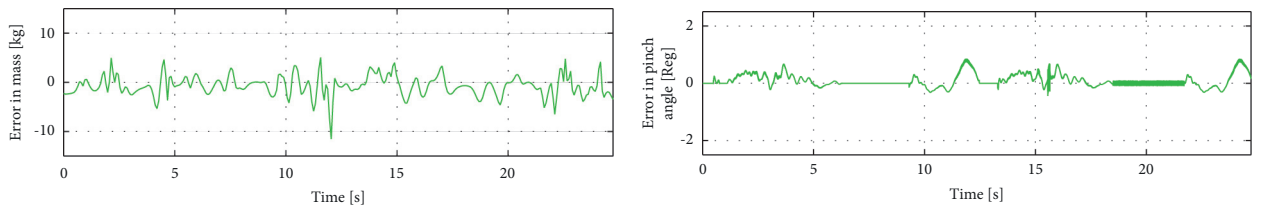
(a) (b)

Figure 7: Estimation results by RLS-MFF, EKF, and BVCIE at 40 km/h on asphalt road. (a) Estimated total mass. (b) Estimated pitch angle.



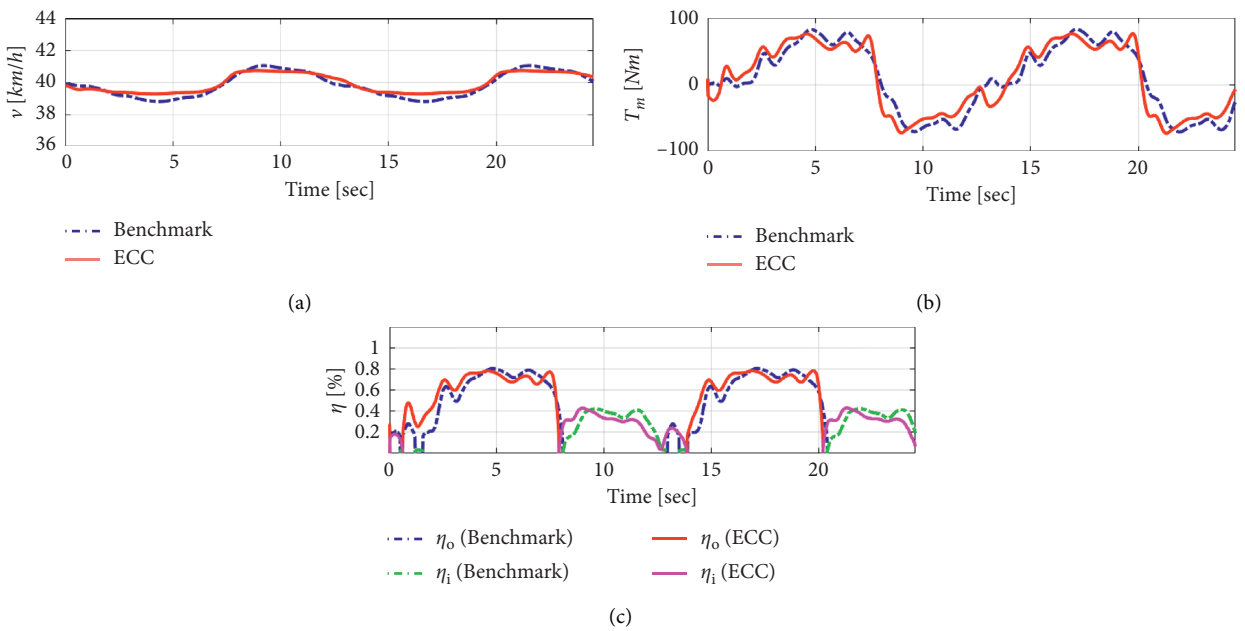
(a) (b)

Figure 8: Estimation results by the improved BVCIE at 40 km/h on asphalt road. (a) Estimated total mass. (b) Estimated pitch angle.



(a) (b)

Figure 9: Estimation errors by the improved BVCIE at 40 km/h on asphalt road. (a) Estimation error in mass. (b) Estimation error in pitch angle.



(a) (b) (c)

Figure 10: Control performances and comparison of ECC and benchmarked cruise control. (a) Vehicle speed. (b) Motor's torque. (c) Motor's efficiency.

by the benchmarked cruise control. The reason for this phenomenon is that the motor drive system works in unreasonable actuation conditions at some sampling instances by the benchmarked cruise control. When the driving torque provided by the motor is smaller than the torque demands, the energy consumption by the benchmarked cruise control is less than that by ECC, but when the driving torque provided by the motor is greater than the torque demands, the energy consumption by the benchmarked cruise control is greater than that by ECC. In this condition, it is difficult to judge which method is better according to Figure 10(c). By referring to the analysis method in [41], the average driving efficiency and the average regenerative braking efficiency are introduced to compare the ECC and the benchmarked cruise control.

In addition, by referring to the analysis method in [42], the ride comfort can be assessed by vehicle's jerk (J), which is the second derivative of the vehicle's longitudinal velocity. In the moving process, the jerk values include positive values and negative values. For this reason, we adopt the average absolute jerk to compare the ride comfort of ECC and the benchmarked cruise control. The smaller the average absolute jerk, the better the ride comfort. The average absolute jerk is defined as

$$|\bar{J}| = \frac{1}{N}\sum_{k=1}^{N} |\ddot{v}(k)|. \tag{35}$$

The average driving efficiency, the average regenerative braking efficiency, and the average absolute jerk are shown in Table 2.

According to the results in Table 2, in the driving mode, the average driving efficiency by the benchmarked cruise control is significantly lower than that by ECC. In the braking mode, the average driving efficiency by ECC is similar to that by the benchmarked cruise control. Thus, ECC reasonably adjusts the motor's toque so that the motor works in the high efficiency range. In addition, the average absolute jerk by ECC is smaller than that by the benchmarked cruise control. From the results in Table 2, we can conclude that the motor's efficiency and the ride comfort by ECC outperform those by the benchmarked cruise control in the driving mode and braking mode.

6.2. Control Performance at 80 km/h. To demonstrate the performance of the proposed ECC at high cruise speed, the cruise speed of 80 km/h is investigated in this part. The vehicle speed, the motor's torque, and the motor's efficiency are presented in Figure 11. In Figure 11(a), it is obvious that the speed maintenance ability by ECC is still more excellent than that by the benchmarked cruise control. Figures 11(b) and 11(c), respectively, demonstrate the motor's torque and the motor's efficiency. Similar to the results in Figures 11(b) and 11(c), it is still difficult to distinguish which method is better. That is because sometimes the motor's torque and the motor's efficiency by ECC are greater than those by the benchmarked cruise control, but sometimes the motor's torque and the motor's efficiency by ECC are lower than those by the benchmarked cruise control. Therefore, we also calculate the average driving efficiency, the average regenerative braking efficiency, and the average absolute jerk, and they are shown in Table 3.

Table 2: Motor's average efficiency and ride comfort by the benchmarked cruise control and the proposed ECC (40 km/h).

Control performances	Benchmark	ECC
Driving efficiency	0.5827	**0.6468 × 100%**
Regenerative braking efficiency	0.2790	**0.2900 × 100%**
Average absolute jerk	1.3964	**1.1930 km/s^2**

From Table 3, it can be found that in the driving mode, the driving efficiency by ECC is higher than that by the benchmarked cruise control. More importantly, the regenerative braking efficiency by ECC is also higher than that by the benchmarked cruise control. In addition, the average absolute jerk by ECC is still smaller than that by the benchmarked cruise control. Therefore, it can be concluded that ECC is better than the benchmarked cruise control at high cruise speed.

7. Energy Saving Performance Analysis

To demonstrate vehicle's energy saving performance of the proposed ECC, the energy consumption of the two scenarios in Sections 6.1 and 6.2 is analyzed in detail. The energy consumption in the driving mode and the energy feedback in the braking mode are calculated as follows:

$$\begin{aligned} E_o &= \int_0^{t_f} P_o(t)\mathrm{d}t, \\ E_i &= \int_0^{t_f} P(t)\mathrm{d}t_i, \end{aligned} \tag{36}$$

where E_o refers to the energy consumption in the driving mode, E_i refers to the energy feedback in the braking mode, and t_f refers to the termination time.

The energy consumption and energy feedback of the two scenarios in Sections 6.1 and 6.2 are shown in Figures 12(a) and 12(b), respectively. When the test vehicle is moving at 40 km/h, the energy consumption by the ECC and that by the benchmarked cruise control are 0.2033 kwh and 0.1956 kwh, respectively. The energy feedback by ECC and the benchmarked cruise control is 0.0424 kwh and 0.0496 kwh, respectively. Obviously, it can be found that the vehicle consumes less energy by ECC when moving at 40 km/h compared with that by the benchmarked cruise control. For the high cruise speed of 80 km/h, the energy consumption by the ECC and by the benchmarked cruise control is 0.1709 kwh and 0.1583 kwh, respectively. In addition, the energy feedback by the ECC and by the benchmarked cruise control is 0.0472 kwh and 0.0483 kwh. The results in Figure 12(b) show the advantage of ECC because the vehicle not only consumes less energy in the driving mode, but also regenerates more energy in the braking mode.

For the two scenarios, the real energy consumption (E) in the whole process can be calculated as

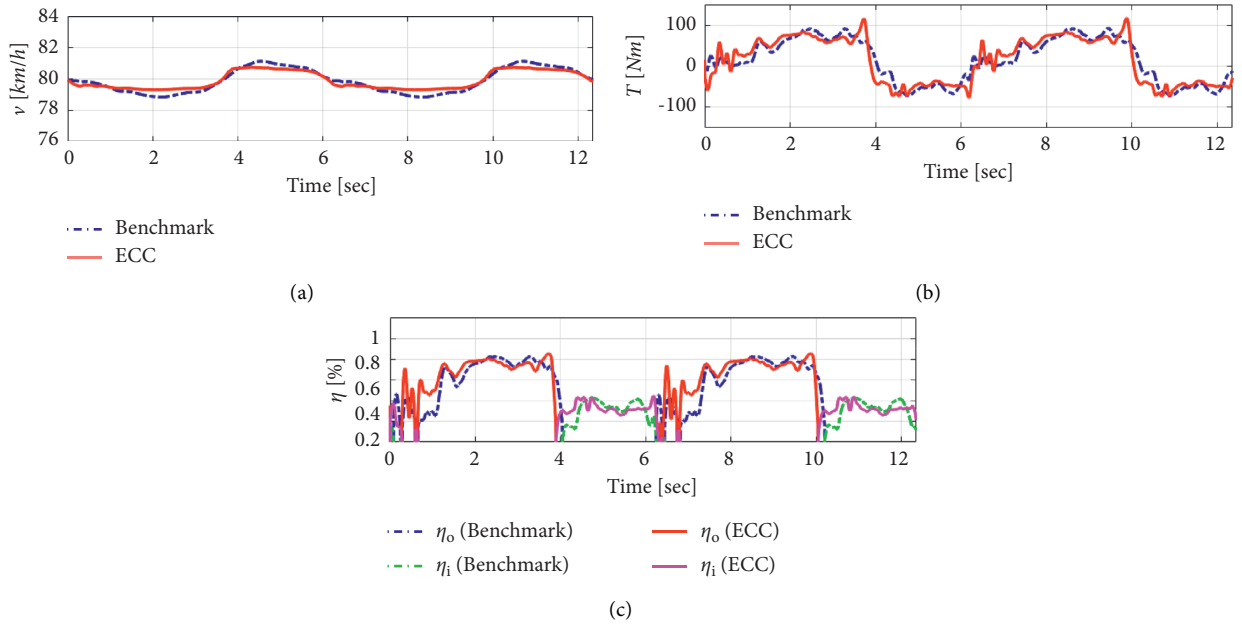


Figure 11: Control performances and comparison of ECC and benchmarked cruise control. (a) Vehicle speed. (b) Motor's torque. (c) Motor's efficiency.

Table 3: Motor's average efficiency and ride comfort by the benchmarked cruise control and the proposed ECC (80 km/h).

Control performances	Benchmark	ECC
Driving efficiency	0.6103	**0.6775 × 100%**
Regenerative braking efficiency	0.2973	**0.3057 × 100%**
Average absolute jerk	1.4265	**1.2646 km/s^2**

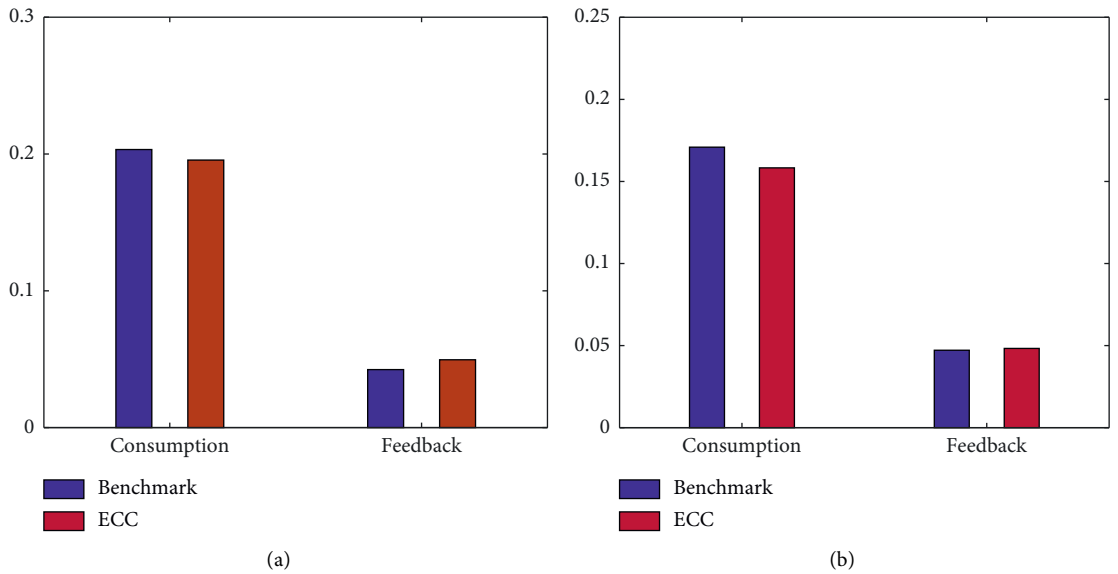


Figure 12: Results of energy consumption and feedback (kwh). (a) At 40 km/h. (b) At 80 km/h.

$$E = E_o - E_i. \tag{37}$$

To analyze the real energy consumption on the test road in detail, the cruise speeds of 40 km/h, 50 km/h, 60 km/h, 70 km/h, 80 km/h, 90 km/h, 100 km/h, and 110 km/h are selected to investigate ECC performance, and the proposed ECC is detailedly compared with the benchmarked cruise control. The total energy consumption on the test road at 40 km/h, 50 km/h, 60 km/h, 70 km/h, 80 km/h, 90 km/h, 100 km/h, and 110 km/h is shown in Figure 13. It is obvious that the test vehicle

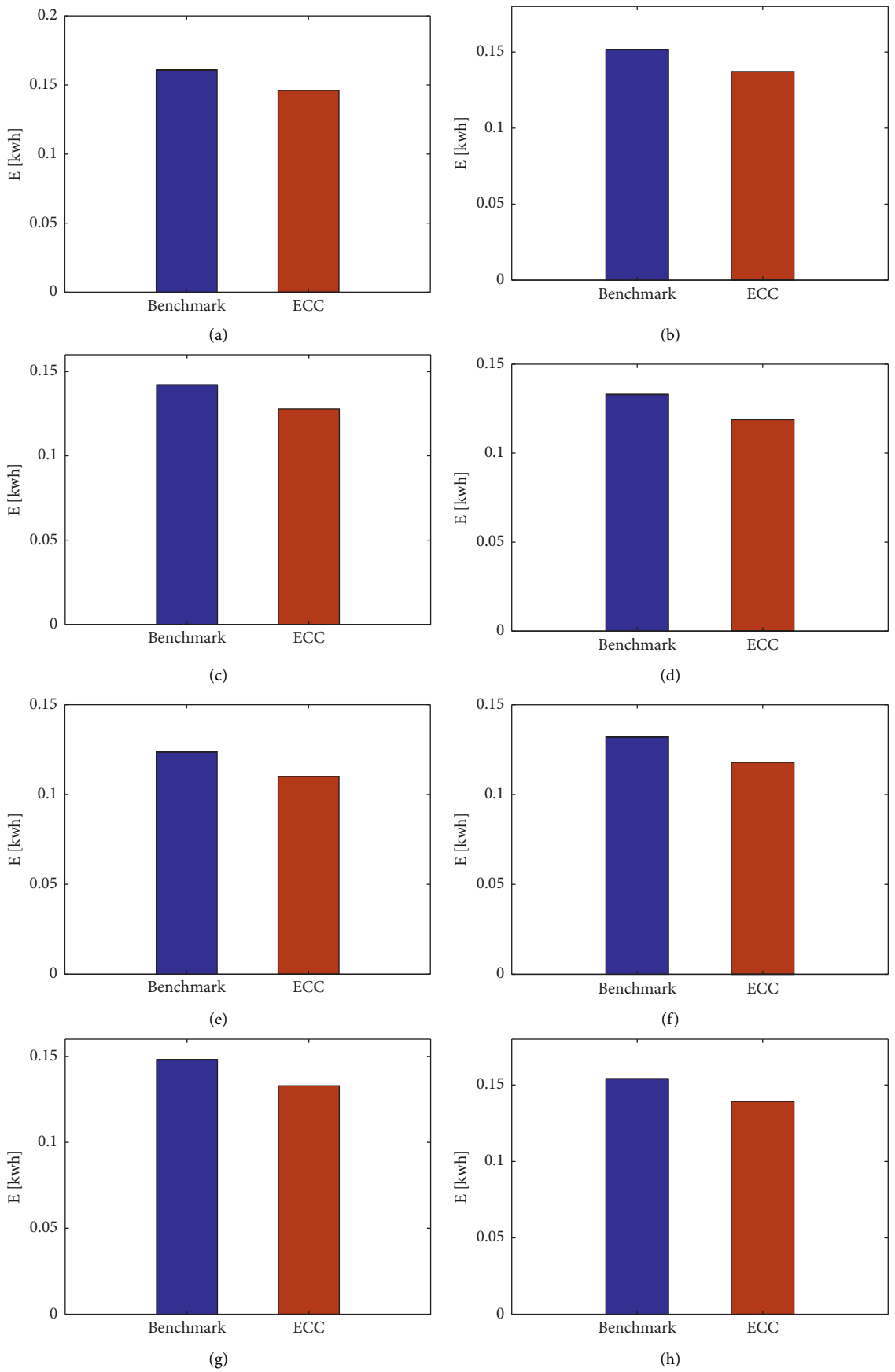


Figure 13: Energy consumption at different cruise speeds. (a) At 40 km/h. (b) At 50 km/h. (c) At 60 km/h. (d) At 70 km/h. (e) At 80 km/h. (f) At 90 km/h. (g) At 100 km/h. (h) At 110 km/h.

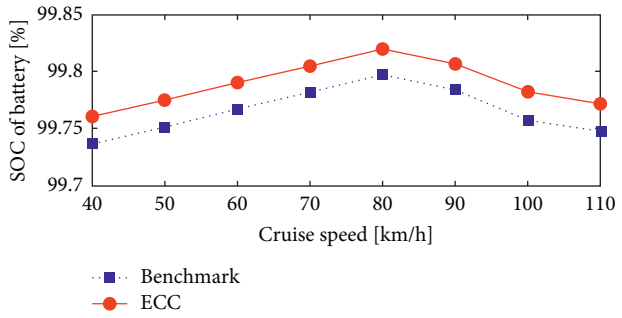


Figure 14: Battery SOC at different cruise speeds.

consumes less energy in all cruise speeds by ECC compared with that by the benchmarked cruise control. The results in Figure 13 effectively prove that the proposed ECC has better performances in decreasing the energy consumption.

Furthermore, to compare the energy saving performance of the proposed ECC and the benchmarked cruise control, the state of charge (SOC) of the battery is adopted. Before implementing the simulation, the battery SOC is 100%. After the simulation, the battery SOC at 40 km/h, 50 km/h, 60 km/h, 70 km/h, 80 km/h, 90 km/h, 100 km/h, and 110 km/h is shown in Figure 14. The dotted blue line refers to the battery SOC by the benchmarked cruise control, and the solid red line refers to the battery SOC by the proposed ECC. It can be easily found that the battery SOC by the proposed method is higher than that by the benchmarked cruise control. Therefore, the proposed ECC outperforms the benchmarked cruise control in energy saving performance.

8. Conclusion

Motivated by the driving range limits of electric vehicles, this paper proposes an ECC to improve the energy utilization efficiency of electric vehicles. Different from the prevailing research, this paper focuses on the condition that the electric vehicle is moving on the slope road without using the digital map's information and the GPS information. Additionally, energy saving is realized by reducing the unreasonable actuation of the motor's torque. Compared with the benchmarked cruise control, the proposed ECC not only improves the electric vehicle's speed maintenance ability and ride comfort, but also improves the energy efficiency. Moreover, different cruise speed tests indicate that the proposed ECC can obviously decrease the energy consumption, and the statistical results of the battery SOC show that the proposed ECC can improve the energy saving performance. The research in this paper provides a novel method to improve electric vehicles' energy efficiency.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFB2101304), the Fundamental Research Funds for the Central Universities (3102020QD1004), the Natural Science Foundation of Shaanxi Province (2021JQ-121), the National Natural Science Foundation of China (51674113 and 62106202), and the Joint Aeronautical Science Foundation Project (202051053002).

References

[1] Z. Q. Liu, T. Zhang, and Y. F. Wang, "Research on local dynamic path planning method for intelligent vehicle lane-changing," *Journal of Advanced Transportation*, vol. 43, no. 1, pp. 47–70, 2019.

[2] F. Farivar, M. S. Haghighi, A. Jolfaei et al., "Covert attacks through adversarial learning: studying the effect of lane keeping attacks on the safety of autonomous vehicles," *IEEE*, vol. 26, no. 3, pp. 1350–1357, 2021.

[3] S. Rafatnia and M. Mirzaei, "Adaptive estimation of vehicle velocity from updated dynamic model for control of anti-lock braking system," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[4] Z. Liu, Q. Yuan, G. Nie et al., "A multi-objective model predictive control for vehicle adaptive cruise control system based on a new safe distance model," *International Journal of Automotive Technology*, vol. 22, no. 2, pp. 475–487, 2021.

[5] L. Zhang, Y. Wang, and Z. Wang, "Robust lateral motion control for in-wheel-motor-drive electric vehicles with network induced delays," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10585–10593, 2019.

[6] L. Zhang, Z. Wang, X. Ding et al., "Fault-tolerant control for intelligent electrified vehicles against front wheel steering angle sensor faults during trajectory tracking," *IEEE Access*, vol. 9, pp. 65174–65186, 2021.

[7] X. Ding, Z. Wang, and L. Zhang, "Hybrid control-based acceleration slip regulation for four-wheel-independently-actuated electric vehicles," *IEEE Transactions on Transportation Electrification*, vol. 7, no. 3, pp. 1976–1989, 2020.

[8] Y. Zhang, Y. Zhang, Z. Ai et al., "Energy optimal control of motor drive system for extending ranges of electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1728–1738, 2021.

[9] G. Q. Song, Y. Yang, H. Hang et al., "Study on vehicle collision-avoiding radar and intelligent cruise control system," *Key Engineering Materials*, vol. 297–300, pp. 311–315, 2005.

[10] A. Swetha, N. R. A. Asmy, V. R. Pillay et al., "Algorithm for autonomous cruise control system - real time scheduling and control aspects," *Advances in Intelligent Systems & Computing*, vol. 327, pp. 449–456, 2014.

[11] X. W. Chen, J. G. Zhang, and Y. J. Liu, "Research on the intelligent control and simulation of automobile cruise system based on fuzzy system," *Mathematical Problems in Engineering*, vol. 2016, Article ID 9760653, 12 pages, 2016.

[12] K. Li, UkawaHiroshi, and HandaMasatoshi, "Nonlinear disturbance decoupling control of heavy-duty truck stop and go cruise system," *Vehicle System Dynamics*, vol. 47, no. 1, pp. 29–55, 2009.

[13] N. Vedam, I. Diaz-Rodrigue, and S. P. Bhattacharyya, "A novel approach to the design of controllers in an automotive cruise-control system," in *Proceedings of the IECON 2014—40th Annual Conference of the IEEE Industrial*

Electronics Society, pp. 2927–2932, Dallas, TX, USA, February 2015.

[14] M. Laheeb, K. Salwan, A. B. Fawaz et al., "Online algorithm for controlling a cruise system under uncertainty in design parameters and environmental conditions using Monte-Carlo simulation," in *Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 424–430, Las Vegas, NV, USA, February 2018.

[15] L. Cai, A. B. Rad, and W. L. Chan, "An intelligent longitudinal controller for application in semiautonomous vehicles," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1487–1497, 2010.

[16] A. Swetha, S. Dasgupta, S. K. Chandran et al., "A real-time performance monitoring tool for dual redundant and resource augmented framework of cruise control system," *International Journal of Computer Applications in Technology*, vol. 92, no. 14, pp. 44–49, 2014.

[17] A. Swetha, P. V. Radhamani, and S. Punnekkat, "Design, analysis and implementation of improved adaptive fault tolerant model for cruise control multiprocessor system," *International Journal of Computer Applications in Technology*, vol. 86, no. 15, pp. 29–26, 2014.

[18] P. Nilsson, O. Hussien, A. Balkan et al., "Correct-by-construction adaptive cruise control:two approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.

[19] S. H. Hosseinnia, I. Tejado, V. Milanés et al., "Experimental application of hybrid fractional-order adaptive cruise control at low speed," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2329–2336, 2014.

[20] K. C. Dey, L. Yan, X. Wang et al., "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491–509, 2016.

[21] F. Li and Y. Wang, "Cooperative adaptive cruise control for string stable mixed traffic: benchmark and human-centered design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3473–3485, 2017.

[22] S. Öncü, J. Ploeg, N. Van de Wouw et al., "Cooperative adaptive cruise control: network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, 2014.

[23] C. M. Filho, M. H. Terra, and D. F. Wolf, "Safe optimization of highway traffic with robust model predictive control-based cooperative adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3193–3203, 2017.

[24] B. V. Arem, C. J. G. V. Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.

[25] B. Liu and A. E. Kamel, "V2X-based decentralized cooperative adaptive cruise control in the vicinity of intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 644–658, 2016.

[26] Y. Zhang, Y. Zhang, Z. Ai et al., "Energy saving control strategy for the high-frequency start-up process for electric mining haul trucks," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 595–606, 2018.

[27] B. Alrifaee, J. G. Jodar, and D. Abel, "Decentralized predictive cruise control for energy saving in REEV using V2I information for multiple-vehicles," *IFAC Papersonline*, vol. 48, no. 15, pp. 320–327, 2015.

[28] B. Alrifaee, Y. Liu, and D. Abel, "Eco-cruise control using economic model predictive control," in *Proceedings of the IEEE Conference on Industrial Automation and Control Emerging Technology Applications*, pp. 1933–1938, Sydney, Australia, September 2015.

[29] B. Alrifaee, J. G. Jodar, and D. Abel, "Predictive cruise control for energy saving in REEV using V2I information," in *Proceedings of the 2015 23rd Mediterranean Conference on Control and Automation (MED)*, pp. 82–87, Torremolinos, Spain, July 2015.

[30] B. Asadi and A. Vahidi, "Predictive cruise control: utilizing upcoming traffic signal information for improving fuel economy and reducing trip time," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 707–714, 2011.

[31] H. Yang, H. Rakha, and M. V. Ala, "Eco-cooperative adaptive cruise control at signalized intersections considering queue effects," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1575–1585, 2017.

[32] E. Talavera, A. Díaz-Álvarez, F. Jiménez et al., "Impact on congestion and fuel consumption of a cooperative adaptive cruise control system with lane-level position estimation," *Energies*, vol. 11, no. 1, p. 194, 2018.

[33] Y. Chen, X. Li, C. Wiet et al., "Energy management and driving strategy for in-wheel motor electric ground vehicles with terrain profile preview," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1938–1947, 2014.

[34] M. A. S. Kamal et al., "Ecological vehicle control on roads with up-down slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, 2011.

[35] R. T. Bambang, A. S. Rohman, C. J. Dronkers et al., "Energy management of fuel cell/battery/supercapacitor hybrid power sources using model predictive control," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 1992–2002, 2014.

[36] D. Tian, Y. Zhu, X. Duan et al., "An effective fuel level data cleaning and repairing method for vehicle monitor platform," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 410–422, 2019.

[37] Y. Zhang, Y. Zhang, Z. Ai et al., "A cross iteration estimator with base vector for estimation of electric mining haul truck's mass and road grade," *IEEE Transactions on Industrial Informatics*, vol. 44, no. 9, pp. 4138–4148, 2018.

[38] Y. Zhang, Y. Zhang, Z. Ai et al., "Study on electric sunroof pinch detection of cars based on model reference adaptive cholesky decomposition filter," *IEEE Transactions on Transportation Electrification*, vol. 4, no. 1, pp. 190–201, 2018.

[39] A. Weißmann, D. Görges, and X. Lin, "Energy-optimal adaptive cruise control combining model predictive control and dynamic programming," *Control Engineering Practice*, vol. 72, pp. 125–137, 2018.

[40] Sajadi-Alamdari, S. Amin, H. Voos, and M. Darouach, "Nonlinear model predictive extended eco-cruise control for battery electric vehicles," in *Proceedings of the 24th Conference on Control and Automation*, pp. 467–472, Athens, Greece, June 2016.

[41] Y. Chen and J. Wang, "Design and experimental evaluations on energy efficient control allocation methods for over-actuated electric vehicles: longitudinal motion case," *IEEE*, vol. 19, no. 2, pp. 538–548, 2014.

[42] C. Lv, X. Hu, A. Sangiovanni-Vincentelli et al., "Driving-style-based co-design optimization of an automated electric vehicle: a cyber-physical system approach," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 2965–2975, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9297218, 14 pages
https://doi.org/10.1155/2021/9297218

WILEY | Hindawi

Research Article

Experimental Study on Different Types of Curves for Ride Comfort in Automated Vehicles

Naohisa Hashimoto,[1] **Yusuke Takinami**,[1] **and Makoto Yamamoto**[2]

[1]*Human Centered Mobility Research Center, National Institute of Advanced Industrial Science and Technology, Umezono, Tsukuba 305-8568, Japan*
[2]*Sanei Giken Cooperation, 3-26 Kaminoboricho, Naka-ku, Hiroshima-shi, Hiroshima-ken 730-0014, Japan*

Correspondence should be addressed to Naohisa Hashimoto; naohisa-hashimoto@aist.go.jp

Received 24 May 2021; Revised 14 September 2021; Accepted 20 September 2021; Published 1 October 2021

Academic Editor: Peng Hang

Vehicle automation is among the best possible solutions for traffic issues, including traffic accidents, traffic jams, and energy consumption. However, the user acceptance of automated vehicles is critical and is affected by riding comfort. In addition, human factors in automated vehicle control should be clear. This study evaluates the effect of different courses on driving comfort in automated vehicles using field experiments with 25 subjects. This study focused on lateral motion, but speed control was not targeted. Further, generating a path for obstacle avoidance and lane keeping, which have several constraining conditions, was also not targeted. Rendering a comfortable path is beneficial for developing an acceptable system as a car developer and for building new curves for automated or driving assistance systems from the perspective of construction. The automated vehicle drove at a speed of 30 km/h on four courses, namely, clothoid, two types of spline curves, and arc, based on the real intersection. Each participant sat on both the driver and passenger seat and answered a questionnaire. The experimental data indicated the clothoid course to be the most comfortable, while the arc was most uncomfortable for a significance level of 1%. These tendencies are applicable to driver and passenger seats, all genders, and experiences and will be beneficial for human factor research in automated vehicle control.

1. Introduction

An increase in urban traffic has led to increased traffic jams, traffic accidents, and air pollution, resulting in serious damage. Automated vehicles are expected to be a solution worldwide [1–3]. Automated vehicles can change the entire traffic system by making it more efficient and can contribute to the safety and reduction of CO_2 emissions [4–6]. Furthermore, the introduction of automated vehicles could provide new services, such as autonomous riding shuttles, autonomous delivery, and autonomous valet parking. Many types of technologies for the introduction of completely automated vehicles have been developed, including position estimation, vehicle control, environmental recognition, public acceptance, human factors, infrastructure for automated vehicles, and laws [7–11].

The present study only considered lateral motion while speed control, generating a path for obstacle avoidance, and lane keeping, which have several constraining conditions, were not targeted. Safety is among the most important factors in the introduction of automated vehicles. Our automated vehicle employs the function of obstacle detection and avoidance, and when an obstacle is detected on the desired course, the priority of the control is to ensure safety to avoid the obstacle. In the experiments, there were no obstacles to the desired course. Thus, the obstacle avoidance function was inactive. In this study, we focused on comfort while ensuring safety. For the path for automated vehicle control, the clothoid curve has been extensively used according to several studies and the result of minimizing the maximum acceleration. Consequently, our hypothesis was that the clothoid curve was the most comfortable. However,

research on automated driving for driving comfort is scarce. Further, the studies related to the human factor have mostly been evaluated using a simulator owing to the ease of setting up experimental conditions. However, evaluating our hypothesis in a simulation study was difficult, and thus, we decided to perform the experiments using a real vehicle to prove the hypothesis with persuasive results.

The present study evaluates the effect of several methods of constructing curves with the same turning radius on riding comfort. A real automated vehicle on test tracks was used, and riding comfort was evaluated by analyzing filled questionnaires. Although there exist several methods for constructing curves, and our experiments, four curves were constructed. The experimental vehicle, method of designing curves, experimental scenario, experimental results, and discussion have been explained in the following sections.

2. Related Works

Extensive research has been undertaken regarding driver human factors for automated vehicles.

Among the several human factors and the control of automated vehicles, comfort on control is critical for the introduction of automated vehicles [12].

Two types of controls for automated vehicles exist: longitudinal control [13–15] and lateral control. Du proposed a set of comfort-based velocity (longitudinal) control strategies based on driving safety. They insisted that the proposed strategy can potentially be applied to automated vehicles to improve the perceived quality of automated driving using field tests [13]. Reschka investigated the manner in which an autonomous vehicle can adapt its longitudinal control to changing road and weather conditions by calculating the proposed value and further provided an outlook on the way this parameter affected whole vehicle guidance [14]. Sohn proposed a longitudinal speed control strategy, which was called pulse and glide, and proved the improvement of driver satisfaction as well as fuel saving using the proposed method [15].

The present study focused on lateral control of riding comfort. The objective of lateral control is to allow the vehicle to travel on the desired course precisely. Researchers have proposed excellent lateral control algorithms [16–21], and using algorithms with optimal parameters, the vehicle can travel on the desired course with high accuracy. Park proposed a lateral control algorithm for automated vehicles using a proportional-integral-derivative (PID) controller [16] and validated the effectiveness of the proposed path at a curved path except at a low curvature path. Lee proposed a longitudinal control algorithm for a platoon of autonomous vehicles [17], which was evaluated using both simulators and real road experiments, proving its effectiveness. Chen proposed a path-following steering controller for an automated lane change system considering adaptive preview time. Further, using a simulator they validated that the proposed algorithm can effectively reduce the path-following error while reducing the lateral acceleration and jerk for a more comfortable lane change maneuver [18]. Yamakado focused on both longitudinal and lateral control of expert drivers while considering jerk information, and they confirmed that the model can emulate certain respect of the expert driver's control strategy for the trade-off between longitudinal and lateral acceleration [19]. Further, Alleyne performed a simulation and experimental study on automatic steering control and clarified that the steering actuator of the vehicle is a significant factor for model reference and PI-based control design [20]. Hingwe explained the design and experimental evaluation of a vehicle lateral controller for automated highway systems based on sliding mode control via experiments using a real passenger vehicle [21]. Except for lane-changing or obstacle avoidance, automated vehicles travel on the desired course. Hence, the driving comfort of traveling on the desired course depends on the course shape.

Several researchers have investigated human factors, particularly automated vehicles. Wang used a haptic interface for a driving assistance system for steering [22]. They concluded that the drivers tended to rely on haptic guidance to achieve better steering performance when visual feedback was limited via a simulation study [22]. Sentouh proposed a driver-automation cooperation-oriented approach, which is shared control between automated steering control and human driver on lane keeping. Their idea is based on a control design based on a human-in-the-loop vehicle system to reduce the conflict between the human driver and the system [23]. They proved the effectiveness of their proposed system experiments with real subjects.

Driving comfort has been investigated by several researchers. Chen investigated the human comfort of electrical vehicles and proposed a comfortable acceleration model [24]. Zuska experimented on the vibrational comfort of two passenger cars, wherein a defined and variable unbalance of a front wheel was set. They concluded that vibrational comfort was evaluated using the root mean square coefficient [25]. Butakov insisted on the importance of the personalization feature for automation and demonstrated the manner in which the methodology can be applied considering an example of adaptive cruise control and automatic lane-changing personalization [12]. Bae investigated the comfort experience lane-changing maneuver using the Bezier curve [26]. Xie proposed a personalized curve model for automated vehicles, insisting on the importance of the personalized curve for automated vehicles, and further evaluated the proposed model in a simulation study [27]. Lee researched the comfort zone for cyclists at different speeds and steering angles in field experiments [28]. To describe the importance of comfortable AV driving style, describing the importance of acceptance of AV is desirable, and it was found that those who did not prefer using AV preferred their own driving style. Oliveira reported user information on trust and acceptance of automated vehicles and insisted on the importance of improving trust and acceptance [29]. In addition, regarding the motion comfort of acceleration during driving, several studies insist that vibration is among the most important key factors for riding comfort. Mrad proposed a measurement technology to perform vibration analysis of a vehicle and a method that considered vibration with the suspension system. In addition, the best combination of rubber mounting, spring, and shock absorbers

focusing on the user's well-being was proposed [30]. Sezgin investigated the effect of vehicle vibrations on passengers using the ISO 2631 standard [31], which sets the standard for mechanical vibration and shock—evaluation of human exposure to whole-body vibration. Further, several studies have been conducted on motion sickness [32]. Htike reported an effective control algorithm for motion sickness in automated vehicles [33], and Li evaluated motion sickness in automated vehicles using simulations and experiments [34]. However, motion sickness was not considered as the participants did not indicate any signs of it. Further, considering these results, in this study, the effect of vibration was not considered because the experimental course was completely flat and the difference among the four courses was not found in the experiment.

Regarding the method for creating the curvature using the features of a clothoid, it should be noted that the clothoid curve is familiar to the steering control. Sedighi proposed an algorithm for generating a parking path and confirmed the validity of the algorithm [35]. Lima proposed predictive control for autonomous vehicles using the features of a clothoid in a simulation study [36]. Berglund proposed a smooth path using the B-spline curve instead of an arc for automated mining vehicles, which can be effective in avoiding obstacles [37]. Further, De-rong used a spline curve to smoothly fit traffic road accidents. The feature of the spline curve, which is easy to calculate, was applied [38]. Wang studied optimizing trajectories in the intersection for automated vehicles, focusing on the trajectory choices for safety, energy savings, and comfort [39]. The present study focuses on human factors, wherein types of curvatures are most appropriate in automated vehicles for human drivers, and factors affecting the results were clarified in real-world experiments. Thus, based on the results of previous studies and results, it is proposed that driving comfort is among the most important issues in the introduction of automated vehicles.

3. Experiments

3.1. Experimental Vehicle. The experimental vehicle used in the experiments (Figure 1) is described in this section. This vehicle was modified from a conventional hybrid vehicle and the system configuration of the experimental vehicle is shown in Figure 1. It can be controlled via a PC through a control area network, and the system can obtain vehicle status information through an OBD-II port. Furthermore, two RTK-GPS sensors (Novatel OEM628), a gyro sensor (Tamagawaseiki TA7265), and five laser scanner sensors (ibeo LUX) for environmental recognition were installed.

3.2. Conditions of Experiments. The experiments were conducted on a test track at the National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. To prevent other undesirable effects, no other vehicles were present on the test track during the experiment. The experiments were conducted during the day, in sunny conditions to maintain consistent conditions.

3.3. Scenario of Experiments. Twenty-five subjects participated in the study. To eliminate age bias, five subjects from each age group (20s, 30s, 40s, 50s, and 60s) were chosen, and the average age of the drivers was 44.9. The experimental flow is illustrated in Figure 2.

First, the subjects received an explanation of the experiments and questionnaires regarding automated driving. The questionnaire regarding the characteristics of the subjects was constructed considering the following factors:

(1) Personal characteristics (age, gender, and history of driving and traffic accidents)

(2) Driving frequency, vehicle type for usual driving, and motion sickness

(3) Comments on automated driving systems

After completing the questionnaires, experiments with automated driving were conducted. In the experiments, one subject was asked to sit in the passenger seat, while another subject sat in the driver's seat or in the second-row passenger seat in the vehicle. All the subjects experienced both the driver and passenger seats. Half of the subjects sat in the driver's seat first and then the passenger's seat and vice versa. The velocity of the vehicle was maintained at approximately 8.3 m/s (30.0 km/h). The effects of the varying speeds were clarified. In fact, preliminary experiments were performed with a small number of subjects before the experiments in this study by using the same experimental vehicle to evaluate the effect of speed. The preliminary experiments were performed under the same scenario at 20 km/h and 40 km/h, and it was found that the tendency of the result was the same as the result at 30 km/h. Before the experiment, we explained the experimental process (including how to stop and steer the vehicle in an emergency) in detail to all subjects, sufficient time was devoted to preparing the experiments because we ensured a safe experimental environment for the automated vehicles. Consequently, the necessary time for the explanations to the subjects before conducting the experiments was long. In addition, because the budget was limited, employing the subjects was not possible. Moreover, owing to the course limitation, the experiments could not be performed at high speeds. Thus, we concentrated on a constant speed. Consequently, as employing more subjects was not possible, we determined the relationship results statistically; therefore, in this study, it is assumed that the number of participants was sufficient.

The control algorithms, including control parameters, are different among car companies and developers of automated vehicles, and they are classified. However, the original objective of the control algorithm was to travel the desired course without deviation, which was assumed as such. However, as we set the parameters of our control algorithm through preliminary experiments, the effect of the control algorithm was removed when evaluating the trajectory of the vehicle. The objective of this study is to evaluate the comfort of the trajectory, not the control algorithm; thus, it was considered that our experimental evaluation was complete for meeting the objective. However, the control algorithm should also be evaluated, particularly

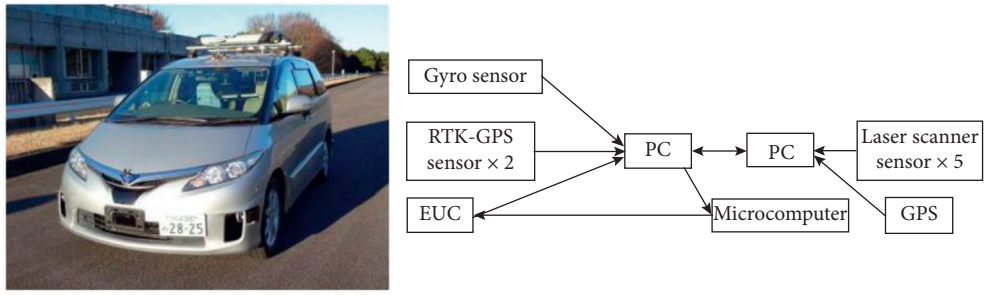


Figure 1: Picture and system configuration of the experimental vehicle.

1. Explanation to the subject about the experiments that are to be performed
2. Questionnaires about subject, automated vehicle, driving style
3. Experiments with automated driving in driver's seat or passenger's seat
4. Questionnaires about the experiments
5. Experiments with automated driving in passenger's seat or driver's seat
6. Questionnaires about the experiments

Figure 2: Experimental flow.

for other situations, wherein the experimental vehicle cannot travel the desired course without deviation due to certain conditions (weather, vehicle, etc.). We plan to conduct an evaluation in this respect in a future study.

Each subject experienced four course patterns. In the experiments, the order of the four courses was set randomly and introduced as courses 1, 2, 3, and 4 to the subjects. Thus, the subjects could not recognize which path corresponded to each of the four path patterns. The course patterns were an arc, clothoid curve, B-spline curve 1, and B-spline curve 2 (explanations have been provided in the next section). The automated vehicle traveled along the target course autonomously, and four patterns in each set were obtained. The subjects experienced two sets in each seat; thus, they performed 16 rides in both seats. For safety, two staff members sat in the first-row passenger seat (next to the driver's seat) and the second-row passenger seat, and every subject completed a training course to prepare for possible emergencies. Furthermore, each subject was asked to control the experimental vehicle for possible emergencies, implying that each subject was responsible when sitting in the driver's seat. There were no accidental situations during the experiment. According to the SAE international definition, the level of the automated vehicle was set to two in our experiments when the subject sat in the driver's seat and four when the subject sat in the passenger's seat [40]. This automated vehicle experiment was performed at an SAE level of 2. Although we did not ask subjects to watch the road, they were responsible for their vehicles; thus, they watched the road.

After one set of experiments, the subjects completed questionnaires regarding the experiments. Thus, the subjects were asked to complete these questionnaires four times in both seats. The questionnaire for each set asked the following questions:

Q1: Which path is comfortable? (multiple selections possible)

Q2: Which path is the most comfortable?

Q3: Why did you choose that path?

Q4: Which path is uncomfortable? (multiple selections possible)

Q5: Which path is the most uncomfortable?

Q6: Why did you choose that path?

Q7: Any comments on the experiments?

3.4. Four Patterns of Target Courses. In this experiment, four target course patterns were used. Each course consisted of straight lines and two curves (Figure 3). The differences in the four course patterns were in the shape of the curves. The curves were an arc, clothoid, B-spline 1, and B-spline 2, hereafter referred to as R50, CL50, BSP1, and BSP2, respectively. All curves were based on a 50 m radius and the real crossing in Hiroshima, Japan (Figure 4). Regarding the clothoid curve, clothoid equations can be defined starting from the condition of the linear relation between the radius (R) and length (L) of the path (equation (1)) as

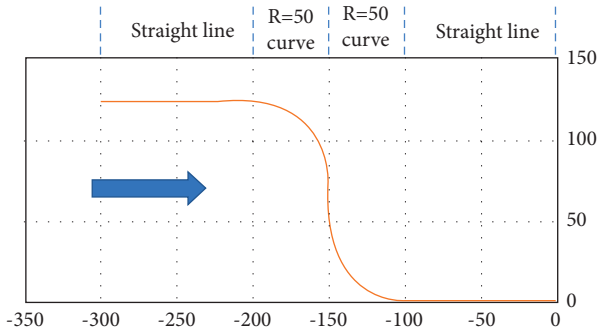


Figure 3: Experimental course of R50 (the arrow shows the direction of the vehicle).

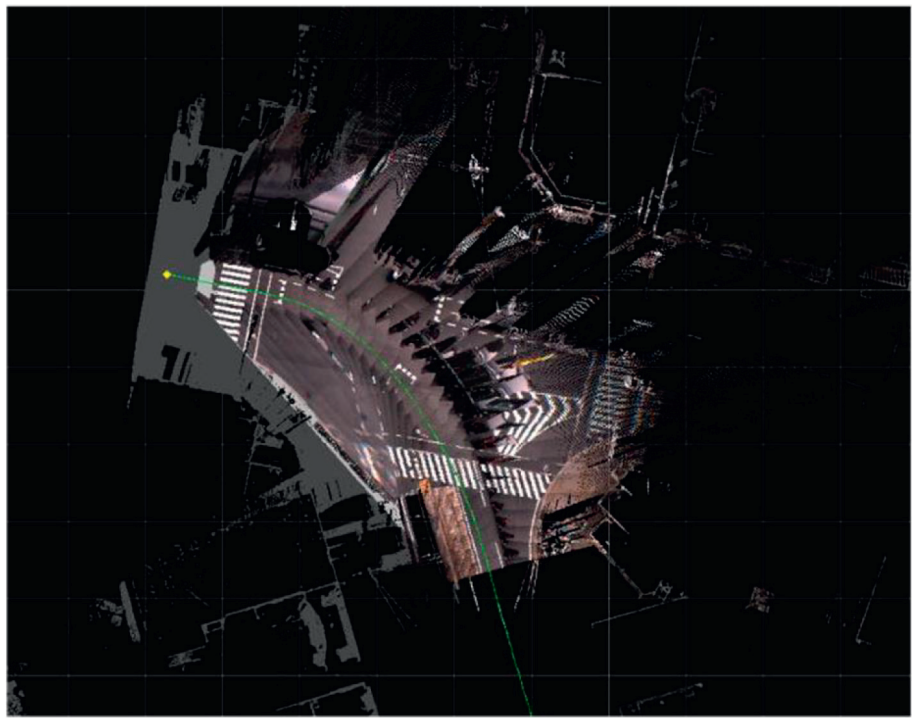


Figure 4: Sample used in Hiroshima, Japan.

$$R * L = A^2, \tag{1}$$

where R is the radius of curvature, L is the length measured along the spiral curve from its initial position, and A is the flatness or homothetic parameter of the clothoid.

In the present study, R and A were set to 50 and 35, respectively.

We employed a clothoid because the enables smooth steering control as indicated in several studies [41–44]. Consequently, we considered one hypothesis regarding comfort, which is that the clothoid curve is better for driving comfort. We conducted preliminary experiments with a few subjects and considering the results, we decided to prove our hypothesis with more subjects using real experimental vehicles. We believe that our experimental results validate our hypothesis.

BSP1 was prepared using the control points of the polyline (Figure 5(a)). Polylines were extracted along the channels (dots of the lane marker). Because spline conversion was performed to smoothen the polylines, the size of the BSP1 wobbles depends on the size of the original polyline rattle.

Further, BSP2 was made to pass three points: the initial point, midpoint, and terminal point of the clothoid curve (Figure 5(b)). Although BSP2 appears looser than the clothoid, on the initial and terminal points of the curve, there a sudden change from straight to curve and vice versa occurs. Typically, BSP2 requires higher transformation costs than BSP1 because the initial and terminal points of the clothoid must be recognized for generating BPS2.

Before performing the experiments with real subjects, preliminary experiments were performed for an objective evaluation. The experimental course for these experiments was set as shown in Figure 6. Each course consisted of straight lines and one curve. In the objective evaluation, evaluation of both left and right turns was not considered as

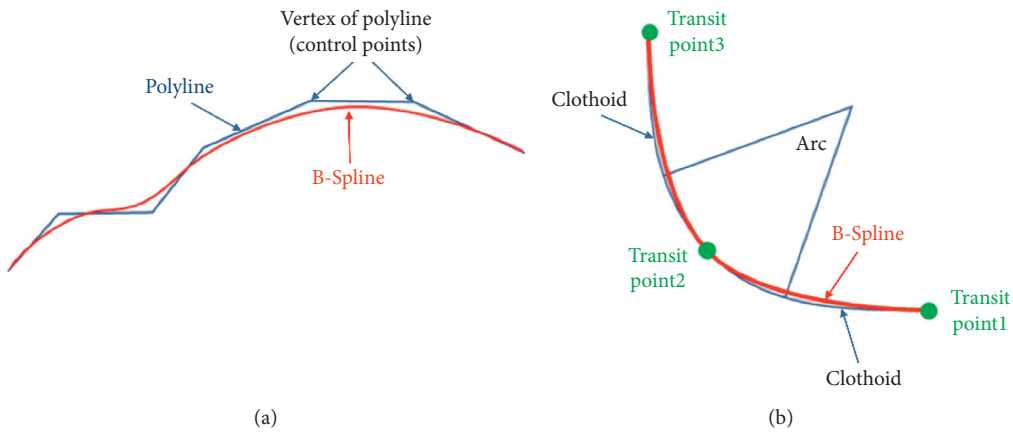


Figure 5: Method of forming B-spline curves. (a) B-spline 1 and (b) B-spline 2.

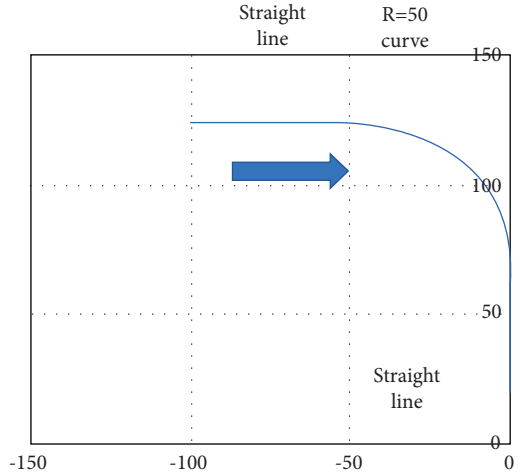


Figure 6: Experimental simple course for objective evaluations on the preliminary experiment.

necessary; thus, we employed a simple course. The lateral acceleration was evaluated, which was measured via an IMU 440 sensor, developed by Sumisei Sangyo. The scale factor accuracy of the sensor was 1%. Figure 7 shows the result of the lateral acceleration of each course, and the result when traveling on the curvature. From the left, the results of clothoid, R50, BSP1, and BSP2 are shown. The maximum value of lateral acceleration with CL50 was the lowest among the four courses. In addition, the absolute peak of jerk was calculated to evaluate riding comfort, as shown in Table 1, wherein CL50 is shown to have the smallest value. However, the difference between them is very small. Several papers have proposed the relationship between the jerk value and riding comfort [45–47]. Considering this result, CL50 was assumed to be the most comfortable.

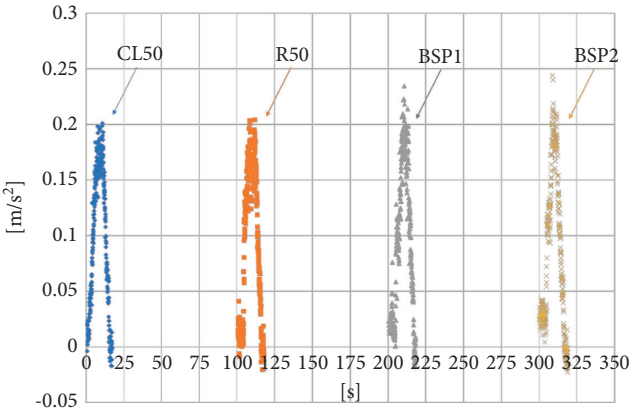


Figure 7: Experimental result for objective evaluation on the preliminary experiment (lateral acceleration).

4. Experimental Results and Discussion

The results of Q1: "Which path is comfortable?" (multiple selections possible) of the questionnaire are shown in Figure 8(a). The maximum number of each value is 100; hence, as evident, approximately 70% of the subjects chose CL50, while approximately 60% chose BSP2. The results of

Table 1: Result of peak jerk on lateral acceleration on the preliminary experiment.

Course	CL50	R50	BSP1	BSP2
Peak absolute jerk on lateral acceleration	1.11	1.16	1.14	1.70

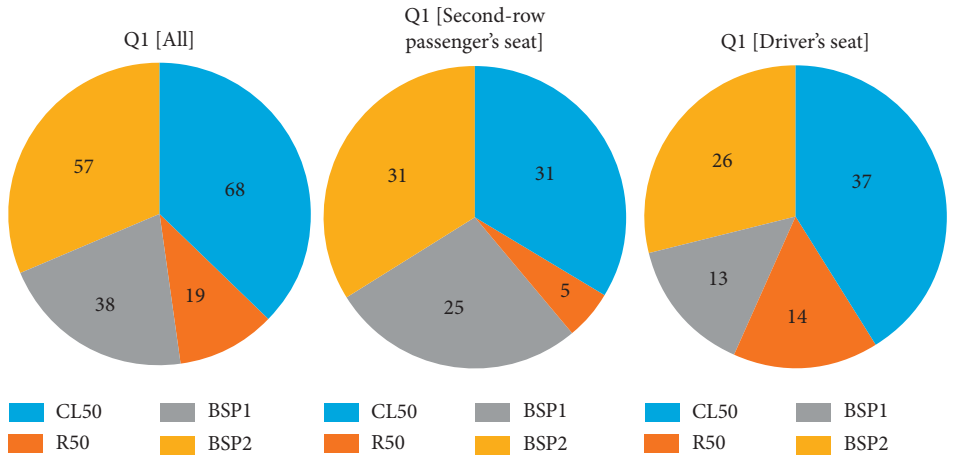


Figure 8: Questionnaire Result 1 (Q1: "Which path is comfortable? (multiple selections possible)"). CL50: clothoid, R50: arc, BSP1: B-spline 1, and BSP2: B-spline 2.

the difference in the subject's seat position are shown in Figure 8(c). Further, a comparison of the driver's seat and second-row passenger's seat results demonstrated that the latter tended not to choose R50.

Questionnaire results of Q2: "Which path is the most comfortable?" are shown in Figures 9(a)–9(c). It indicates that more than 50% of the subjects chose CL50 as the most comfortable path and that the clothoid curve was appropriate for a comfortable path. The questionnaire results for the riding positions are also shown in Figures 9. The CL50 was the most answered choice in both the driver and passenger seat positions, and subjects in the driver's seat tended to choose CL50 more than in the passenger seat. Regarding the first and second trials, the questionnaire results from the experience of tests are also shown in Figures 9(d) and 9(e). As explained in the experimental section, two tests were performed in each scenario. The "1st" and "2nd" are the orders of experiments. This result has the same tendency as the other results. Further, regarding gender, the questionnaire results by gender are also shown in Figures 9(f) and 9(g). The difference between this result and the others is that the score of BSP1 is larger than that of BSP2, but the largest score is for CL50, the same as the others. In addition, the questionnaire results for nonelderly and elderly individuals are also shown in Figures 9(h) and 9(i). Elderly subjects chose different answers because they had different feelings. To evaluate the results statistically, we employed the chi-squared test. The chi-squared test is commonly used in academic papers [48–50]. The formulation of the chi-squared test is as follows:

$$X_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}, \tag{2}$$

where O_i is the observed numbers and E_i is the expected numbers.

The results of the chi-squared test for Q2 are presented in Table 2. All questionnaire results for Q2 were satisfied with a 1% significance level. Further, subjects can identify four types of courses and decide which is the most comfortable. The questionnaire result of Q3: "Why did you choose that path?" is shown in Figure 10. The subjects provided reasons for choosing the path. Most subjects chose smooth control or less shaking. As clothoid is familiar with steering control [26] resulting in a small jerk, it is presumed that the subjects felt comfortable.

The questionnaire results of Q4: "Which path is uncomfortable? (multiple selections possible)" are shown in Figure 11. The maximum number of each value is 100; therefore, approximately 70% of the subjects chose R50, while approximately 60% chose BSP1. Regarding the driving position, subjects in the driver's seat chose BSP1 more than R50; although, the difference was small. It is presumed that subjects in the driver's seat can observe the shaking of the steering angle directly because the steering angle is subjected to minute shaking when traveling on BSP by the feature of BSP1.

The questionnaire results of Q5: "Which path is the most uncomfortable?" are shown in Figure 12. These results show that more than 50% of the subjects chose R50 as the most uncomfortable path. The results of the chi-squared test for Q5 are shown in Table 3. From these results, all questionnaire results of Q5, except for the driver seat, were satisfied with a 1% significance level.

Thus, the result indicates that R50 should not be applied as a comfortable path. Furthermore, subjects can identify the four types of courses and decide which is the most uncomfortable. The questionnaire results based on riding positions are also shown in Figures 12(a)–12(c). The choice of R50 is the maximum in both the driver and passenger seat positions, but in the former, the result of the chi-squared test is not satisfied with a 5% significance level. It is presumed that the subjects in the driver's seat could observe the steering angle directly (Figure 11). Further, the maximum number of R50 choices is the same as that of the others. Regarding the first and second trials, the questionnaire results from the experience of tests are shown in Figures 12(d) and 12(e). No differences were observed in the order of these results, which have the same tendency as the other results. The questionnaire's results by gender are also shown in Figures 12(f) and 12(g). Furthermore, these results show no differences between genders and have the same

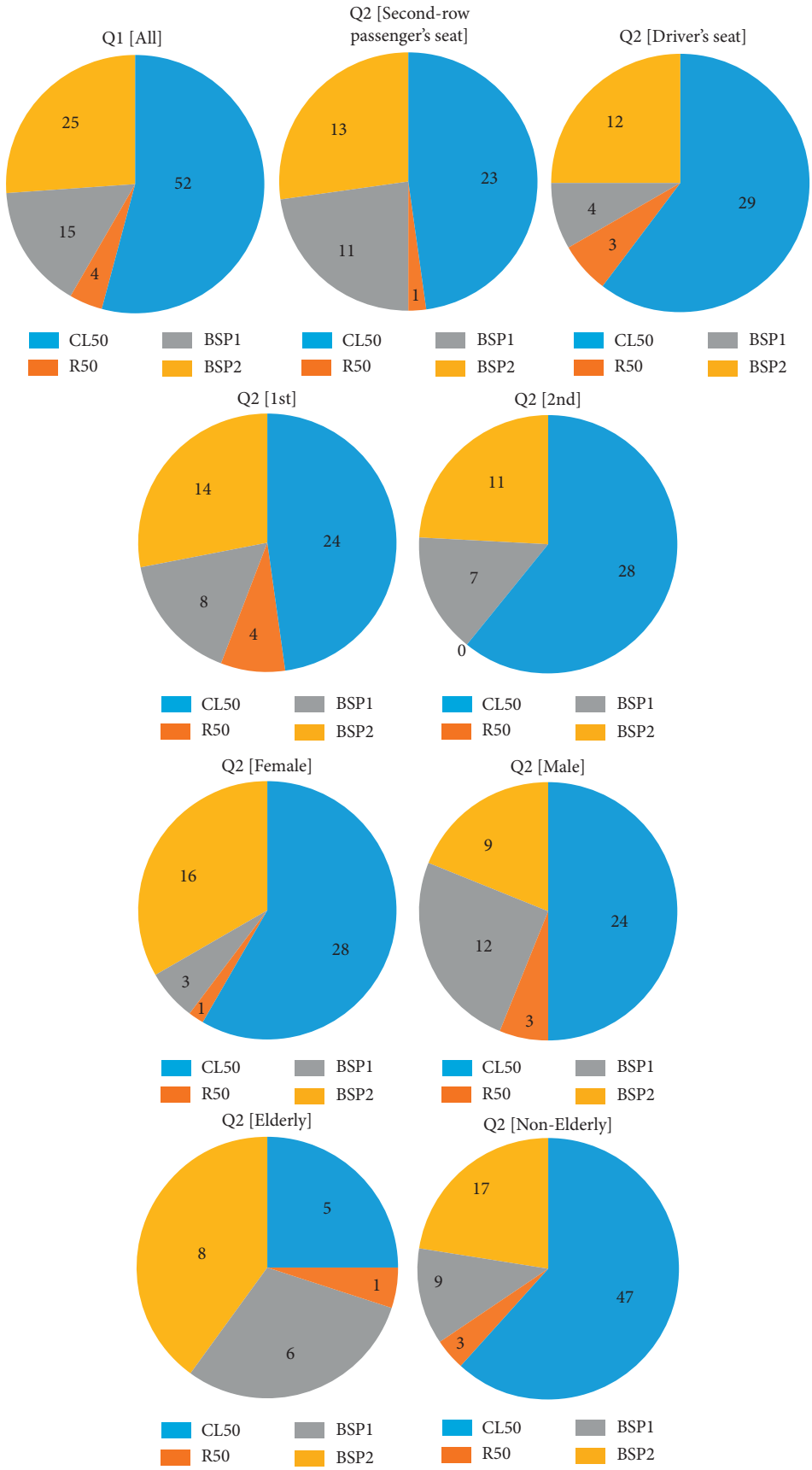


Figure 9: Questionnaire Result 2 (Q2: "Which path is the most comfortable?"). CL50: clothoid, R50: arc, BSP1: B-spline 1, and BSP2: B-spline 2.

Table 2: Chi-squared test result of Q2: "Which path is the most comfortable?".

	(Which is the most comfortable?) degree of freedom = 3								
	All	Ds	Rs	1st	2nd	Female	Male	Nonelderly	Elderly
X^2 ($7.82 \leq, p < 0.05, 11.34 \leq, p < 0.01$)	52.75	24.08	20.33	18.16	36.96	39.5	19.5	60.2	5.2

Ds: driver's seat; Rs: second-row passenger's seat.

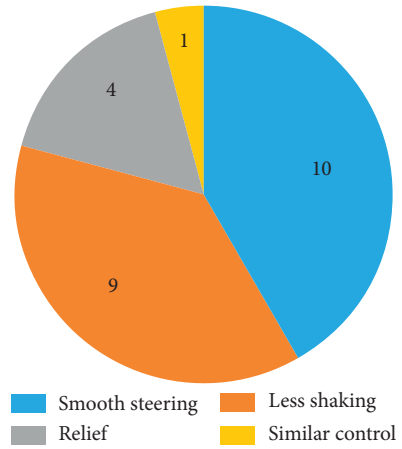


Figure 10: Questionnaire Result 3 (Q3: "Why did you choose that path?").

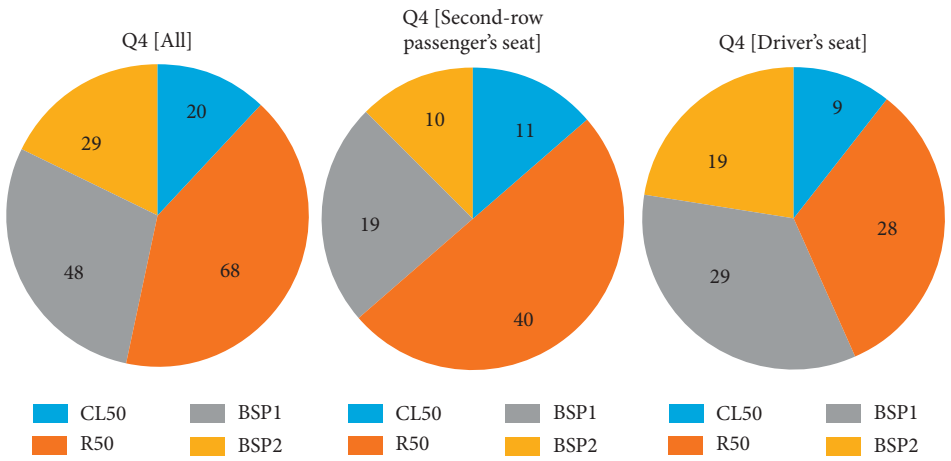


Figure 11: Questionnaire Result 4 (Q4: "Which path is the most uncomfortable (multiple selections possible)"). CL50: clothoid, R50: arc, BSP1: B-spline 1, and BSP2: B-spline 2.

tendency as the other results. Regarding age, the questionnaire results for the nonelderly and elderly are also shown in Figures 12(h) and 12(i). These results exhibit the same tendency as in Figure 11, showing a less statistical difference between the answers. However, the introduction of an automated driving system for elderly drivers should be considered from other perspectives. The questionnaire result of Q6: "Why did you choose that path?" is shown in Figure 13. The subjects also provided reasons for choosing the path. The reasons were primarily not smooth and shaking, which are not as complicated to comprehend as Q3. From this result and the chi-squared test results, it is presumed that it is easier to decide on a comfortable path than on an uncomfortable path.

The comments regarding the experiments are presented in Figure 14. The maximum number of comments was related to the number of repetitions. As shown in the results of the chi-squared test in Table 2, the value of the 2nd was larger than that of the 1st. After one trial, the subjects experienced the automated vehicle and understood the difference better. Further, from the results regarding the riding position in Table 3, Figure 11, and Figure 14, it is presumed that observing the movement of the turning steering wheel could affect comfort; therefore, a human-machine interface for passengers should be installed to improve acceptability.

In addition, with respect to the feeling of safety, the results of this experiment included the feeling of safety, because if subjects felt unsafe on the course, they opted not

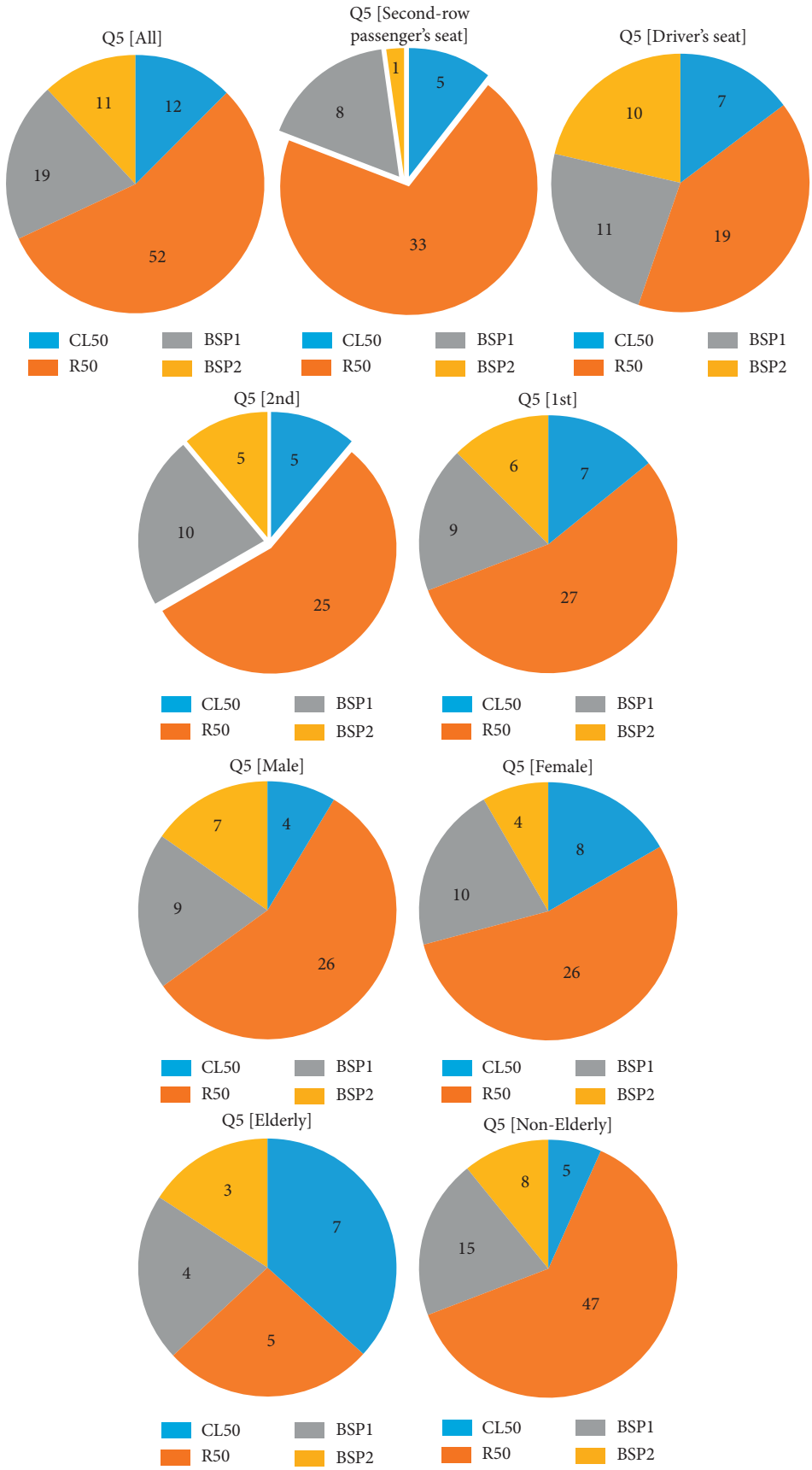


Figure 12: Questionnaire Result 5 (Q5: "Which path is the most uncomfortable?"). CL50: clothoid, R50: arc, BSP1: B-spline 1, and BSP2: B-spline 2.

Table 3: Chi-squared test result of Q5: "Which path is the most uncomfortable?".

	(Which is the most uncomfortable?) degree of freedom = 3								
	All	Ds	Rs	1st	2nd	Female	Male	Nonelderly	Elderly
X^2 ($7.82 \leq, p < 0.05, 11.34 \leq, p < 0.01$)	47.70	6.70	53.34	24.06	23.89	23.33	18.16	59.56	1.84

Ds: driver's seat; Rs: second-row passenger's seat.

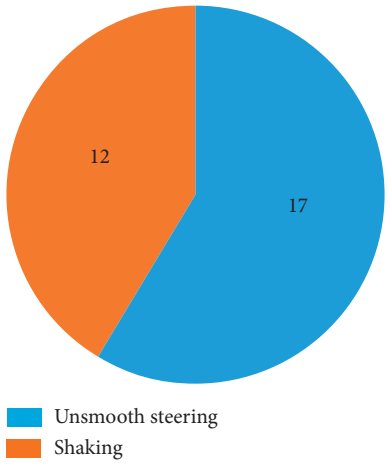


Figure 13: Questionnaire Result 6 (Q6: "Why did you choose that path?").

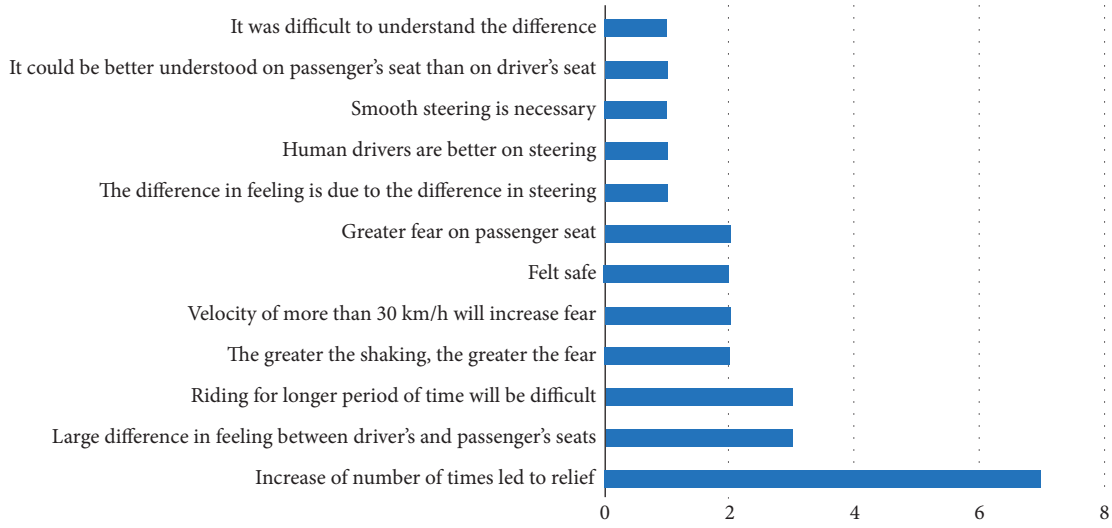


Figure 14: Questionnaire Result 7 (Q7: "Any comments on the experiments?").

to choose the course. Thus, it is presumed that the results show that clothoid provides the best feeling of safety among the four courses. Hence, in future studies, the feeling of safety should be considered along with the function of obstacle avoidance. We also enquired about motion sickness in the questionnaire because the human factor of motion sickness might affect the results in our hypothesis. However, we could not find any relationship between the results and the factor of motion sickness as no subject reported the occurrence of motion sickness.

As explained in the scenario of experiments section, the level of automation of the experiments was set to two [40]. In the future, levels 3, 4, or 5 automated vehicles may be introduced [40]. A driver who rides on level 3, 4, or 5 automated vehicles can perform eyes-off road driving and nondriving tasks because the driver is not responsible for driving. In contrast, the driver who drives the level 2 automated vehicle is responsible for the vehicle. In Japan, only two types of vehicles with level 3 functions are produced. Level 4 and 5 automated vehicles have not yet been allowed by the government. Given the current situation in Japan, as level 3 automated vehicles are not yet well known, it is difficult for subjects to imagine level 3 automated vehicles. In Japan, several experiments related to level 2 automated vehicles and level 2 assistance systems are well known. In our experiments, we explained to the subjects that the level of

automated vehicles was level 2, and the responsibility was with the driver. In addition, for developing level 3 vehicles, the process of providing all explanations before using the vehicles is necessary, which results in requiring more time to prepare the experiments in real vehicles rather than in the simulator. Thus, in this study, it was difficult to perform the experiments with level 3, 4, or 5 automated vehicles. However, if the experimental vehicle is a level 3, 4, or 5 automated vehicle, the experimental result may be different, and performing experiments related to the eyes-off situation and nondriving task would be possible.

5. Conclusions

The present study investigated several courses to determine the most comfortable course for further introduction of automated vehicles. The conditions for the different curves based on real situations were assumed. This study evaluated the effect of different courses on driver comfort in automated vehicles through field experiments with 25 subjects. The automated vehicle drove at 30 km/h on four different courses (clothoid, two types of spline curves, and arc). The experimental data indicated that the most comfortable course was clothoid, and the most uncomfortable course was an arc when the significance level was set at 1%. The reasons for this were primarily smoothness while driving. Thus, the experimental results show that, for driving comfort, paths for automated vehicles should be clothoid and not arc. These tendencies were effective for driver and passenger seats, gender, and experience (number of times).

In the future, we plan to conduct more experiments with automated vehicles other conditions, including more paths and more scenarios. In addition, we plan to expand the number of subjects and evaluate other human factors, including occupation.

Data Availability

The data used to support the findings of this study have not been made available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the New Energy and Industrial Technology Development Organization.

References

[1] "US. DOT," April 2021, http://www.its.dot.gov/automated_vehicle/index.htm.

[2] "ITS Japan," April 2021, http://www.its-jp.org/english/files/2015/04/SIP_Worlshop2015_leaflets_e_20150326.pdf.

[3] European Commission, "IOT large scale pilot 5 autonomous vehicles in a connected environment," April 2021, https://www.ertrac.org/uploads/documents_publications/2015%20ART%20Info%20Day/IoTPilot-autonomous%20vehicle-nov5.pdf.

[4] M. Aeberhard, S. Rauch, M. Bahram et al., "Experience, results and lessons learned from automated driving on Germany's highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 42–57, 2015.

[5] S. Shladover, "Challenges to evaluation of CO2 impacts of intelligent transportation systems," in *Proceedings of the 2011 IEEE Integrated and Sustainable Transportation System*, Vienna, Austria, June-July 2011.

[6] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 68–77, 2016.

[7] R. Hoeger, A. Amditis, M. Kunert et al., "Highly automated vehicles for intelligent transport: HAVEit approach," in *Proceedings of the 15th ITS World Congress*, New York, NY, USA, November 2008.

[8] F. O. Flemisch, K. Bengler, H. Bubb, H. Winner, and R. Bruder, "Towards cooperative guidance and control of highly automated vehicles: H-mode and conduct-by-wire," *Ergonomics*, vol. 57, no. 3, pp. 343–360, 2014.

[9] E. Boer and P. Spyridakos, "Control-theoretic attention-switching driver model," in *Proceedings of the International Conference on Traffic and Transport Psychology*, Brisbane, Australia, August 2016.

[10] M. Omae, N. Hashimoto, T. Sugamoto, and H. Shimizu, "Measurement of driver's reaction time to failure of steering controller during automatic driving," *Review of Automotive Engineering*, vol. 26, pp. 213–215, 2005.

[11] G. M. Fitch, R. J. Kiefer, J. M. Hankey, and B. M. Kleiner, "Toward developing an approach for alerting drivers to the direction of a crash threat," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 49, no. 4, pp. 710–720, 2007.

[12] V. Butakov and P. Ioannou, "Driving autopilot with personalization feature for improved safety and comfort," in *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas de Gran Canaria, Spain, September 2015.

[13] Y. Du, C. Liu, and Y. Li, "Velocity control strategies to improve automated vehicle driving comfort," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 1, pp. 8–18, 2018.

[14] A. Reschka, J. R. Böhmer, F. Saust, B. Lichte, and M. Maurer, "Safe, dynamic and comfortable longitudinal control for an autonomous vehicle," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Alcala de Henares, Spain, June 2012.

[15] C. Sohn, J. Andert, and R. N. Nanfah Manfouo, "A driveability study on automated longitudinal vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3273–3280, 2020.

[16] M. Park, S. Lee, and W. Han, "Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm," in *Proceedings of the 2014 14th International Conference on Control, Automation and Systems*, Gyeonggi-do, South Korea, October 2014.

[17] G. Lee, S. Kim, Y. Yim, J. Jung, S. Oh, and B. Kim, "Longitudinal and lateral control system development for a platoon of vehicles," in *Proceedings of the 1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, Tokyo, Japan, October 1999.

[18] B.-C. Chen, C.-T. Tsai, and K. Lee, "Path-following steering controller of automated lane change system with adaptive preview time," in *Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, Kowloon, Hong Kong, October 2015.

[19] M. Yamakado and M. Abe, "Proposal of the longitudinal driver model in coordination with vehicle lateral motion based upon jerk information," in *Proceedings of the 2008 International Conference on Control, Automation and Systems*, Seoul, South Korea, October 2008.

[20] A. Alleyne and M. DePoorter, "Lateral displacement sensor placement and forward velocity effects on stability of lateral control of vehicles," in *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico, June 1997.

[21] P. Hingwe and M. Tomizuka, "Experimental evaluation of a chatter free sliding mode control for lateral control in AHS," in *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico, June 1997.

[22] Z. Wang, R. Zheng, T. Kaizuka, and K. Nakano, "The effect of haptic guidance on driver steering performance during curve negotiation with limited visual feedback," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 600–605, Redondo Beach, CA, USA, June 2017.

[23] C. Sentouh, A.-T. Nguyen, M. A. Benloucif, and J.-C. Popieul, "Driver-automation cooperation oriented approach for shared control of lane keeping assist systems," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1962–1978, 2019.

[24] S. Chen, S. H. Li, B. Ma, and Z. Ran, "Ride comfort control of electrical vehicle in the process of speed regulation," in *Proceedings of the 2006 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 2006.

[25] A. Zuska and D. Więckowski, "The impact of unbalanced wheels and vehicle speed on driving comfort," in *Proceedings of the 2018 XI International Science-Technical Conference Automotive Safety*, Častá, Slovakia, April 2018.

[26] I. Bae, J. H. Kim, J. Moon, and S. Kim, "Lane change maneuver based on Bezier curve providing comfort experience for autonomous vehicle users," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, October 2019.

[27] J. Xie, J. Gong, S. Wu, G. Xiong, and C. Lu, "A personalized curve driving model for intelligent vehicle," in *Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS)*, Beijing, China, October 2017.

[28] O. Lee, A. Rasch, A. L. Schwab, and M. Dozza, "Modelling cyclists' comfort zones from obstacle avoidance manoeuvres," *Accident Analysis & Prevention*, vol. 144, p. 105609, Article ID 105609, 2020.

[29] L. Oliveira, K. Proctor, C. G. Burns, and S. Birrell, "Driving style: how should an automated vehicle behave?" *Information*, vol. 10, no. 6, p. 219, 2019.

[30] F. L. Mrad, D. M. L. Machado, G. J. C. Horta, and A. U. Sad, "Optimization of the vibrational comfort of passenger vehicles through improvement of suspension and engine rubber mounting setups," *Shock and Vibration*, vol. 2018, pp. 1–9, Article ID 9861052, 2018.

[31] A. Sezgin and N. Yagiz, "Analysis of passenger ride comfort," in *Proceedings of the International Conference on Structural Nonlinear Dynamics and Diagnosi MATEC Web of Conferences*, vol. 1, Marrakech, Morocco, April-May 2012.

[32] "ISO standard 2631," August 2021, https://www.iso.org/standard/76369.html.

[33] Z. Htike, G. Papaioannou, E. Siampis, E. Velenis, and S. Longo, "Minimisation of motion sickness in autonomous vehicles," in *Proceedings of the 2020 IEEE Intelligent Vehicles Symposium*, pp. 1135–1140, Nagoya, Japan, July 2020.

[34] D. Li and J. Hu, "Mitigating motion sickness in automated vehicles with frequency-shaping approach to motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7714–7720, 2021.

[35] S. Sedighi, D. V. Nguyen, and K. D. Kuhnert, "A new method of clothoid-based path planning algorithm for narrow perpendicular parking spaces," in *Proceedings of the Paper Presented at Proceedings of ICMRE 2019*, Rome, Italy, February 2019.

[36] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *Proceedings of the Paper Presented at 2015 European Control Conference (ECC)*, Linz, Austria, July 2015.

[37] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010.

[38] T. De-rong and Y. Peng, "Study on the forecast method of road accidents based on extension of cubic spline curve," in *Proceedings of the Paper Presented at 2009 Second International Conference on Intelligent Computation Technology and Automation*, Changsha, China, October 2009.

[39] Y. Wang, P. Cai, and G. Lu, "Cooperative autonomous traffic organization method for connected automated vehicles in multi-intersection road networks," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 458–476, 2020.

[40] SAE International, *Automated Driving-Levels of Driving Automation are Defined in New SAE International Standard J3016*, SAE International, Warrendale, PA, USA, 2021, https://www.sae.org/misc/pdfs/automated_driving.pdf.

[41] N. Montes, A. Herraez, L. Armesto, and J. Tornero, "Real-time clothoid approximation by Rational Bezier curves," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pp. 2246–2251, Pasadena, CA, USA, May 2008.

[42] M. Yamamoto and S. Nishiyama, "Proposal for a transition curve with smoothly changing curvature using multiple clothoid curve," *Transactions of the JSME*, vol. 85, no. 873, 2019, in Japanese.

[43] D. J. Walton and D. S. Meek, "A controlled clothoid spline," *Computers & Graphics*, vol. 29, no. 3, pp. 353–363, 2005.

[44] V. Girbés, G. Vanegas, and L. Armesto, "Clothoid-based three-dimensional curve for attitude planning," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1886–1898, 2019.

[45] N. Yusof, J. Karjanto, J. Terken, F. Delbressine, and M. Hassan, "The exploration of autonomous vehicle driving styles: preferred longitudinal, lateral, and vertical accelerations," in *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 245–252, Ann Arbor, MI, USA, October 2016.

[46] K. Ikeda, A. Endo, R. Minowa, T. Narita, and H. Kato, "Ride comfort control system considering physiological and psychological characteristics: effect of masking on vertical vibration on passengers," *Actuators*, vol. 7, no. 3, p. 42, 2018.

[47] J. Zhou, R. He, Y. Wang et al., "Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 439–446, 2021.

[48] Y. Li, "Applications of chi-square test and contingency table Analysis in customer satisfaction and empirical analyses," in *Proceedings of the 2009 International Conference on Innovation Management*, pp. 105–107, Wuhan, China, December 2009.

[49] B. Brumback and M. Srinath, "A Chi-square test for fault-detection in Kalman filters," *IEEE Transactions on Automatic Control*, vol. 32, no. 6, pp. 552–554, 1987.

[50] B. Ahmed and P. H. Holmes, "A voice activity detector using the chi-square test," in *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I–625, Montreal, Canada, May 2004.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9940126, 18 pages
https://doi.org/10.1155/2021/9940126

Research Article

At the Traffic Intersection, Stopping, or Walking? Pedestrian Path Prediction Based on KPOF-GPDM for Driving Assistance

Xudong Long,[1] **Weiwei Zhang**,[2] **Bo Zhao**,[1] **and Shaoxing Mo**[1]

[1]*Shanghai Engineering and Technology University, Shanghai 200000, China*
[2]*Tsinghua University, Beijing 100000, China*

Correspondence should be addressed to Weiwei Zhang; zwwsues@163.com

Received 30 March 2021; Revised 2 July 2021; Accepted 14 July 2021; Published 30 July 2021

Academic Editor: Peng Hang

Pedestrian detection has always been a research hotspot in the Advanced Driving Assistance System (ADAS) with great progress in recent years. However, for the ADAS, we not only need to detect the behavior of pedestrians in front of the vehicle but also predict future action and the motion trajectory. Therefore, in this paper, we propose a human key point combined optical flow network (KPOF-Net) in the vehicle ADAS for the occlusion situation in the actual scene. When the vehicle encounters a blocked pedestrian at a traffic intersection, we used self-flow to estimate the global optical flow in the image sequence and then proposed a White Edge Cutting (WEC) algorithm to remove obstructions and simply modified the generative adversarial network to initialize pedestrians behind the obstructions. Next, we extracted pedestrian optical flow information and human joint point information in parallel, among which we trained four human key point models suitable for traffic intersections. At last, KPOF-GPDM fusion was proposed to predict the future status and walking trajectories of pedestrians, which combined optical flow information with human key point information. In the experiment, we did not merely compare our method with other four representative approaches in the same scene sequences. We also verified the accuracy of the pedestrian motion state and motion trajectory prediction of the system after fusion of human joint points and optical flow information. Taking into account the real-time performance of the system, in the low-speed and barrier-free environment, the comparative analysis only uses optical flow information, human joint point information, and KPOF-Net three prediction models. The results show that (1) in the same traffic environment, our proposed KPOF-Net can predict the change of pedestrian motion state about 5 frames (about 0.26 s) ahead of other excellent systems; (2) at the same time, our system predicts the trajectory of the pedestrian more accurately than the other four systems, which can achieve more stable minimum error ±0.04 m; (3) in a low-speed, barrier-free experimental environment, our proposed trajectory prediction model that integrates human joint points and optical flow information has higher prediction accuracy and smaller fluctuations than a single-information prediction model, and it can be well applied to automobiles' ADAS.

1. Introduction

In the automatic driving scene, efficient detection of vehicles and pedestrians around the vehicle has become the basic ability of autonomous vehicles [1]. Recently, some researchers focused their attention on the understanding of pedestrians' behaviors and intentions in front of the car and conducted simulation experiments in the Atlanta world assumption [2]. If the collision between pedestrians and vehicles can be predicted in advance, many unnecessary traffic accidents can be averted. For example, in the ADAS of Mercedes-Benz E-class and S-class car models [3], a pedestrian prediction algorithm based on stereo vision is introduced, which is applied in emergency braking in dangerous scenarios. In complex scenarios, such as intersections and crosswalks, it is necessary to accurately estimate the current and future positions of pedestrians relative to the moving vehicle. In the process of pedestrian trajectory prediction, we need to consider several influencing factors. First of all, for pedestrians showing more random during walking on the road, such as the interaction between people, a particular pedestrian trajectory is affected by the position of other pedestrians. In addition, people with social attributes will also have an impact on the final trajectory, and the

quantification of these indicators is a cumbersome process. Secondly, the pedestrian movement in the eyes of the ADAS is regarded as the common result of pedestrian movement and vehicle movement. Therefore, the prediction range of the active pedestrian prediction system is very short with even a small improvement eliciting a significantly improved performance. This article focuses on the prediction of pedestrian positions at intersections and crosswalks. With the method of fusion of optical flow information and joint point models, the paper first predicts whether the pedestrian's state in the future is standing or stopping, and then forecasts the pedestrian's trajectory and position in the future.

With auxiliary information provided for the ADAS, it discusses the status and location information of pedestrians in the future, adjusts the speed in advance to avoid traffic accidents due to proximity between the car and the pedestrian.

In summary, we highlight our main contributions as follows:

(i) We propose KPOF-Net, a novel framework of pedestrian trajectory prediction algorithm, which combines optical flow information and pedestrian joint models to collaboratively predict the state and trajectory of pedestrians in the future.

(ii) By evaluating the complexity of the application scene, we propose using an optical flow estimation module to estimate the optical flow of pedestrians with occlusion through the self-flow network, and then design a WEC algorithm based on Canny to remove occluded objects, and finally modify UCTGAN slightly to generate complete pedestrian optical flow information.

(iii) Through a large number of observations and researches on pedestrian motion status at intersections and crosswalks, we used the posture change information when pedestrian motion status changes to train four human joint point models of standing, stopping, standing tendency, and stopping tendency. The human joint point model trained at this time can more accurately predict the motion state and motion trajectory of pedestrians in the future.

(iv) Considering that a single optical flow cannot obtain detailed information of the pedestrian's posture at a traffic intersection, it can only roughly predict the pedestrian's motion state and trajectory in the future. The KPOF-GPDM prediction algorithm is proposed, which integrates optical flow and human joint point information, and it combines the movements of the upper and lower limbs of pedestrians in different traffic situations to predict the movement state and trajectory of pedestrians in the future, providing more efficient and active safety data for the ADAS.

(v) We propose ADE and FDE evaluation methods based on Euclidean distance, and compare and analyze the effect of KPOF-Net fusion of optical flow information and joint point model in improving the accuracy of pedestrian trajectory prediction in an unobstructed experimental environment.

2. Related Work

In this section, we provide a review of the optical flow and key point prediction approaches for pedestrian trajectory prediction under occlusion. We focus on learning related research to solve the problem of pedestrian trajectory prediction at intersections and sidewalks.

2.1. Optical Flow. Optical flow estimation is mainly divided into three categories: Supervised Learning of Optical Flow, Unsupervised Learning of Optical Flow, and Self-Supervised Learning. FlowNet [4] is the first end-to-end optical flow learning framework, which takes continuous images as input and dense optical flow graphs as output. SpyNet [5] uses a pyramid network with a compact space structure to scale the image to deal with the large-scale displacement of the object. LiteFlowNet [6] achieves lightweight by distorting the feature objects extracted by CNNs [4, 6, 7]. However, this type of method needs to use the rules [4, 8] to pretrain multiple synthetic datasets, which consumes a lot of time, and it involves low-speed, offline operation, and not real time. Moreover, the result is too dependent on the pretraining results of the synthetic dataset, and its optical flow accuracy does not meet our scenario requirements. The unsupervised learning method mainly uses the principle of constant brightness [9] and spatial smoothness [10], by measuring the pixel difference between the initial image and the test image, which can handle optical flow estimation with obstructions. Janai et al. [11] use multiple frames of images to jointly derive optical flow images. However, the detection accuracy of this scheme needs to be improved. DDFlow [12] proposes an optical flow data distillation method to learn the optical flow of occluded objects, but this type of method has limitations, which means it can only handle occluded objects under specific circumstances and cannot be applied to all scenes. Self-supervised learning adopts the data itself as a supervised signal, which is widely used to learn features from unlabeled data [13], and is often used to deal with image restoration [14], image coloring [15], and stitching problems [16]. Doersch and Zisserman [17] combine feature learning based on low-level motion cues. The study in [18] proposes S4L-Rotation and S4L-Exemplar algorithms to deal with the classification loss problem. However, the linear classifier obtained by this method is very dependent on the adjustment strategy of the learning rate and has uncertainty. Self-flow [19] takes reliable predictions of nonoccluded pixels as the self-supervision signal to guide our optical flow learning of occluded pixels. The network not only has a simple structure but also changes the image pyramid to a feature pyramid and uses a multiframe input method, which increases the information input of the network, and has good effects in terms of progress and real-time performance. In the scene of intersection and sidewalk, considering the comprehensive robustness of the automotive ADAS, the self-flow network is used to extract the optical flow information of pedestrians, and at the same time, when pedestrians are blocked by obstructions, a WEC algorithm based on Canny [20] was proposed on the basis of self-flow by

comprehensively considering the complexity and timeliness of the optical flow system as well as the accuracy of trajectory tracking in the presence of obstacles. To obtain smooth pedestrian optical flow, an UCTGAN network [21] was also used to jointly recover pedestrian optical flow in blocked scenes.

2.2. Key Point Prediction. In the past few decades, human pose estimation [22] and pedestrian trajectory prediction [23] have made rapid development. In the KF [24], the current state of a dynamic system can be propagated to the future by means of the underlying linear dynamical model without the incorporation of new measurements. IMM KF [24] introduces a similar method to predict pedestrian trajectories in a multithreaded dynamic model. Choi and Savarese [25] propose a framework that can track multiple objects, recognize the atomic activities performed by individuals, such as walking or standing, identify interactions between pairs of individuals (i.e., interactive activities), and understand the activities of groups of individuals. However, this method could appear to be biased in assigning interaction labels. In the research process, Hu et al., respectively, proposed an improved Bernoulli heatmap [26] and a new convolutional recurrent network model [27] to estimate the joint point information of various parts of the human body. Although these methods can quickly and accurately construct a human head joint point model, the performance needs to be improved when processing large-angle samples. Karasev et al. [28] proposed to use the Jump-Markov process to model the pedestrian's movement and infer the state of the pedestrian through the Rao-Blackwellized filter. However, the predictable change event types of this scheme are limited, and it cannot be widely used in various traffic scenarios. Anca Marginean et al. [29] proposed a set of pose-based and recursive framework-based algorithms to deal with imbalances in pedestrian estimation. When our scene is set at intersections and sidewalks, Keller and Gavrila [30] used the Gaussian dynamics model and probabilistic hierarchical trajectories based on dense optical flow to obtain pedestrian characteristics. Goldhammer et al. [31] proposed the use of polynomial least squares approximation and multilayer perceptron (MLP) to predict the trajectory of pedestrians in the next 2.5 s. However, the accuracy of pedestrian trajectories predicted by this method needs to be improved. The study in [32] also used a similar method to predict the trajectory of pedestrians riding bicycles. Alahi et al. [33] proposed an algorithm based on LSTM to predict the trajectory of pedestrians by considering the interdependence of pedestrians. Urtasun et al. [34] used a GPDM to track a small number of 3-D body points that have been derived using an image-based tracker and the system is trained with one gait cycle from six subjects and is able to handle several frames of occlusions. However, the exported 3D body points are limited, and the occlusion processing effect is not good. Minguez et al. [35] used balanced GPDMs for intention detection and trajectory forecasting of pedestrians based on 3D poses, through training four types of postures, namely, starting, stopping, standing, and walking, to predict pedestrian trajectory movements. However, this method only considers the situation that the pedestrian is always in the same motion state, and ignores the detailed information of the body when the motion state of the pedestrian changes. Kress et al. [36] proposed the use of 3D human poses for trajectory forecasting of vulnerable road users (VRUs), such as pedestrians and cyclists, in road traffic. The 3D poses represent that the entire body posture of the VRUs can provide important posture information for pedestrian trajectory prediction. The above methods can predict the trajectory and movement classification of pedestrians at intersections and crosswalks, but there is still room for improvement in accuracy. Based on B-GPDMs [35], this paper trains a human joint point model with walking and stopping trends, and then couples optical flow information and joint point information to predict the pose information and movement classification of future pedestrians. Although all the above schemes can predict the motion behavior and motion trajectory of pedestrians, the accuracy of the prediction of the motion state and motion trajectory of pedestrians at traffic intersections is limited.

In summary, the optical flow information prediction model can also predict the movement state and trajectory of pedestrians at traffic intersections. But, the optical flow information lacks detailed information about pedestrians in the process of movement, which makes it impossible to accurately predict the spatial position of pedestrians in the future. Therefore, under the premise of considering obstructions, this paper combines optical flow information and human body joint point information to propose a KPOF-Net prediction model to collaboratively predict the motion state and trajectory of pedestrians in the future.

3. Overview of KPOF-Net

3.1. Main Network. Figure 1 summarizes the main framework of KPOF-Net, composed of three main parts of occlusion object removal, pedestrians' state estimation, and trajectory prediction. In occlusion object removal, we use self-supervised learning method of self-flow [19] to detect the optical flow information of pedestrians at intersections and crosswalks, and then propose a Canny-based White Edge Cutting (WEC) algorithm to remove obstructions, and restore the pedestrian posture behind the obstructions by modifying the UCTGAN network. In pedestrians' state estimation, we use Carnegie Mellon University (CMU) [37] to train pedestrians' key point station, e.g., stopping, walking, stopping tendency, and walking tendency. In trajectory prediction, we propose the KPOF-GPDM method, which combines pedestrians' key point and optical flow information to predict pedestrians' future trajectory.

3.2. Pedestrian in Painting behind Occlusion. In actual scenes, pedestrians may be obscured by luggage, handbags, trash cans, stone pillars, animals, and other objects at intersections and crosswalks. In the process of extracting optical flow information, the pedestrian mask cannot be completely obtained. It has great influence on the pedestrian

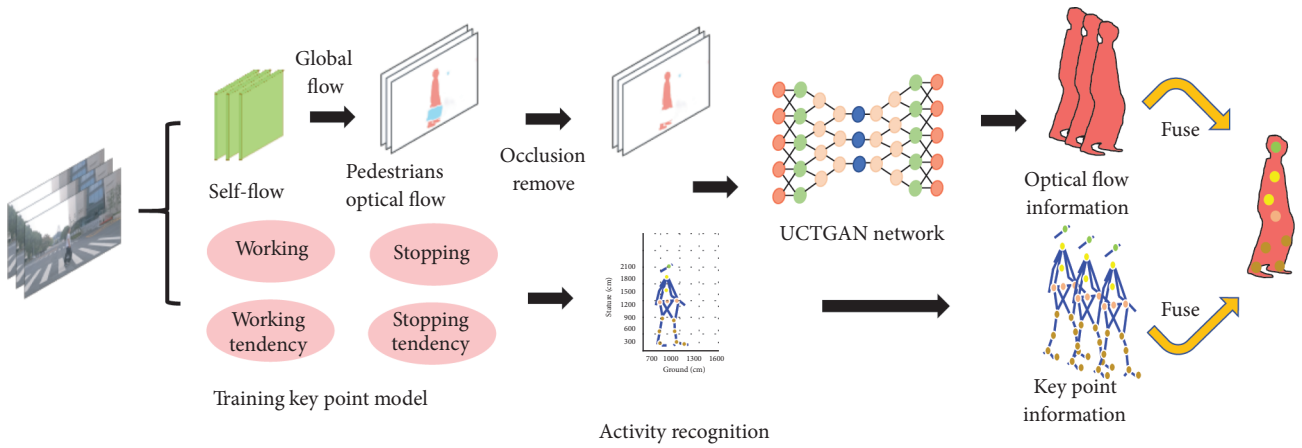


Figure 1: KPOF-Net is composed of an optical flow module and a joint point module. On the one hand, the optical flow extraction of pedestrians and obstructions is completed through self-flow, WEC is proposed to remove obstructions, and then the UCTGAN network is used to generate a complete pedestrian optical flow diagram. On the other hand, it is based on B-GPDM training four types of joint models of walking, stopping, walking tendency, and stopping tendency. Finally, the pedestrian optical flow information is combined with the joint point model to predict the pedestrian's posture and trajectory.

trajectory prediction at the back end, and an accurate trajectory route cannot be obtained. Therefore, we introduce the self-flow network to extract the optical flow information of the pedestrians at the intersection, integrate the WEC to remove the contour information of the obstruction, and then initialize the pedestrian optical flow information after the obstruction through the modified confrontation generation network.

3.2.1. Self-Supervised Learning Method, Self-Flow. Self-flow net is an excellent method to get the objects' optical flow information behind the occlusion. It builds on PWC-Net [38] and extends it to multiframe optical flow estimation. PFC-Net uses pyramid processing to improve the resolution from coarse to thin, and uses feature distortion, cost volume constructs to estimate the optical flow of each layer. Based on these principles, it has achieved state-of-the-art performance with a compact model size.

As shown in Figure 2, the reason why we chose the self-flow network can be seen. First, it uses three images as input to generate multi-frame optical flow estimation of three feature representations of F_{t-1}, F_t, and F_{t+1}. Then, self-flow uses the initial backward flow and backward cost volume information for the previous frame. I_{t-1} can provide effective information about the occlusion, especially the area that is occluded in I_{t+1} but not occluded in I_{t-1}, and self-flow combines this information to get a more accurate optical flow estimation.

At the same time, the self-flow network uses five frames of images as input to perform consistency checks when estimating the optical flow between two frames, thereby inferring the occlusion map between two consecutive images. For the forward-backward consistency check, when the mismatch between the forward flow and the reverse forward flow is too large, the self-flow network considers a pixel to be occluded. A pixel is considered occluded whenever it violates the following constraint:

$$\left|w_{t\longrightarrow t+1}+\hat{w}_{t\longrightarrow t+1}\right|^2<\alpha_1\left(\left|w_{t\longrightarrow t+1}\right|^2+\left|\hat{w}_{t\longrightarrow t+1}\right|^2\right)+\alpha_2, \quad (1)$$

when $\alpha_1 = 0.01$ and $\alpha_2 = 0.05$, we get accurate optical flow information of pedestrians and obstructions.

3.2.2. Removal of Occlusion Region. In the video sequence at intersections and crosswalks, obstruction objects can be divided into two categories as static occlusion (such as stone pillars, trash cans, railings) and dynamic occlusion (such as suitcases, luggage bags, animals). In self-flow network, static occlusion will not generate optical flow for no motion between frames. The dynamic occlusion will produce striking interference optical flow, which is difficult to eliminate. Then, in this part, we only consider the situation when pedestrians are blocked by dynamic objects. A White Edge Cutting (WEC) algorithm is based on Canny, which removes the optical flow information in the occluded area:

$$H_{ij}=\frac{1}{2\pi\sigma^2}e^{\left(-\left(\left((i-(k+1))^2+(j-(k+1))^2\right)/2\sigma^2\right)\right)}, \quad 1\le i,j\le(2k+1), \quad (2)$$

$$\begin{aligned} G&=\sqrt{G_x^2+G_y^2},\\ \theta&=\arctan\left(\frac{G_y}{G_x}\right). \end{aligned} \quad (3)$$

In formulas (2) and (3), H_{ij} represents the Gaussian convolution kernel, (i, j) represents the pixel coordinates, k is the dimension of the convolution kernel, G represents the gradient descent value, G_x and G_y represent the bias value in the (x, y) direction, and θ represents the gradient descent direction.

The pixels in the optical flow image are filtered by Gaussian filter (2) to calculate the wave recorder core to obtain the pixel threshold with weight, then (3) is used to calculate the gradient value and gradient direction, and finally the WEC algorithm is considered to remove the optical

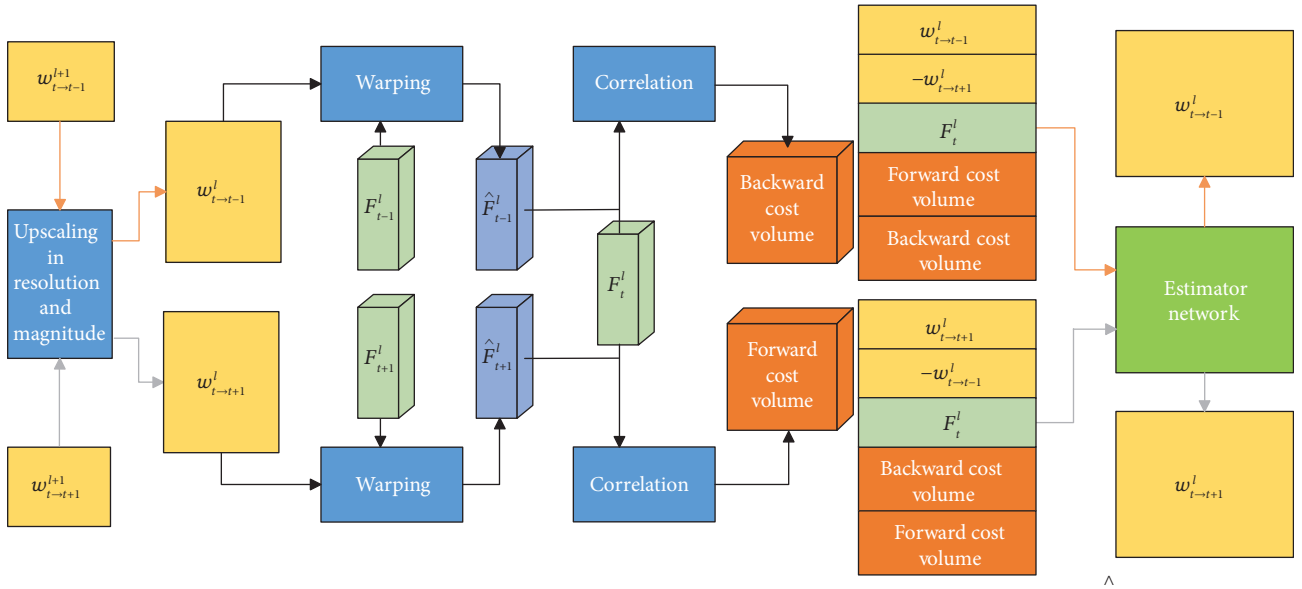


FIGURE 2: Self-flow network architecture at each level (similar to PWC-Net). w^l denotes the initial coarse flow of l and $\hat{F}^l$ denotes the warped feature representation.

flow in the obstructed area information, as the result shown in Figure 3.

In the process of detecting WEC, we take a small portion of the optical flow into the pixel level. In the field of 3×3, we can find that at the junction of two objects, the optical flow information will be close to white or white, so we use the principle to connect the pixel values that tend to be white at the boundary to obtain the exact boundary between the pedestrian and the dynamic object, and remove this part of the area:

$$\text{difference} = \frac{\left|S_{\text{WEC}} - S_{\text{origin}}\right|}{S_{\text{origin}}} * 100\%. \tag{4}$$

We use formula (4) to calculate the percentage of the area difference before and after the occlusion segmentation to show the performance of the WEC segmentation occlusion. In the formula, S_{WEC} represents the area of the occlusion after division, and S_{origin} represents the area of the occlusion before division. From Table 1, it is found that the WEC method can control the area difference within 5%. Although the area difference increases when facing small obstructions, it can still separate different types of obstructions from pedestrians.

3.2.3. Pedestrians Inpainting according to the UCTGAN Network. When pedestrians at the intersections and crosswalks are blocked by static objects, they get incomplete optical flow information. When they are occluded by a dynamic target, the WEC algorithm is used to remove the occluded area to get the pedestrian optical flow map of the incomplete area. In view of the above situation, this paper integrates the GAN network to generate a complete pedestrian optical flow image, as shown in Figure 4.

We simply modified the network based on UCTGAN, and deleted the multi-scale scheme of the original network according to the actual needs of the experiment. The image area of size 256×256 is directly extracted from the external square center of the occlusion area and input to the network. In this way, multiple calculations on different scales of the network can be avoided and the operation efficiency of the network is greatly improved.

The UCTGAN network is trained in an end-to-end fashion, which consists of two branches. The UCTCAN framework mainly includes three network modules: manifold projection module E_1, conditional encoder module E_2, and generation module G. The primary branch consists of a manifold projection module E_1 and a generation module G, which is responsible for learning one-to-one image mapping between two spaces in an unsupervised way by projecting instance image space S_i and conditional completion image space S_i into one common latent manifold space S_m. The second branch consists of a conditional encoder module E_2, which acts as conditional constraint similar to the conditional label. The UCTGAN framework could maximize the conditional log-likelihood of the training instances, which involves a variational lower bound:

$$\begin{aligned}\log p(I_c|I_m) \geq & -KL\left(f_\varphi(Z_c|I_i, I_m) \parallel f_\phi(Z_c|I_m)\right) \\ & + E_{Z_c \sim f_\varphi(Z_c|I_i, I_m)}\left[\log g_\theta(I_c|Z_c, I_m)\right],\end{aligned} \tag{5}$$

where I_c, I_m, and I_i are instance image, masked image, and the repaired image, respectively. Z_c is the latent vector of I_i in space S_m. f_φ, f_ϕ, and g_θ are the posterior sampling function, conditional prior, and likelihood, respectively, where φ, ϕ , and ϑ are the corresponding deep network parameters.

One of the reasons why we choose the UCTGAN network is its series of training loss, including L_{ccl} Condition Constraint Loss, L_{KL} KL Divergence Loss, L_{rec}

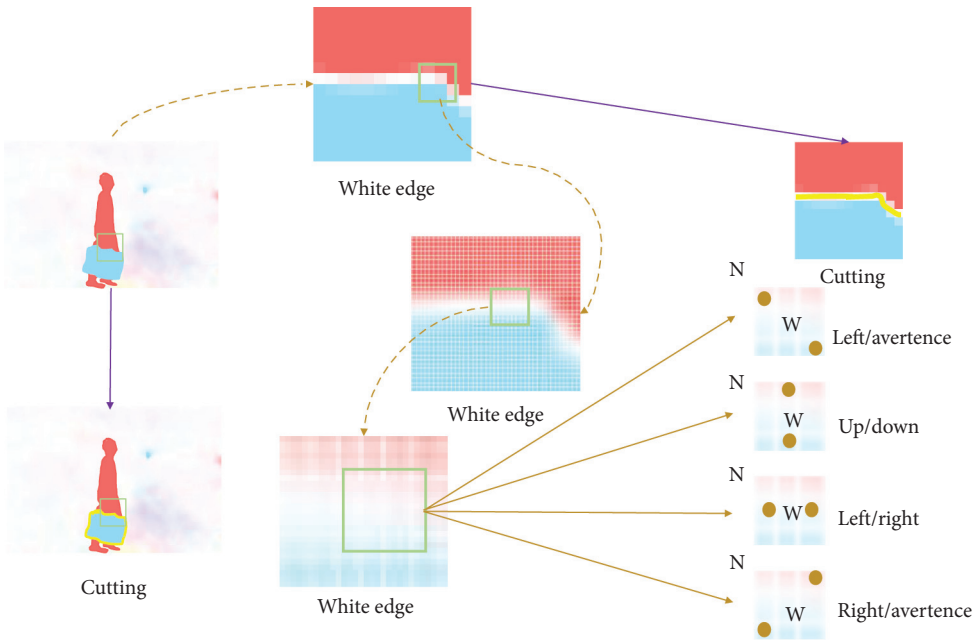


Figure 3: In the WEC detection process, a small part of the optical flow is taken and enlarged to the pixel level. In the $3 * 3$ pixel matrix, there is a transition zone that is close to white at the intersection of the obstruction and the pedestrian in the light flow. Based on this, WEC determines four types of symbols: left/avertence, up/down, left/right, and right/avertence. The N in the figure indicates that the value signs of the corresponding two points are different. When the four pairs of values are opposite in sign and the absolute value of the difference is less than a certain threshold, W is recorded as the WEC boundary. All W points are mapped to the original image to obtain the boundary of the occlusion in the input image.

Table 1: The area value ratio before and after the occluder is moved by WEC.

Object	Trunk (%)	Animal (%)	Handbag (%)	Motorcycle (%)	Bicycle (%)
Difference	2.37	4.32	3.21	3.56	4.77

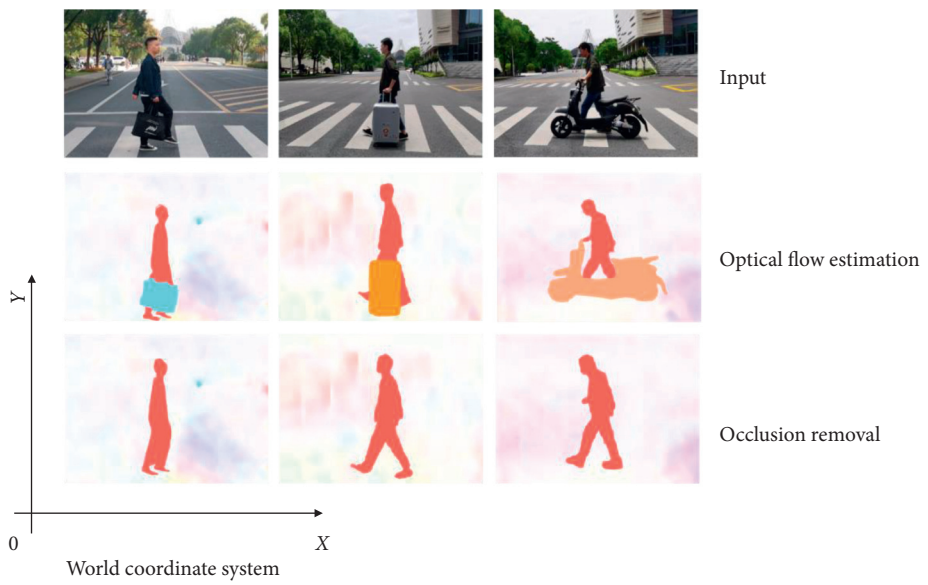


Figure 4: Pedestrian movement spectrogram. In the spatiotemporal sequence, pedestrians with obstructions obtain the initialized pedestrian optical flow diagram through three steps, which include picture input, optical flow estimation, and occlusion removal.

Reconstruction Loss, and L_{adv} Adversarial Loss. The total loss function L_{total} of UCTGAN consists of four groups of component losses, as shown in formula (6). Among them λ_{rec}, λ_{ccl}, λ_{adv}, and λ_{KL} are the hyperparameters corresponding to the constraints, which play a relatively important role in each constraint:

$$L_{\text{total}} = \lambda_{\text{rec}}\left(L^{g}_{\text{rec}} + L^{l}_{\text{rec}}\right) + \lambda_{\text{ccl}}\left(L^{a}_{\text{ccl}} + L^{f}_{\text{ccl}}\right) + \lambda_{\text{adv}}L_{\text{adv}} + \lambda_{KL}\left(L^{i}_{KL} + L^{m}_{KL}\right). \tag{6}$$

The condition constraint losses L^{a}_{ccl} and L^{f}_{ccl} encourage consistency and integrity between completion contents and known contents, reconstruction losses L^{g}_{rec}, L^{l}_{rec}, and L^{l}_{rec} encourage one-to-one mapping between the instance image and the repaired image, and avoid falling into mode collapse, and adversarial loss L_{adv} makes repaired images fit in with the distribution of the training dataset. The loss of all these prompted us to get excellent pedestrian optical information. If you are interested in the specific details of how to inpainting pedestrian optical flow information after proposing dynamic objects in this article, you can do intensive reading UCTGAN network [21].

3.3. Pedestrians' Key Point Model Establishment. In actual scenes, at intersections and crosswalks, pedestrians will produce corresponding stance based on the current status of traffic lights, traffic flow, and their own consciousness. This posture information can help us predict the state of pedestrians in the future. Therefore, in this paper, we extract human joint point information based on the hidden Markov joint point recognition model in Minguez et al. [35], and then train a joint point model that adapts to the human posture in the intersection and crosswalk scenes, including stopping, walking, stopping tendency, and walking tendency.

3.3.1. Data Set Description. In this section, our main goal was to train accurate models with different pedestrian dynamics. For this, we used high-frequency, low-noise datasets released by Carnegie Mellon University (CMU) [39]. On the one side, the high frequency of the dataset helps the algorithms to properly learn the dynamics of different activities and increases the probability of finding a similar test observation in the trained data without missing intermediate observations. On the other side, low-noise models improve the prediction when working with noisy test samples. The pedestrian motion simulation dataset contains a typical pedestrian motion sequence package. Among this, we collect the three-dimensional coordinates of 41 joints along the body with a frequency of 120 Hz. However, according to the actual situation of intersections and crosswalks, we focused on using part of the joint point information of the legs and the body. At the same time, according to our actual needs, we selected four categories from the CMU dataset that meets stopping, walking, stopping tendency, and walking tendency sequence; a total of 200 sequences were extracted, which consisted of 143,207 pedestrian poses from 25 different subjects. See Table 2 for details.

Pedestrian skeleton estimation algorithm, based on point clouds extracted from a stereo pair and geometrical constraints, was implemented to test the proposed method with noisy observations. The algorithm is based on references [40, 41], and the specific details can be learned from the literature [42].

3.3.2. Learning Pedestrians' Key Point Model. After extracting the human body joint point information, we need to identify the human body joint point model of the corresponding joint point state in the intersection and crosswalk scenes. In this part, we use the B-GPDMs method in the literature [35] to identify the joint points of pedestrians.

Mínguez et al. just trained four models suitable for their experimental needs, including walking, starting, stopping, and standing. However, these four models are only limited to the joint point model in which the pedestrian has been in an upcoming motion state, and did not consider the detailed information of the joint point when the pedestrian's motion state changes, resulting in the system being unable to predict accurate pedestrian trajectory information. So, we use the B-GPDMs algorithm to train four types of joint point models: standing, walking, stopping tendency, and walking tendency. This is because when pedestrians pass crosswalks and intersections, their movement is not restricted to only two states of walking and stopping alone, as they will constantly judge the current traffic situation to change their own motion state. When there is a vehicle in front, pedestrians in the walking state will collect forward environment information in real time by leaning their upper limbs forward, and the distance between the lower limbs will continue to shrink. When the front is passable, the upper limbs also lean forward to collect the front environmental information in real time, while the distance between the lower limbs is increasing.

In the process of learning all the sequences contained in the CMU dataset, since the coordinate system of these sequences is affected by the sensor, we deleted the 3D data of each observation and obtained the coordinate system with the pedestrian as the origin, which allows us to deal with pedestrians in any location. Then, by subtracting the mean and dividing each mean by the standard deviation to scale, it is more convenient to obtain zero mean and unit variance data. Since B-GPDM needs to use the smallest posterior function to iterate, we give appropriate initialization potential positions, hyperparameters, and constants according to the literature [30]. We initialize the potential coordinates through PCA [43], the kernel parameters and the corresponding values in the constants, and finally used the dataset in Table 2 to learn the four types of human joint point models suitable for intersections and crosswalks in Figure 5.

3.4. Pedestrian Path Prediction by KPOF-GPDM. Pedestrians are prone to wandering at intersections. At crosswalks, they may also stand or walk due to the status of traffic lights and vehicle driving on the road. However, the use of optical flow information alone can lead to the loss of some posture information of the walking or stopping state. Therefore, we introduce the human joint point information into the optical flow information prediction algorithm to form the KPOF-GPDM algorithm in this article. The algorithm supplements the detailed information of the pedestrian's posture during the movement and can more accurately predict the pedestrian's movement state and movement trajectory in the future.

Table 2: Pedestrian data sequences.

	Sequence				
Orientation	Stopping	Walking	Stopping tendency	Walking tendency	Total
Left to right	18	15	32	36	101
Right to left	15	17	37	30	99
Total	33	32	69	66	200

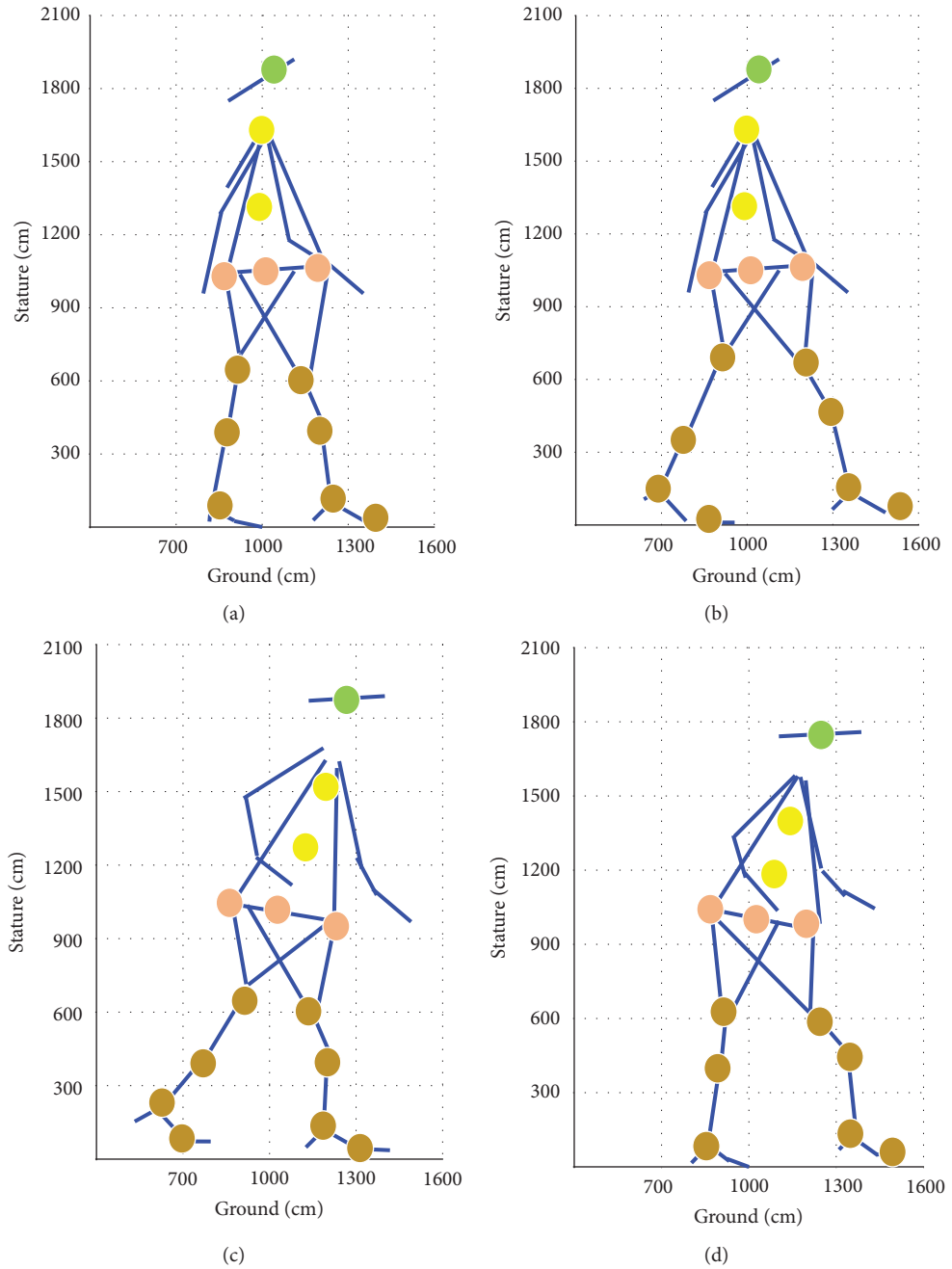


Figure 5: Four types of human joint point models are shown. When the pedestrian is in the stopping state, the whole body is in a vertical posture, and the direct distance between the legs is 0–30 cm. When the pedestrian is in the walking state, the whole body is basically in a vertical posture, and the distance between the feet is 50–70 cm. When a pedestrian has a change in the state of motion, the upper limbs will lean forward, but the distance between the pedestrian's legs in (c) is 40–60 cm, and the distance between the legs in (d) is about 0–30 cm. (a) Stopping. (b) Walking. (c) Stopping tendency. (d) Walking tendency.

KPOF-GPDM integrates pedestrian optical flow characteristics and joint point information to predict the pedestrian's lateral motion state at crosswalks and intersections and the motion trajectory in world coordinates. Firstly, we construct the pedestrian's lateral attitude information in the world coordinate system. Then in the same world coordinate system, the pedestrian's movement state at the intersection and crosswalk, the detailed information of the upper and lower limbs are extracted, and the pedestrian optical flow information and the joint point information are merged. Secondly, the underlying spatial dynamic model GPDM is used to reduce the dimension of the feature information. Finally, trajectory prediction and motion feature reconstruction are performed in low dimensional space.

3.4.1. The Lateral Position of Pedestrians in World Coordinates. We need pedestrian distance information when we establish the mapping between pedestrian dynamic optical flow features and real pedestrian speed. In monocular ranging model, we assume that the road surface is flat and pedestrians walk upright. In this process, we need to calibrate the camera's internal and external parameters. The camera's internal parameters are fixed. The camera's height and pitch angle will remain unchanged once the camera is fixed on the vehicle. Based on the above premises, the world coordinate system can be established. The projection of the camera's optical axis on the ground is the center, the direction of the vehicle is X axis, and Y axis is perpendicular to the ground.

We regard the center of the pedestrian's projection on the ground as the ranging point. Correspondingly, we use the projection of the pedestrian's abdominal transverse center on the bottom of the pedestrian mask to calculate the ranging points $p(u, v)$. Among them, u is the maximum value in the Y direction of the mask area, and v is the average value of the x-axis in the mask area:

$$\begin{aligned} u &= \max\left(y_{\text{mask}}\right), \\ v &= \text{mean}\left(x_{\text{mask}}\right). \end{aligned} \tag{7}$$

According to reference [44], the focal length of the camera is f and the ranging point is $p(u, v)$. Then, the world coordinates of pedestrians can be obtained as follows:

$$\begin{aligned} Y &= \frac{H}{\tan\left(\alpha - \arctan\left(\left(v - v_0\right)/f_y\right)\right)}, \\ X &= \frac{\left(u - u_0\right) \times H}{\sqrt{f_x^2 + \left(v - v_0\right)^2} \times \sin\left(\alpha - \arctan\left(\left(v - v_0\right)/f_y\right)\right)}. \end{aligned} \tag{8}$$

With the angle between the optical axis and the horizontal road surface α, the height between the camera and the ground is H. $f_y = f/d_y$; $f_x = f/d_x$; and d_x and d_y represent the pixels distance in image coordinates u and v. Then, the distance between pedestrians and vehicle is computed using $D(P) = \sqrt{X^2 + Y^2}$.

3.4.2. Key Point Feature Fuse. In the fusion module of optical flow information and connection point model, firstly, pedestrian horizontal optical flow is the transverse component of dynamic optical flow. The pedestrian mask area obtained by self-flow can locate the pedestrian position and obtain more accurate optical flow characteristics. At the same time, complete pedestrian mask can be repaired through the UCTGAN network. Simultaneously, the pedestrian connection point model we trained in Section 3.3 is used to identify the pedestrian connection points in the image sequence. Finally, in the same world coordinate system, pedestrian optical flow information and human body node information are fused; the specific process is shown in Figure 6.

When only using optical flow information to predict pedestrian trajectories, the predicted pedestrian optical flow velocity V_{of} can help the ADAS to predict the motion trajectory of the behavior in the future to a certain extent. But, at this time, V_{of} reflects the overall speed of a pedestrian, while the detailed information of the upper and lower limbs of the pedestrian in the process of walking and parking is lost, and the pedestrian track with higher accuracy cannot be obtained in the future. Therefore, this article introduces the human body joint point information to form a trajectory prediction model that combines optical flow information with human optical nodes. When pedestrians pass through an intersection or crosswalk, they will not only collect current traffic information in real time by leaning forward but also adjust the movements of their upper and lower limbs to reflect changes in their own movement status. For example, when there are vehicles ahead, the upper limbs of the pedestrian in the walking state will lean forward to judge the forward traffic situation, the lateral velocity V_{Tkp_i} of the joint points of the upper limb of the human body will gradually slow down. The lateral spacing between the joint points of the lower limbs gradually decreases, the lateral velocity $V_{Dk\ p_i}$ gradually decreases. At this time, the pedestrian movement status changes from walking to stopping. When the front traffic condition is good, the upper limbs of pedestrians will also lean forward to judge the current traffic situation, and the lateral velocity V_{Tkp_i} of the joint points of the upper limbs will increase positively. The distance between the joint points of the lower limbs and the lateral velocity are also gradually increasing. At this time, the pedestrian motion state shows a trend of gradually changing from the stopping state to the walking state.

In the traditional optical flow trajectory prediction, the car's ADAS only uses V_{of} in equation (9) to calculate the pedestrian's speed on road, and then roughly predicts the pedestrian's motion state and trajectory in a Gaussian low-dimensional space, unable to provide effective anti-collision data. After introducing the information of human joints, the ADAS uses the detailed information of the joints of the upper and lower limbs of the human body to estimate the trend of pedestrian movement in the future. It also uses formula (10) to calculate the speed of the upper and lower limbs of the pedestrian, which reflects the detailed information of the pedestrian when facing different traffic con-

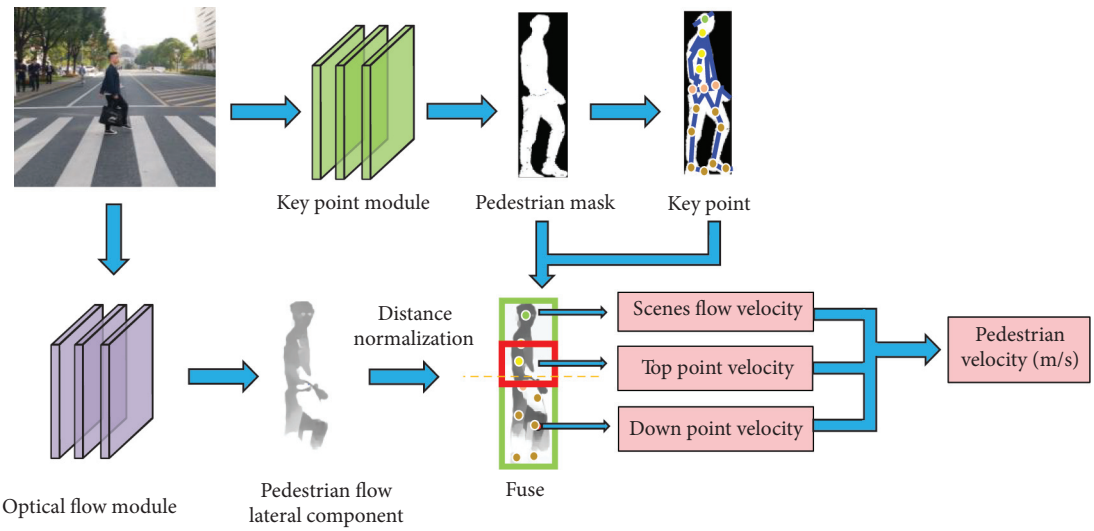


Figure 6: The fusion process of optical flow information and node model can improve the accuracy of pedestrian trajectory prediction through the fusion of pedestrian nodes with different motion states and different postures.

ditions, and provides more accurate data input for the back-end motion state and trajectory prediction. At this time, the three calculation data of V_{Tkp_i}, $V_{Dk\ p_i}$, and $V_X(p)$ are used to predict the pedestrian's motion state, and the trajectory prediction can better reflect the real pedestrian state and provide more effective active safety data for the car's ADAS.

With the pedestrian's lateral optical flow V_{of}, lateral velocity of key point V_{Tkp_i} and $V_{Dk\ p_i}$, $(V_{Tkp_i}, V_{Dk\ p_i} \in V_{kp_i})$, and the camera interval of each frame Δt, the pedestrians' walking speed $V_X(p)$ and X_t can be calculated by the following formula:

$$V_X(p) = \frac{\tau \cdot V_{of}(p)}{D \cdot \Delta t}, \tag{9}$$

$$V_{T/Dk\ p_i} = \frac{1}{n} \cdot \frac{\tau \cdot \sum_{i=0}^{n}\left(V_{kp_i}\right)}{D \cdot \Delta t}, \tag{10}$$

$$V_X(p) = \frac{1}{(n+1)} \frac{\tau \cdot \left(V_{of}(p) + \sum_{i=0}^{n}\left(V_{kp_i}\right)\right)}{D \cdot \Delta t}, \tag{11}$$

where $V_X(p)$ is the velocity of pedestrians, the function D represents pedestrian distance, and τ is a constant. The average value of velocity optical flow in pedestrian upper body can be regarded as the average speed of the pedestrian $\overline{v}$. We resize $V_X(p)$ to 32×16 pixel. We construct a feature vector $y_t \in \mathbb{R}^D$, $(D = 515)$ that includes position, average speed, and velocity optical flow.

3.4.3. Gaussian Model. GPDM is a latent variable model. It established the mapping relation from a latent space x_t to observation space y_t and a latent dynamical model which account for the temporal dependence on pedestrian motion features [39].

The observation space $Y = [y_1, y_2, \ldots, y_N]^T$ is N frames motion feature vector. $X = [x_1, x_2, \ldots, x_N]^T$ is the dynamic mapping on latent positions. The mapping relation can be described:

$$p(Y|X, \overline{\beta}, W) = \frac{|W|^N}{\sqrt{(2\pi)^{N\times 16}|K_Y|^{16}}} \exp\left(-\frac{1}{2} T_r\left(K_Y^{-1} Y W^2 Y^T\right)\right), \tag{12}$$

where K_Y is a kernel matrix of size $N \times N$ constructed by the kernel function κ_Y. The parameter of the kernel matrix is $\overline{\beta} = \{\beta_1, \beta_2, \beta_3\}$. For our data, we use the RBF (radial basis function) kernel $\kappa_Y(x, x') = \beta_1 \exp(-(\beta_2/2)\|x - x'\|^2) + \beta_3^{-1}\delta_{x,x'}$; W is a $D \times D$ diagonal matrix that represents the weight of different dimensions of y_t. Assuming that the dynamics of the data in the latent space x_t satisfies the first-order Markov model, the dynamics of the time series data is incorporated using

$$p(X|\overline{\alpha}) = \frac{p(x_1)}{\sqrt{(2\pi)^{(N-1)\times d}|K_X|^d}} \exp\left(-\frac{1}{2} Tr\left(K_X^{-1} X_{2:N} X_{2:N}^T\right)\right), \tag{13}$$

where $X_{2:N} = [x_2, \ldots, x_N]^T$, the kernel matrix K_X is $(N-1) \times (N-1)$ that is constructed from $X_{1:N-1} = [x_1, \ldots, x_{N-1}]^T$ and defined by a kernel function $\kappa_X(x, x')$. We use RBF and a linear kernel in the kernel function with kernel hyperparameters $\overline{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$:

$$\kappa_X(x, x') = \alpha_1 \exp\left(-\frac{\alpha_2}{2}\|x - x'\|^2\right) + \alpha_3 x^T x' \alpha_4^{-1} \delta_{x,x'}. \tag{14}$$

Latent mapping and latent dynamics model combined with time series observations:

$$p(X, Y, \overline{\alpha}, \overline{\beta}, W) = p(Y|X, \overline{\beta}, W) p(X|\overline{\alpha}) p(\overline{\alpha}) p(\overline{\beta}) p(W). \tag{15}$$

The process of GPDM inference is finding hidden space variables X and kernel parameters $\{\overline{\alpha}, \overline{\beta}\}$ by minimizing the negative logarithm joint posterior $-\ln p(X, \overline{\alpha}, \overline{\beta}|Y)$. It can be optimized by the scaled conjugate gradient (SCD) algorithm. The dimension of latent space $d = 3$. Figure 7 illustrates this

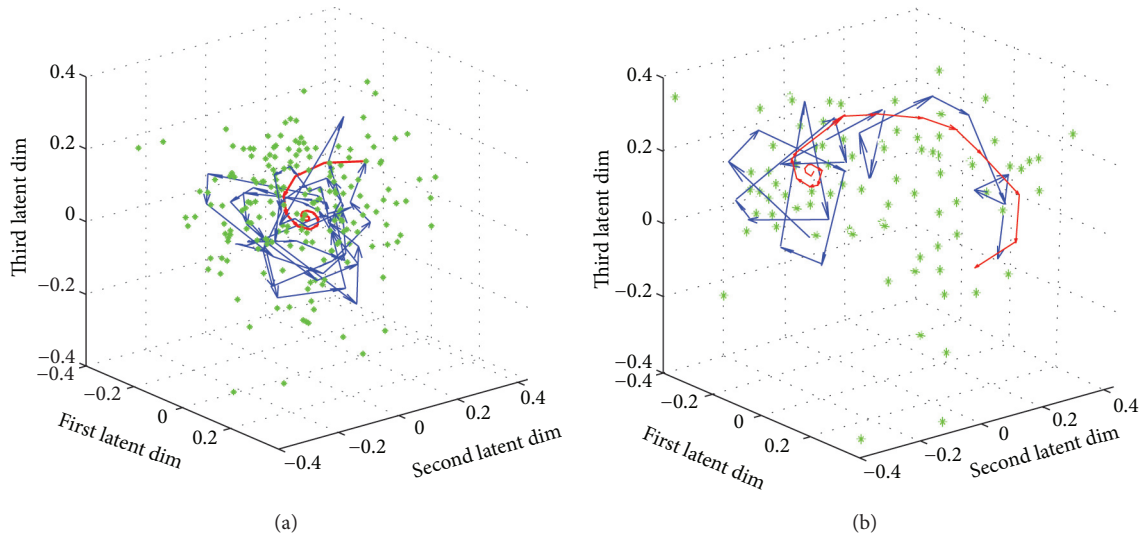


FIGURE 7: (a) The hidden space 3D trajectory obtained by walking data training and (b) hidden space 3D trajectory obtained by stopping data training. The green lines represent the trajectories of pedestrian walking and stopping features under hidden space projection, respectively, and red represents the average predicted trajectory learned by KPOF-GPDM.

mean prediction of a point for several frames on the low-dimensional space.

The motion state of the pedestrian at time t is described by $\phi_t = [x_t, X_t]$, where $x_t \in \Re^d$ is a point in the low-dimensional space and X_t is the horizontal position of the pedestrian in practice. Given an observed motion feature y_t and observed lateral position Y_t, the probability of a pedestrian state ϕ_t is computed by

$$p(\phi_t|y_t, Y_t) = \eta p(y_t, Y_t|\phi_t) \int p(\phi_t|\phi_{t-1}) p(\phi_{t-1}|y_{t-1}, Y_{t-1}) d_{\phi_{t-1}}, \tag{16}$$

with normalization constant η. The probability $p(\phi_t|\phi_{t-1})$ of observing a future state is computed from the GPDM latent space mean prediction.

3.4.4. Motion Feature Reconstruction. KPOF-GPDM can be obtained by Bayesian law which can generate new observation sequences. With the trained model $\Gamma = \{Y, X, \overline{\alpha}, \overline{\beta}, W\}$, a new observation sequence and the joint conditional distribution of the scene stream feature corresponding to the hidden space feature sequence is expressed as

$$p(Y^*, X^*|\Gamma) = p(Y^*|X^*, \Gamma) p(X^*|\Gamma). \tag{17}$$

The new latent variable sequence x^* can be predicted by maximizing the formula (16). The process of predicting a new sequence by the first latent variable X_1 requires two steps:

(a) A new latent space variable is predicted based on the data at the previous time.

$$\mu_X(x^*) = k_X(x_{t-1})^T K_X^{-1} X_{2:N}, \tag{18}$$

where the vector $k_X(x)$ is containing $k_X(x, x_i)$ in the ith entry, and x_i is the ith training vector.

(b) The new data in the observation space is constructed using

$$\mu_Y(x^*) = k_Y(x^*)^{\mathrm{T}} K_Y^{-1} Y. \tag{19}$$

Figure 8 shows the reconstructed scene flow when pedestrians cross the road, and Figure 9 shows the pedestrian velocity in the future.

4. Experiment

We deploy our system on PX2 mobile devices, using its TensorRT neural network inference engine and cuDNN deep neural network library to improve its real-time performance. During the experiment, on the one hand, our system and four excellent pedestrian trajectory prediction systems (KF, IMM-KF, HoM/Traj [3], and SFlowX/GPDM [30]) are placed in the same video sequence to compare their pedestrian trajectory position prediction accuracy and pedestrian action classification probability accuracy. On the other hand, our fusion model is compared with the prediction model using only one piece of information to verify the improvement in our system performance. In this section, we use a monocular camera (baseline 33 cm, 30 fps) mounted on the inside of the windshield and behind the rearview mirror to collect video data at a busy intersection in the campus. Video data are divided into two scenarios, both of which are pedestrians crossing the road on the crosswalk without being covered. In the first scene, Figure 10(a), when pedestrians stop at the side of the road, they observe the traffic flow on the road and decide whether to stay in place or walk through the crosswalk for the next stage of action. In the

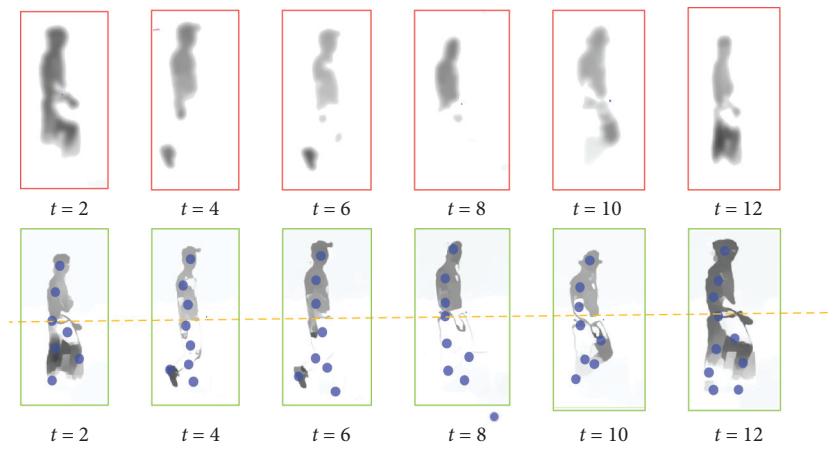


Figure 8: (Top row) Reconstructed optical flow based on current state ($t = 0$) and state predictions ($t = 2, \ldots, 12$) in low-dimensional latent space. (Bottom row) Optical flow that is (will be) actually measured at the corresponding time steps.

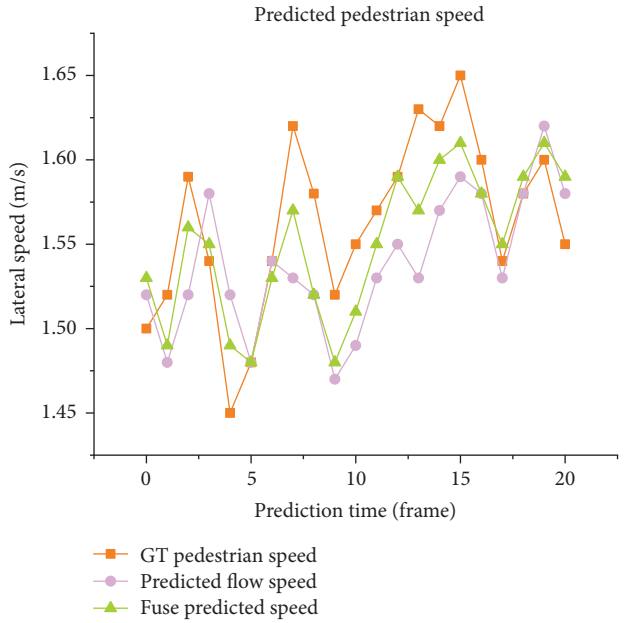


Figure 9: The comparison between using only optical flow and using the fusion model to predict pedestrian speed and the true value shows that KPOF-Net can predict pedestrian speeds that are closer to the true value.

(a) (b)

Figure 10: Campus experiment scene. (a) Pedestrians stand by the roadside. (b) Pedestrians pass the crosswalk.

second scene, Figure 10(b), when pedestrians pass the crosswalk, they judge the traffic situation in real time and change their own motion state.

In order to evaluate the performance of the pedestrian trajectory prediction and action classification in this paper, we first evaluate the performance of the pedestrian motion

classification model through experiments. Through manual annotation of pedestrians in the images of the shape, we obtained the coordinates of the ground truth (GT) of pedestrians in the world coordinate system. Since pedestrian basically walk upright, set the pedestrian's standing point as the center origin instead of the pedestrian's center of gravity, so that we can obtain the horizontal and vertical coordinates of the pedestrian on the ground in order to obtain more reliable basic data. Due to the displacement, it is not reliable to show an action classification based on starting and stopping (with displacement of starting and stopping), and we use the key information to get body posture. At the beginning, the legs are in a separate position, and at the stop, the legs shift from closed to walking, and the arms and legs swing alternately. By combining posture and displacement, we can better classify movement types. In the experiment, the tester must choose to stop or cross the road, and the probability of pedestrian action classification is expressed by the values of [0,1], and the probability value is calculated by formula (16). In terms of alignment along the time axis, for each trajectory in which the pedestrian is stopping, the moment of the last placement of the foot is labeled as the stopping moment. The time-to-stop (TTS) value is used to count the number of frames before the event, meaning that the TTS value for the frames before the stop event is positive and the TTS value for the frames after the stop event is negative. In sequences in which the pedestrian continues walking, the closest point to the curbstone (with closed legs) is labeled. Analogous to the TTS definition, the latter is called the time-to-curb (TTC) value. We assume that $TTC = 0$ represents the time when pedestrians change from walking to stopping; when the TTC is bigger than 0, it represents the previous frames of stopping, and when the TTC is less than 0, it represents the frames after stopping.

4.1. Pedestrian Action Classification. We compare the predicted probabilities of the system with three excellent systems in the same video sequence set for its action classification, and judge whether pedestrians will stop or walk in a short time in the future. When pedestrians are at intersections or crosswalks, they will choose to stand or walk because of the current status of traffic lights, road traffic, and their own decision-making awareness. Figure 11 shows the predicted probabilities of whether the five systems are walking or stopping in the future in the same sequence set.

In order to fully test the performance of the five systems for pedestrian motion classification, we tested multiple video datasets on campus. Pedestrians wandered at intersections and choose to stop at the crosswalk due to road traffic conditions. When the current safe traffic environment is determined, they resumed walking and other movements. For each test sequence, we used a slider between 0 and 1 to provide probability (confidence), which is displayed in the most intuitive way.

In Figure 11, we can see that when the car is in motion, it has an impact on the predicted probability of our state change. When the pedestrian state is about to change, the prediction probability of the pedestrian state change in the dynamic scene of the vehicle is lower than that in the static scene, and the prediction ability is reduced. However, our system has the least decline range and the least impact. At the same time, regardless of whether the car is in a stopping or moving state, when the pedestrian state changes (from walking state to stopping state or from stopping state to walking), each system starts with a low probability, and predicts the probability gradually as the pedestrian state changes increase. However, whether the car is in motion or stopped, it can be clearly seen that our system is more sensitive than other systems, reacts more quickly, and can keenly grasp the characteristics of the human body when the pedestrian's state changes, so as to predict the change of the pedestrian's state more quickly.

In the classification and discrimination of pedestrian movement, since our system has learned four types of human joint point models in Section 3.3, it not only includes two basic models of standing and walking but also two pedestrian joint point models of standing tendency and walking tendency. When pedestrians are about to stop or walk at intersections or crosswalks, they will judge the traffic situation ahead by leaning forward to prepare for changes in their own state. As a result, we can capture more human body posture information, prompting our system to achieve better prediction results than other systems in experiments.

At each moment of the input trajectory, we determine the category membership degree by estimating the stopping probability through the threshold, adjusted the parameters through experiments, and set the appropriate threshold. When the probability of our state change is greater than the threshold, we determine that our pedestrian state is walking or stopping. In our experimental results, when the car is in a stopped state, there is a probability of 0.402 to predict the future state of the pedestrian 7 frames before the state change.

4.2. Pedestrian Trajectory Prediction. We also attach importance to the system's ability to predict the accuracy of pedestrian location. Accurate location information can establish an excellent pedestrian prediction model and provide auxiliary information for ADAS functions. During the test, we considered that the state of the car's movement also affects the ADAS's prediction of the change of the pedestrian's movement state. Therefore, in Table 3, we collected video sequences of the car in different states of movement. We compared the positioning accuracy between systems by the average value and standard deviation of the RMSE of each video sequence. The range of pedestrian frames is [−20, 15], when frame 0 means the manually labeled TTS/TTC moment. The position between [0,15] is predicted by the system, which represents the comparison between our positioning accuracy. It can be seen from Table 4 that when the pedestrian is in motion, all systems can capture enough pedestrian posture information, and obtain a smaller posture prediction error compared to the pedestrian stopped state. At the same time, no matter whether the pedestrian is walking or stopping, our system can extract more pedestrian pose information by fusing optical flow information and

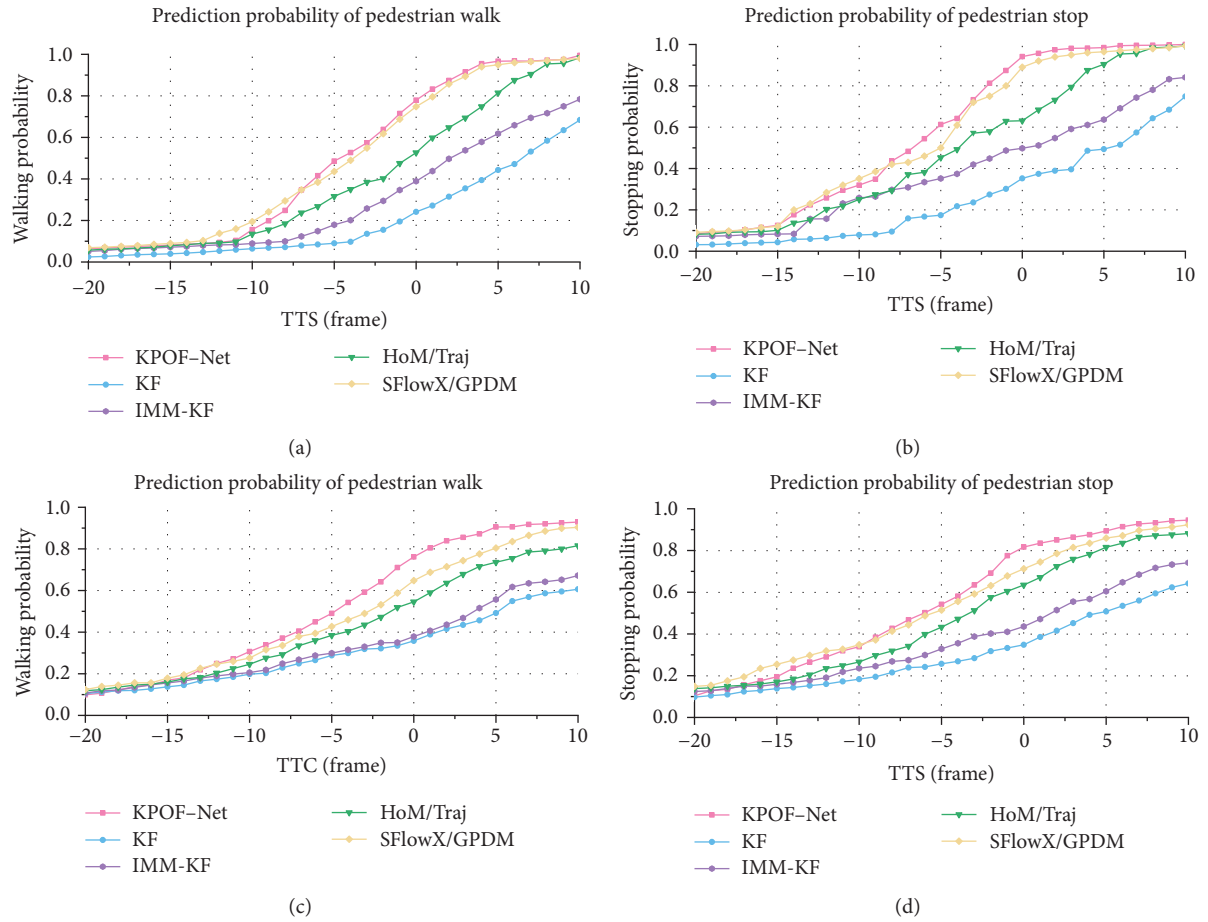


Figure 11: (a) Prediction probability of pedestrian walk. (b) Prediction probability of pedestrian stop. (c) Prediction probability of pedestrian walk. (d) Prediction probability of pedestrian stop. (a, b) The car is in a stopped state and is stopped in front of the sidewalk, and the probability of the state change at the pedestrian intersection and sidewalk is measured. (c, d) The car is in a state of motion, and when it is gradually approaching the sidewalk, we predict the probability of the pedestrian's state change.

Table 3: Data source method.

Sequence	Vehicle standing	Vehicle moving	Vehicle standing + moving
Ped. Stopping	11	5	16
Ped. Walking	9	4	13

Table 4: Pedestrian lateral trajectory prediction error.

Systems		State			
		Walking		Stopping	
		0	15	0	15
KF	Mean	0.28	0.62	0.43	1.27
	±Std	0.05	0.25	0.09	0.24
IMM-KF	Mean	0.34	0.58	0.62	1.15
	±Std	0.06	0.34	0.15	0.31
HoM/Traj	Mean	0.22	0.43	0.31	0.82
	±Std	0.03	0.13	0.09	0.24
SFlow/GPDM	Mean	0.17	0.51	0.37	0.54
	±Std	0.06	0.27	0.08	0.18
KPOF-Net	Mean	0.15	0.38	0.27	0.42
	0 ± Std	0.04	0.13	0.05	0.14

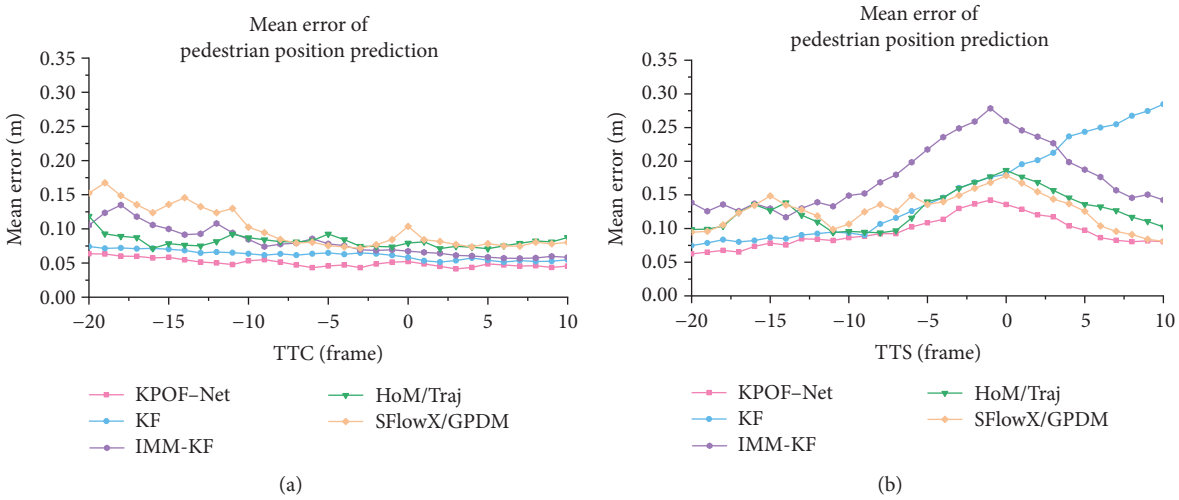


Figure 12: (a) Graph shows the performance of KPOF-Net's pedestrian trajectory prediction accuracy. (b) Curve shows that KPOF-Net is better than others when the state changes. (a, b) The cars are slowly approaching the crosswalk. An excellent system has a faster response speed and can handle the trajectory prediction error caused by the state change in time.

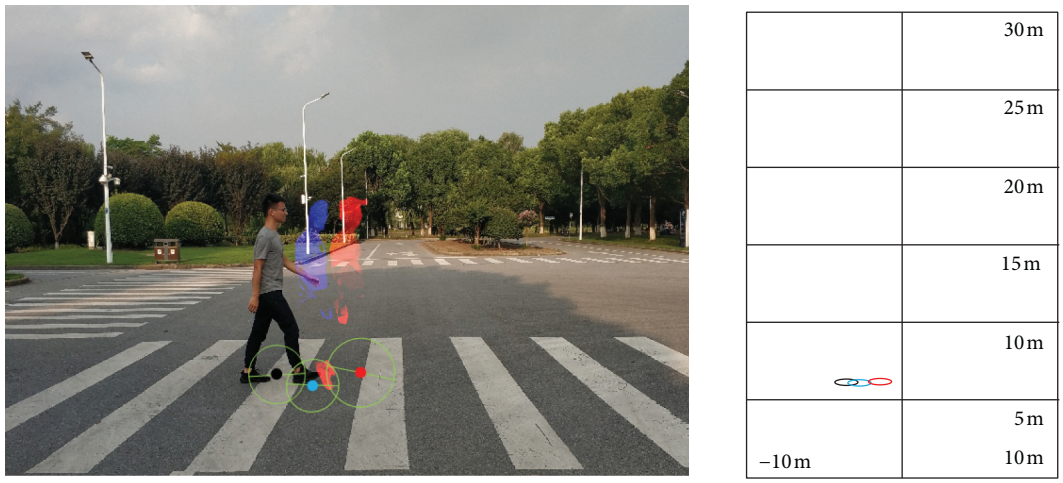


Figure 13: The trajectory prediction model at 0.25 s and 0.5 s when traveling without occlusion.

Table 5: Euclidean distance error analysis.

Metric	Datasets	Point	Flow	Point + flow
Average displacement error (m)	Video 1	0.1832	0.2418	0.1047
	Video 2	0.2463	0.1716	0.0781
	Video 3	0.2314	0.2159	0.0948
Final displacement error (m)	Video 1	0.1657	0.2314	0.1164
	Video 2	0.2591	0.1843	0.0841
	Video 3	0.2534	0.2317	0.0973

joint point information, which enables us to predict the position of pedestrians more accurately. Also, it can be found from Figure 12 that when pedestrians are always in the same state of motion, KPOF-Net shows better prediction performance than other excellent systems, and can always maintain a low pedestrian position prediction error. When the pedestrian motion status changes, with a strong anti-interference ability, our KPOF-Net can suppress the increase in pedestrian position prediction error caused by the motion status change, which is faster than other systems.

4.3. Pedestrian Prediction Model. After removing dynamic object occlusions and fusing the repaired optical flow information with the human joint point model, our system has improved pedestrian action classification and pedestrian trajectory prediction compared with several other excellent systems with excellent experimental results. At the same time, in order to verify the performance improvement of the prediction model that integrates optical flow information and joint point information, we run the KPOF-Net model, single optical flow information prediction model, and joint point prediction model in pedestrian unobstructed video sequences. In this experiment, consider the system's real-time performance and trajectory prediction accuracy on

common hardware devices. On the one hand, the judgment logic we set in the algorithm will skip the WEC link directly when the pedestrian is in an unobstructed state. On the other hand, the experiment in this section is carried out at a low vehicle speed of 15 km/h. And, we propose Average Displacement Error (ADE) and Final Displacement Error (FDE) evaluation rules to verify the performance improvement of our system.

The results of prediction experiments are shown in Figure 13, in which the blue optical flow is the predicted pedestrian position after 0.25 s and the red optical flow is the predicted pedestrian position after 0.5 s.

Next, we make path prediction and select the most accurate activity model to estimate the future state of pedestrians, and two error indexes are used to evaluate the overall position prediction. The comparison results are shown in Table 5:

(1) Average displacement error (ADE): the average Euclidean distance between the predicted location and the actual location over time.

(2) Final displacement error (FDE): the Euclidean distance between the Final predicted location and the real point on the ground.

By comparing the ADE and FDE of the three model systems in multiple video sequences, we can find that when only the optical flow information and joint point information are used to predict the pedestrian trajectory, the ADE and FDE values fluctuate around 0.2 m. There is still a large deviation between the position information predicted by using only the optical flow information or optical node information and the actual position of the pedestrian. The prediction accuracy of KPOF-Net can be set to reach 0.1 m, which can effectively predict the trajectory of pedestrians in the future.

5. Conclusion

At intersections and crosswalks, in the optical flow module, we use the self-flow network to obtain pedestrian optical flow information containing obstructed objects, and then propose the WEC algorithm to segment the obstructed objects from pedestrians, and finally use the UCTGAN Network to restore the pedestrian optical flow image. In the human body joint point module, four human body joint point models are trained: standing, stopping, walking tendency, and stopping tendency. After completing the optical flow module and the human optical node module, we merge the two modules to form a KPOF-Net network for pedestrian trajectory prediction. The network supplements the detailed information of the body movement of pedestrians when passing intersections and crosswalks in the optical flow information of the human body. At the same time, we compare the KPOF-Net system with KF, IMM-KF, HoM/Traj, and SFlowX/GPDM, four excellent pedestrian trajectory prediction systems. Experiments show that after our KPOF-Net system integrates optical flow information and human body joint point information, both the probability of pedestrian state prediction and the accuracy of pedestrian trajectory are improved. Even in the time when the pedestrian state changes, the prediction accuracy of pedestrian estimation fluctuates greatly, and it can respond quickly, restrain the increase of the error, and restore the accuracy to a normal value. It can be found from the full text that the KPOF-Net prediction model after fusion of human joint point information could provide accurate auxiliary information for our advanced driving assistance system.

Data Availability

Some relevant data are available on the website https://github.com/604627144/KPOF-GPDM, wherein some codes, pedestrian data, and experimental results will be presented.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Chen, F. Wang, S. Qu et al., "Pseudo-image and sparse points: vehicle detection with 2D LiDAR revisited by deep learning-based methods," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13. In press, 2020.

[2] Y. Liu, G. Chen, and A. Knoll, "Globally optimal vertical direction estimation in Atlanta World," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, In press.

[3] Mercedes-Benz E-Class, *Passenger car yearbook*, pp. 117-118, SAE, Warrendale, PA, USA, 2014.

[4] A. Dosovitskiy, P. Fischer, I. Eddy et al., "Flownet: learning optical flow with convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[5] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[6] T.-W. Hui, X. Tang, and C. Change Loy, "LiteFlowNet: a lightweight convolutional neural network for optical flow estimation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR*, Salt Lake City, UT, USA, June 2018.

[7] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, "Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 34–49, 2020.

[8] N. Mayer, I. Eddy, P. Hausser et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[9] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Lecture Notes in Computer Science, Computer Vision – ECCV 2016 Workshops*, Springer, Berlin, Germany, 2016.

[10] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proceedings of the Thirty-First AAAI Conference on Artificial*

Intelligence (AAAI-17), San Francisco, CA, USA, February 2017.

[11] J. Janai, F. G¨uney, A. Ranjan, J. B. Michael, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Computer Vision—ECCV 2018, Lecture Notes in Computer Science*, Springer, Berlin, Germany, 2018.

[12] P. Liu, I. King, M. R. Lyu, and X. Jia, "Ddflow: learning optical flow with unlabeled data distillation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8770–8777, 2019.

[13] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: a survey," 2019, https://arxiv.org/abs/1902.06162.

[14] D. Pathak, P. Krahenbuhl, J. Donahue, Trevor Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[15] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[16] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Computer Vision—ECCV 2016, Lecture Notes in Computer Science*, Springer, Berlin, Germany, 2016.

[17] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[18] X. Zhai, A. Oliver, A. Kolesnikov et al., "S4L: self-supervised semi-supervised learning," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, November 2019.

[19] P. Liu, M. Lyu, I. King et al., "SelFlow: self-supervised learning of optical flow," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.

[20] J. Canny, "A computational approach to Edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[21] L. Zhao, Q. Mo, S. Lin et al., "UCTGAN: diverse image inpainting based on unsupervised cross-space translation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5740–5749, Seattle, WA, USA, June 2020.

[22] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: a survey of deep learning-based methods," *Computer Vision and Image Understanding*, vol. 192, Article ID 102897, 2020.

[23] Z. Sun, "A survey of multiple pedestrian tracking based on tracking-by-detection framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, 2021.

[24] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley, Hoboken, NJ, USA, 2001.

[25] W. Choi and S. Savarese, "Understanding collective activitiesof people from videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1242–1257, 2014.

[26] Z. Hu, "Deep convolutional neural network-based Bernoulli heatmap for head pose estimation," *Neurocomputing*, vol. 436, pp. 198–209, 2021.

[27] Z. Hu, "A CRNN module for hand pose estimation," *Neurocomputing*, vol. 333p. 156, 168.

[28] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2543–2549, Stockholm, Sweden, May 2016.

[29] A. Marginean, R. Brehar, and M. Negru, "Understanding pedestrian behaviour with pose estimation and recurrent networks," in *Proceedings of the 2019 6th International Symposium on Electrical and Electronics Engineering (ISEEE)*, pp. 1–6, Galati, Romania, October 2019.

[30] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? a study on pedestrian path prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[31] M. Goldhammer, S. K¨ohler, K. Doll, and B. Sick, "Camera based pedestrian path prediction by means of polynomial least-squares approximation and multilayer perceptron neural networks," in *Proceedings of the SAI Intelligent Systems Conference (IntelliSys), 2015*, pp. 390–399, London, UK, November 2015.

[32] S. Zernetsch, S. Kohnen, M. Goldhammer, K. Doll, and B. Sick, "Trajectory prediction of cyclists using a physical model and an artificial neural network," in *Proceedings of the 2016 Intelligent Vehicles Symposium (IV)*, pp. 833–838, Gothenburg, Sweden, June 2016.

[33] K. G. Alahi, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, Las Vegas, NV, USA, June 2016.

[34] R. Urtasun, D. J. Fleet, and P. Fua, "3D people tracking with Gaussian process dynamical models," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pp. 238–245, New York, NY, USA, June 2006.

[35] R. Quintero Minguez, I. Parra Alonso, D. Fernandez-Llorca, and M. A. Sotelo, "Pedestrian path, pose, and intention prediction through Gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2019.

[36] V. Kress, S. Zernetsch, K. Doll, and B. Sick, "Pose based trajectory forecast of vulnerable road users," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1200–1207, Xiamen, China, December 2019.

[37] CMU graphics lab motion capture database, http://mocap.cs.cmu.edu/.

[38] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR*, Salt Lake City, UT, USA, June 2018.

[39] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[40] R. Quintero, J. Almeida, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction using body language traits," in *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium*, pp. 317–323, Dearborn, MI, USA, June 2014.

[41] S. Worrall, "Multi-sensor detection of pedestrian position and behaviour," in *Proceedings of the 23rd World Congress on Intelligent Transport Systems*, pp. 1–12, Tokyo, Japan, 2016.

[42] R. Quintero, I. Parra, D. Fernández-Llorca, and M. A. Sotelo, "Pedestrian intention recognition by means of a Hidden Markov Model and body language," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, Yokohama, Japan, October 2017.

[43] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[44] J.-T. Jun-Tao Xue, S.-P. Shi-Ming Wang, and S.-M. Wang, "Research of vehicle monocular measurement system based on computer vision," in *Proceedings of the 2013 International Conference on Machine Learning and Cybernetics*, pp. 957–961, Tianjin, China, July 2013.