Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02052-5

ORIGINAL RESEARCH

Traffic parameter estimation and control system based on machine vision

Zhe Dai[1] · Huansheng Song[1] · Haoxiang Liang[1] · Feifan Wu[1] · Xuan Wang[2] · Jinming Jia[1] · Yong Fang[1]

Received: 8 August 2019 / Accepted: 29 April 2020

Abstract
With the rapid development of urbanization in the world, it has brought enormous pressure on urban traffic management and control such as traffic congestion. An excellent urban traffic management and control system consists of three critical aspects: obtaining traffic parameters, developing traffic control scheme, and evaluating traffic control scheme. Intersection signal timing is one of the most important parts in urban traffic control. This paper proposed an intersection signal timing system based on traffic video which consists of three parts: acquisition of video-based traffic parameters, calculation of traffic flow-based signal timing scheme, and evaluation of intersection signal timing scheme. In the first part, we used advanced techniques such as deep learning and image processing to obtain traffic parameters such as traffic flow, vehicle type, composition of different vehicle types, and speed of vehicles passing through a scene in a traffic video. In the second part, we calculated the signal timing scheme of the video at the traffic scene through the obtained traffic flow information with Webster method. In the third part, the detailed traffic parameters and signal timing scheme were input into the VISSIM software for traffic microscopic simulation, which was used to evaluate the signal timing scheme. The experimental results show that the accuracy of the detailed traffic flow information obtained by the proposed system can reach more than 90%, the accuracy of composition of different vehicle types can be achieved more than 98%, and the vehicle speed accuracy can reach more than 95%. Therefore, the system improves the reliability and adaptability of the whole signal timing network. At the same time, the simulation results show that the proposed system integrates the acquisition of traffic parameters and the calculation and evaluation of signal timing schemes, and provides a good solution for solving research problems and actual needs such as signal timing optimization.

Keywords Intelligent transportation system · Traffic parameter · Image processing · Traffic simulation

1 Introduction

With the rapid development of urbanization in the world, it has brought enormous pressure on urban traffic management and control such as traffic congestion. The main reason is that it is difficult to find an excellent traffic control system due to the complexity of the traffic environment. Intersection signal timing scheme is the key for an urban traffic control system. A good scheme can not only improve the traffic efficiency of the intersection, but also effectively alleviate the congestion of the roads connected to the intersection.

In order to create a good intersection signal timing scheme we need to consider the following three major steps. Firstly, it is necessary to analyze the scene characteristics and actual traffic parameters of the specific intersection (Liu 2015). Secondly, we need to calculate the signal timing scheme of the intersection (Chiou 2018; Mahalle 2019). Finally, the scheme is evaluated by tests under actual conditions or the results of simulation (Hwang 2018). We can obtain an optimal solution by optimizing the signal timing scheme and repeating the three steps continuously. However, in the process, there are still some difficult problems to solve, such as the difficulty in obtaining traffic parameters in traffic scenarios,

✉ Huansheng Song
hshsong@chd.edu.cn

Zhe Dai
zhedai@foxmail.com

Xuan Wang
xuanwang_91@126.com

[1] Chang'an University, Xi'an, China

[2] Yantai University, Yantai, China

Published online: 25 May 2020

and implementing and evaluating traffic timing scheme under the real conditions.

For the issue of intersection signal timing (Guo et al. 2017; Lu et al. 2017; Roshandeh et al. 2014; Lo 2016), the most critical traffic parameters include traffic flow, vehicle types, composition of different vehicle types, and speed of vehicles. Traffic parameters are obtained traditionally by installing various sensors such as magnetic induction coils, infrared lasers, radars, and ultrasonic sensors. The method of using the magnetic induction coil to calculate the traffic flow through the road section is relatively conventional. The detection effect of this method is also good. However, the disadvantage of this method is that it is necessary to embed the induction coil under the ground, which enhances the construction work load and the difficulty in maintenance. Additionally, once the road surface is changed, the coil needs to be relocated. In contrast, using infrared lasers to calculate traffic flow does not cause damage to the road surface, but is easily interfered by factors such as pedestrians and non-motor vehicles, thereby affecting accuracy. Using radar to measure the speed of a vehicle is to set up a radar transmitter next to a road. A radar beam is transmitted to the direction of the road, which produces a reflected echo of the vehicle. The speed of the vehicle is then calculated through echo analysis. This is also the current mainstream speed measurement method, but this method is difficult to use in complex scenes. For example, in the road intersection scene, the motion path of a vehicle is completely random, and it is not possible to obtain the speed of different vehicles through the radar. For the ultrasonic sensing method, the sound wave velocity is transmitted by an ultrasonic transmitting device, and the distance and speed of a vehicle can be calculated according to the time that the receiver takes to receive the ultrasonic wave. This is similar to the principle of radar speed measurement, but the ultrasonic sensor has a service life of only a few weeks in the extremely dusty environment of the intersection, so it is not practical. In addition to the traditional methods, it is currently popular to obtain traffic parameters through traffic video (McLauchlan et al. 1997; Haiying 2018). The advantage of this method is that it is not limited by the road surface conditions, the installation does not damage the road surface, and the traffic parameter data is detected by installing a camera above the road. Inspired by the rapid development of various image-processing technologies, video detection can be used to analyze and capture the violation behaviors of vehicles as well as to obtain important traffic data required for signal timing tasks such as traffic flow and vehicle speed. Acquisition of the vehicle speed needs to be calibrated. The traffic flow information, the vehicle types, and the composition of different vehicle types are obtained by acquiring the trajectory of the vehicle through the object detection and the tracking method and followed by analysis the trajectory using the trajectory-processing method.

The calibration methods (Lee 2019) for vehicle speed include the following processes. Firstly, the video scene is calibrated to obtain the conversion relationship between the image pixel information of the video and the real-world information, thereby obtaining the actual distance corresponding to the image distance. Then combine the actual distance with the time to get the speed of the vehicle object. At present, the existing calibration methods are mainly divided into three categories, the traditional calibration method (Wang and Wu 2010), the self-calibration method (Kampel 2011), and the active vision-based calibration method (Davani 2016). The traditional calibration method is highly accurate, but it needs to be assisted by markers such as a checkerboard. The most classical algorithm in the traditional calibration method is PnP-based calibration theory (Xie 2012). For example, the flexible new technique for camera calibration proposed by Zhang (2000), and Sinha (2017) from Microsoft Research Institute proposed a camera automatic calibration algorithm. Since the self-calibration method needs to automatically acquire a large amount of parameter information of the scene, the accuracy of the method will be drastically reduced under complicated conditions, which is not conducive to practical application; the calibration method based on active vision is simple to calculate, but the inadequacy is that it is necessary to control the movement of the camera. This method requires high experimental equipment and platforms, and the accuracy is not as high as traditional methods.

Visual object detection (Wang 2016; Zhang 2018) is a research hotspot in the field of computer vision and machine learning. Object detection algorithms can be broadly divided into methods based on traditional manual features and methods based on deep learning (Berg 2016; Sun 2017). Target detection methods based on manual features include Pail Viola and a boosted cascade of simple features to rapidly detect object used by Jones (2001), HOG features proposed by Dalal and Triggs (2005), and deformable part based model (DPM) (Felzenszwalb and Ramanan 2005) proposed by Felzenszwalb et al. which was the best feature detect algorithm. The deep learning-based object detection algorithm has started to develop rapidly since Girshick et al. proposed (regions with CNN features) R-CNN (Girshick et al. 2014). The latest and best comprehensive deep learning algorithm is YOLOv3 (Farhadi 2018) proposed by Joseph Redmon. Compared with other algorithms, it has greatly improved the detection accuracy and processing speed.

The current visual object tracking method (Kim 2019) is divided into two categories. The first one is the discrimination method (Sikora 2017), also called the tracking-by-detection method, which performs object detection for each frame of the video sequence followed by pairing and

connecting the detection results of the objects between adjacent frames to obtain the object image trajectory. The second one is the generation method (Matas 2013), First, a rectangle is manually given in the first frame of the video sequence. In actual cases, the manual label rectangle is usually the detection result of the tracking object. Then model the area in the rectangular box. Finally, in the subsequent frames, the area in which is most similar to the model is continuously searched for the predicted position of the tracking object. The detection accuracy of the object is the key of the first tracking strategy. When the detection effect is poor or unstable, the object misdetection and missed detection will cause the tracking failure. The second tracking strategy does not need to be detected in every frame, but the construction of the tracking target feature and the feature-matching algorithm in the new image are the core of the strategy. The well-known generation methods include Kalman-filter-based method (Kaur and Sahambi 2016), correlation-filter-based method (Yang 2017), deep-learning-based method (Han 2015) Kernelized-correlation-filters (KCF)-based method (Batista 2015).

The time ratio of the red and green lights at the intersection is called the signal timing which is an important part of traffic control system. The advanced urban traffic control system is one of the important ways to improve the efficiency of the traffic operation, and it is also an important symbol of urban modernization. At present, there are many popular signal timing calculation methods in different countries, such as the US HCM method, the British TRRL method (Webster), and the Australian ARRB method. Axelek innovatively proposed the concept of parking compensation coefficient in the ARRB method and combined it with the vehicle delay time to evaluate and calculate the signal timing scheme. But the most classic and most recognized method is the Webster method, which is based on the shortest delay time of the vehicle passing through the intersection, and is modified according to various constraints under realistic conditions to determine the best signal timing scheme.

On the basis of researches, countries around the world have also compiled a variety of intersection signal timing manuals, which elaborated on the basic concepts, operational procedures, maintenance updates, and other issues involved in the signal timing design, for examples, the Traffic Signal Timing Manual (TSTM) prepared by the Federal Highway Administration (FHWA), the guidelines for traffic sign (RiLSA) prepared by the German Institute of Road and Traffic Engineering Research. In addition to various signal timing manuals, a series of traffic signal control systems have also been used widely, including the British SCOOT system and the Australian SCATS system, and traffic simulation software such as VISSIM (Noyce 2013), SYNCHRO, and CROSIM.

Although there has been a lot of research in the extraction of the traffic parameters and the intersection signal timing, few people consider it as a complete work due to the large difference in the professional knowledge between the two fields. In this paper, we propose an intersection signal timing system based on traffic video, which is mainly to solve three problems: (i) the diverse traffic parameters is difficult to obtain; (ii) the traffic simulation conditions are set unrealistic; and (iii) the acquisition of traffic parameters and traffic control are difficult to combine.

2 Overview of the system

This paper presents an intersection signal timing system based on traffic video. The system mainly consists of five steps. First, we input a traffic video. Second, we process the traffic video and obtain the traffic parameters using the techniques of deep learning and image processing. The obtained traffic parameters include the detailed traffic flow information and the composition of different vehicle type. The third step is to calculate the signal timing scheme of the intersection according to the obtained traffic flow information by using the classic Webster signal timing method. In the fourth step, according to the actual situation and the obtained traffic parameters, the virtual environment of the traffic scene is constructed and microscopically simulated by using VISSIM simulation software. The final step is to obtain the evaluation result of traffic simulation of VISSIM.

In the second step of video processing, the classic six-point calibration algorithm is used to calibrate the video scene for obtaining the conversion relationship between the image pixel information of the video and the real-world information. The vehicle object is detected using the YOLOV3 algorithm in deep learning. At the same time, a multi-object tracking method based on the object detection result and KCF algorithm is proposed to track the objects and obtain the trajectories. Then, the image trajectories of all vehicle objects are processed and analyzed to obtain traffic parameters. In the third step, we introduced how to use the Webster method and traffic flow information to calculate the multi-phase signal timing scheme. The fourth step introduces how to use the detailed traffic parameters to build a more realistic traffic simulation environment on VISSIM software. Finally, the traffic control scheme is simulated and the performance of traffic control scheme is evaluated. Figure 1 gives the framework of the proposed intersection signal timing system based on traffic video. In the traffic parameter extraction part, we showed three scenarios of intersection, T-junction, and road, which means that the proposed traffic parameter extraction algorithm is not limited by the scene. But in the traffic control part, we only discussed the intersection scenario.

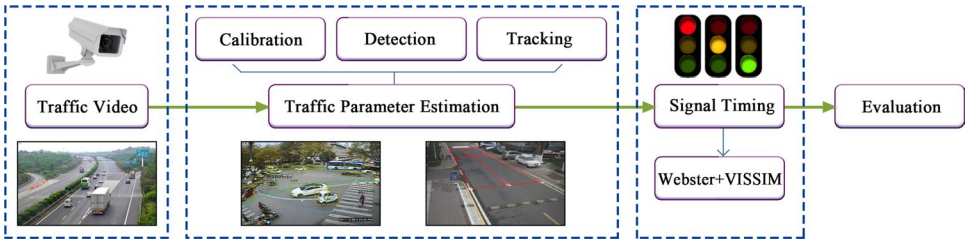

Fig. 1 Overview of the proposed system

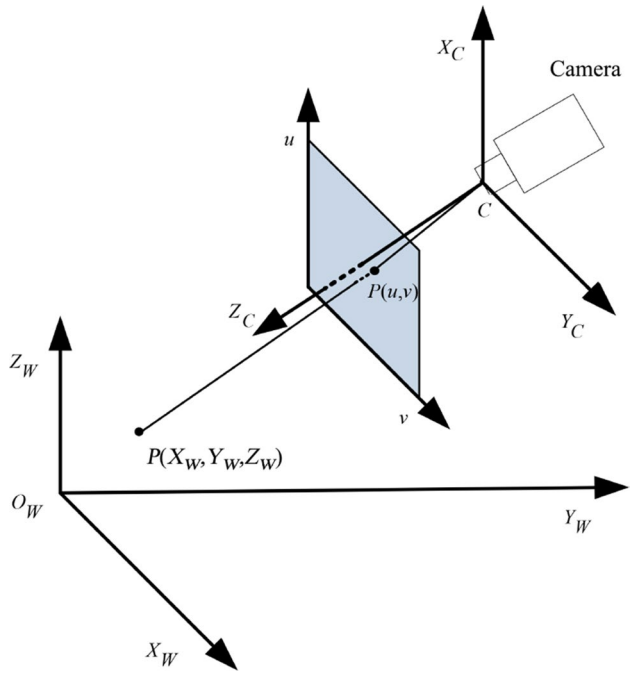

Fig. 2 The linear imaging model of the camera

3 Traffic parameters detection

3.1 Vehicle speed

The acquisition of vehicle speed needs to be achieved by the calibration method of the camera. The imaging model of the camera determines the geometric relationship between the image captured by the camera and the real-world space. Calibration is the process of determining the imaging model of the camera. For different camera models, the traditional calibration technique can be divided into two types, linear calibration and nonlinear calibration. Linear scaling is a solution to the camera's parameters for a linear imaging model of the camera. Figure 2 shows the linear imaging model of the camera, ignoring the distortion.

Three coordinate systems exist in the model, which are the world coordinate system (X_W, Y_W, Z_W), the camera coordinate system is represented by (X_C, Y_C, Z_C), and the image pixel coordinate system (u, v). According to the linear perspective projection transformation of the camera, the perspective projection matrix transformation equation of the camera can be obtained, as shown in formula (1):

$$\lambda_i p = K[R_{3\times3}|t_{3\times1}]P = C_{3\times4}P. \tag{1}$$

In the Formula (1), $p = [u_i, v_i, 1]^T$ represents the homogeneous coordinate value of a certain pixel point in the image coordinates. $P = [X_i, Y_i, Z_i, 1]^T$ represents the homogeneous coordinate value of the pixel point in the real-world space. λ_i represents the scale factor. K represents the inside parameter matrix of the camera, which is a 3×3 matrix. $R_{3\times3}$ represents the rotation matrix between the camera coordinate system and the world coordinate system. $t_{3\times1}$ represents the translation vector between the camera coordinate system and the world coordinate system. $C_{3\times4}$ represents the perspective projection matrix of the camera.

A number of sophisticated methods have been developed to calibrate cameras. A more typical method is the flexible new technique for camera calibration proposed by Zhang et al. Firstly, the method needs to establish a pinhole model. The real-world information of at least 6 points is then manually determined. Using the correspondence between the image information of the 6 points and the real-world information, the parameters of the camera can be determined, thereby solving the 12 coefficients of the matrix $C_{3\times4}$. The calibration of the camera is now complete. In this paper, the calibration method proposed by Zhang et al. is used to calibrate the traffic scene to obtain the internal and external parameters of the camera. The real-world distance on the ground plane corresponding to the image distance is obtained by converting the image points into real-world points by Formula (1). The final step is to calculate the speed of the traffic target.

Suppose there are two points $p_1(u_1, v_1)$, $p_2(u_2, v_2)$ on the image. Two points $P_1(x_1, y_1, 0)$, $P_2(x_2, y_2, 0)$ on the corresponding ground plane in real world space can be obtained by Formula (1), the unit is meter. If the time difference T is known, the average speed v of the object moving from $P1$ to $P2$ can be calculated by Formula (2).

$$v = \frac{|P_1, P_2|}{T} = \frac{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}{T}. \quad (2)$$

3.2 Detection and tracking

The ultimate goal of object tracking is to obtain an image trajectory of a vehicle object within the video. Object tracking methods can be divided into two types: data association-based and feature-based. The data association-based method uses a reliable detection algorithm to obtain the detection result of the current video frame, and then correlates and matches the detection results of two adjacent frames of the video to achieve object tracking. The feature-based method only needs to obtain the object detection result in the first frame of the video and extract the object feature, then matching the feature in subsequent frames to achieve object tracking. The limitations of two tracking methods are relying on the detection accuracy and the performance of the feature extractor, respectively. Due to the complexity of the traffic scene, it is difficult to achieve a better object tracking effect simply relying on traditional trackers or links of detection results. Therefore, this study attempts to find a way to solve this difficulty. This section introduces the proposed multi-object tracking method based on the object detection result combined with the KCF tracker. This method is used to obtain an image trajectory of a vehicle object within the video.

We chose the YOLOV3 algorithm in deep learning and used the training model provided by AlexeyAB on the Github website to provide us the detection result of the vehicle object. The training model is publicly accessible and can be downloaded from the following URL: https://github.com/AlexeyAB/Yolo_mark. We also proposed a template matching method to pair the two sets of target position results between successive frames of video to achieve the continuation of the trajectory. The KCF tracker comes from the Tracker library of OpenCV3.4 and provides us with the predicted position in the image based on the image template of the object. The whole process of the proposed multi-object tracking method is shown in Fig. 3.

The proposed method comprises four steps.

Step 1: The video of the traffic scene is cyclically read, where the pixel information of the image of the t-th frame is recorded as I_t, and the length and width of I_t are recorded as w_I and h_I. The I_t is input to the detection training model to obtain detection result information $D_t = \{DB_i, i = 1, 2, \ldots, n\}$ in t-th frame, and one DB_i corresponds to one detected object information. DB_i contains 4 attributes, the pixel midpoint coordinate $Ct = (x, y)$ of the object detection box, the *width* and *height* of the object detection box, and the category CL (car=1, bus=2, and truck=3). Finally, I_t and D_t are bound to F_t as a summary of the t-th frame information.

Fig. 3 Multi-object tracking method used in this study

Step 2: Create a temporary trajectory list $TT = \{T_j, j = 1, \ldots, n\}$, the initial value is empty, T_j is the i-th trajectory in the temporary trajectory list, $T_j = \{B_i, i = 1, \ldots, n(last)\}$, B_i is the i-th node in the j-th trajectory. Each T_j also has a suspicion attribute Dt, which takes a positive integer, an initial value of 0, and a category attribute TCL with a value of 1, 2, 3 (car=1, bus=2, truck=3), and a speed attribute V in km/h. Create the final track list $AT = \{T_t, t = 1, \ldots, n\}$, the initial value is empty, and the T_t and T_j have the same properties.

Step 3: Start with t equal to 1 and read F_t cyclically. After acquiring an F_t, the trajectories in the TT are first updated, and T_i in which the Dt is larger than the video frame rate or the B_{last} is located at the edge of the video frame is taken out from the TT. The taken trajectory considers that the vehicle object has left the video area and add the trajectory into the final trajectory list AT. Secondly, it is judged whether Ts is empty. If it is empty, each DB_i in the current D_t is used as the first node of the track to create a new trajectory, and returns to step 3 to

read the next frame. If not, the process proceeds to step 4.

Step 4: The B_{last} of each T_j in the TT is matched with all DB_i in the current D_t to calculate the matching value m, which is denoted as α_{ji}. Then, each DB_i in D_t is matched with all B_{last} of each T_j in the TT to calculate the matching degree, which is denoted as β_{ji}. When α_{ji} and β_{ji} of the same landmark are simultaneously the maximum in the respective sets, it is considered that the trajectory T_j matches the detection result DB_i successfully. There are three cases at this point. Case1, if the matching is successful, the corresponding object detection information DB_i is added to the trajectory list as the next node of the trajectory T_j, and the suspicion Dt of the track is set to 0; Case2, if the detection frame DB_i fails to match, the DB_i is used as the starting point of the new trajectory adding to the TT; Case3, if the B_{last} of the trajectory T_j match fails, the KCF algorithm is used to find the predicted position PB of B_{last} on the I_t in the current F_t. The PB has the same attributes with DB_i, whose category CL inherits the value from the corresponding B_{last}. At this point, two situations will occur. If the KCF tracker successfully obtains the predicted position PB of B_{last}, then PB is added to the track list as the next node of the trajectory T_j, and the value of the suspicion Dt of the trajectory T_j is increased by one. If the KCF tracker cannot obtain the predicted position information of B_{last}, a B_{last} is copied as the next node of the trajectory T_j to join the trajectory list, and the value of the suspicion Dt of the trajectory T_j is incremented by one. Finally, return to step 3.

The calculation of the matching value m is obtained by calculating the midpoint distance and the overlapping ratio of the two rectangular frames. The input is a B_{last} and a DB_i. As shown in Fig. 4a, when the two rectangular boxes have overlapping regions, the overlap value o is equal to the ratio of the area A and the sum of the areas A, B, and C. As shown in Fig. 4b, when the two rectangular boxes have no overlapping area, the overlap value is equal to zero. Next, the pixel distances d of the two rectangles are calculated using the midpoint coordinates of the two rectangles, as shown in Fig. 4c. Finally, the matching value m can be obtained by Formula (3).

$$d_I = \sqrt{w_I^2 + h_I^2}$$
$$m = \begin{cases} 0 & \text{if } m = 0 \text{ or } \frac{d}{d_i} > \max[w, h] \cdot \\ m = 0.8 \times o + 0.2 \times \frac{d}{d_i} & \text{otherwise} \end{cases} \tag{3}$$

3.3 Trajectory processing

In this section, the final trajectory list AT obtained in Sect. 3.2 is mainly processed to obtain traffic parameters that can be used for signal timing. This section is divided into three tasks. The first is to classify the vehicle objects represented by each trajectory. The second is to classify and count the vehicles according to the moving path of the vehicle. The third is to confirm the vehicle's average speed when it passes the video camera.

3.3.1 Vehicle type

The confirmation of the vehicle category is determined based on the category detection result of each node in one trajectory. This method operates on each T_j in the AT. First, take the category attribute CL of all nodes B_i in T_j, calculate the sum of all CL values, and divide by the number of nodes to obtain the average CL of the entire track node. The category attribute TCL assigned to the trajectory after rounding the average, as shown in Formula (4).

$$TCL = \frac{\sum_{i=1}^{last} CL}{last}. \tag{4}$$

3.3.2 Traffic flow

For the counting task of the vehicle, it is necessary to manually divide the intersection area of the video scene. As shown in Fig. 5, the scenes are partitioned as needed (A, B, C, D etc.) for different intersection types. Take the intersection as an example, and operate on each T_j in the AT. First, take out the midpoint coordinates Ct_1 and Ct_{last} of B_1 and B_{last} in T_j, and then judge which area of Fig. 5a they are located.

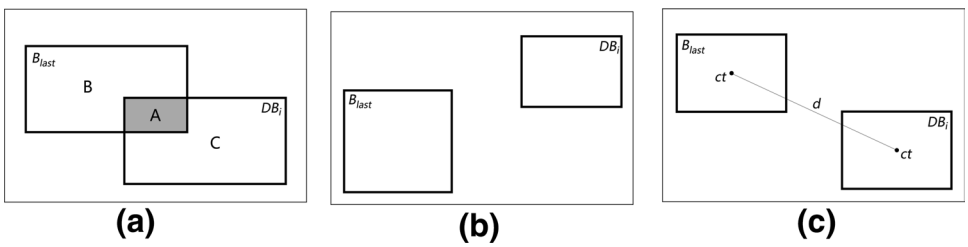

Fig. 4 Relationship between two boxes. **a** With overlapping area. **b** No overlap area. **c** Pixel distance of two boxes

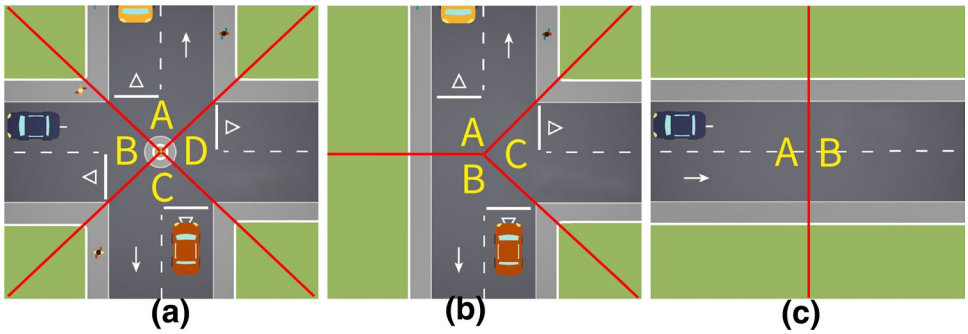

Fig. 5 Partition of scenes at different intersection types. **a** Intersection. **b** T-junction. **c** Road

Assuming that Ct_1 is in the area A and Ct_{last} is in the area C, and the *TCL* of the trajectory equal to 1, then the counter of the car from A to C is incremented by one. If Ct_1 and Ct_{last} are in the same area, the T_j will not be counted. Obviously, when the video scene is an intersection, according to the different motion paths and categories of the vehicle, there are 36 counters in total, and other scenes are similar. After all the T_j in the AT are processed in this way, detailed traffic flow information in the traffic scene video can be obtained.

3.3.3 Vehicle speed

The vehicle speed is calculated to obtain the average speed of the vehicle as it passes through the traffic video scene, and still operates on each T_j in the AT. We take out a T_j and calculate the actual world distance d_i between the midpoint B_i and B_{i+1} in turn since i from 1 to n. The calculation of d_i depends on the calibration method in Sect. 3.1. Converting $Ct_i = (u_i, v_i)$ and $Ct_{i+1} = (u_{i+1}, v_{i+1})$ to real world coordinates $(x_i, y_i, 0)$ and $(x_{i+1}, y_{i+1}, 0)$ respectively by Formula (1), and then the d_i can be calculated by Formula (5). By summing all the d_i, the real-world distance D_j of the trajectory T_j can be obtained in meters, as in Formula (6).

$$d_i = \sqrt{(y_2 - y_1)^2 - (x_2 - x_1)^2}, \tag{5}$$

$$D_j = \sum_{i=1}^{n} d_i. \tag{6}$$

Assuming that the frame rate of the video is f, the unit is frame per second. The actual average speed of the vehicle passing through the traffic video scene corresponding to the trajectory T_j can be calculated by the Formula (7), the unit is kilometers per hour, and the *last* is the number of B_i in the T_j. Finally, bind V_j to the speed attribute V of T_j.

$$V_j = 3.6 \times \frac{D \cdot f}{last}. \tag{7}$$

4 Signal timing

4.1 Webster-based Signal timing

To the best of our knowledge, the most classic and widely used method of signal timing calculation in the world is Webster method. The Webster method obtains a timing scheme by calculating the signal period duration and the green time. The specific calculation process is as follows.

Firstly, Formula (8) is used to calculate the signal period duration C in second, the C is recommended to range between 40 and 180. L in Formula (8) is the total loss time of the signal and is calculated by Formula (9). In Formula (9), L_s stands for the loss time when the vehicle start, which should be measured. If there is no measured data, it can take 3s. A is yellow light time, which can take 3s. I is green light interval time, which can take 3s. k is the number of signal phase of the intersection. Y in Formula (8) is the sum of traffic flow ratio, which represents the degree of load at the intersection, calculated according to Formula (10). Y is the sum of the maximum flow ratios of y_j or $y_j^{'}$ values of all signal phases of the composition period. j is the number of phases in one cycle. y_j and $y_j^{'}$ are the flow ratios of the j-th phase. q_d is the actual traffic volume in passenger car unit (PCU) per hour. S_d is the design saturation flow in PCU per hour. The value of Y is usually not more than 0.9. Then, the total effective green time G_e of each cycle is calculated by Formula (11). Finally, the effective green time g_{ej} of each phase can be obtained from Formula (12). The signal timing scheme based on traffic flow information is obtained through the value of g_{ej}.

$$C = \frac{L}{1 - Y}, \tag{8}$$

$$L = \sum_{k} (L_s + I - A)_k, \tag{9}$$

$$Y = \sum_{j=1}^{j} \max[y_j, y_j^{'}, \dots] = \sum_{j=1}^{j} \max\left[\left(\frac{q_d}{S_d}\right), \left(\frac{q_d}{S_d}\right)_j^{'}, \dots\right], \tag{10}$$

$$G_e = C - L, \tag{11}$$

$$g_{ej} = G_e \cdot \frac{\max[y_j, y_j', \dots]}{den}. \tag{12}$$

4.2 VISSIM-based simulation

VISSIM is a microscopic multi-modal traffic flow simulation software package developed by PTV Planung Transport Verkehr AG in Karlsruhe, Germany. This study mainly uses the VISSIM system to evaluate the calculated intersection timing signal timing scheme. Firstly, a realistically simulated intersection network should be established. Secondly, the traffic flow characteristics and driving rules are set. The traffic flow characteristics include the number, category and average speed of the passing vehicles. Next, set the traffic signal control model and the signal timing scheme is put into the model. The evaluation parameter detector is set then, and the main evaluation parameters for the signal timing of the intersection are the delay and the traffic volume. Finally, run the traffic simulation and output the simulation results.

Due to the complexity of the traffic scene, when using the traffic micro-simulation software, the system parameters need to be corrected according to the specific situation. However, in practical applications, most users often ignore the work of parameter calibration, but use the default recommended value of the system or the recommended value in the national traffic specification manual, or even the empirical value to directly perform modeling analysis. This kind of simulation results lack of rationality and persuasiveness. Through detailed traffic parameters obtained by the system, this study can provide VISSIM with more realistic traffic parameter setting basis and obtain more accurate simulation evaluation results, so as to optimize traffic control more accurately and realistically.

5 Experiment

In this section, we conducted experiments on the accuracy of traffic parameters and the evaluation of signal timing schemes. The traffic parameters mainly include speed information of the vehicle object and detailed traffic flow information. The experimental part is arranged as follows, Sect. 5.1 is accuracy analysis of the speed extraction, Sect. 5.2 is the accuracy analysis of the traffic flow information and the vehicle composition, and Sect. 5.3 is the evaluation result of the traffic simulation.

5.1 Speed analysis

The actual speed of the vehicle in the traffic video needs calibrating the traffic video scene. The calibration, as described in Sect. 3.1, requires obtaining the internal and external parameters of the camera to obtain the conversion relationship between the image information and the real-world information. Camera internal parameters such as focal length, main point, aspect ratio, etc. can be obtained from the camera's factory configuration. The external parameters of the camera could be obtained by the flexible new technique for camera calibration proposed by Zhang et al., which needs to provide real distance information between at least 6 points in the scene and pixel coordinates in the image corresponding to the 6 points. In order to obtain better calibration results, nine landmarks evenly distributed on the road were selected in the traffic video scene. Figure 6a shows the image of the traffic video scene. The image size is 2048*1536 pixels. We have numbered 9 points selected in the road scene according to 1 to 9, which is shown in Fig. 6b. The image coordinates of these 9 points are (697, 1147), (1358, 975), (1871, 849), (347, 500), (746, 436), (1077, 391), (222, 265), (499, 227), and (742, 201) in the order of numbering, and the real-world distance relationship of these nine points is shown are Fig. 6c. The camera's

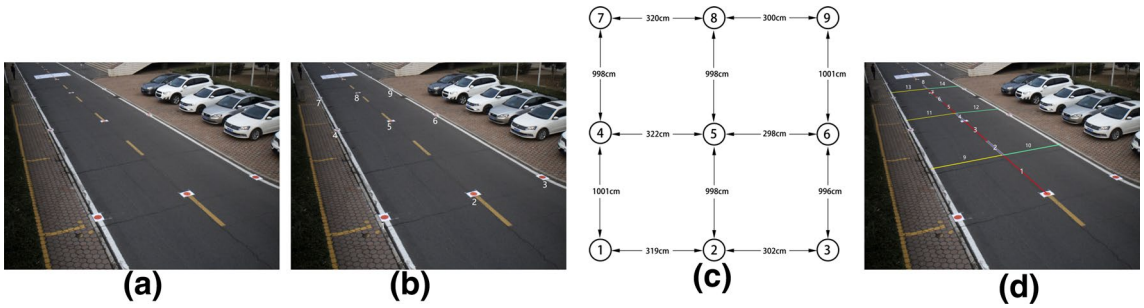

Fig. 6 Calibration process. **a** An image of the traffic video scene. **b** 9 points selected in the road scene. **c** The real-world distance relationship of the nine points. **d** Selected 14 lines in the scene

external parameters could be obtained by calling the solve-PnP function of the calib3d class in the Opencv3.4 library with the 9 points. This is an implementation code for the calibration method provided proposed by Zhang et al. Next, the conversion relationship between the image information and the real-world information is obtained by substituting the camera's internal and external parameters into Formula (1). As shown in Fig. 6d, after obtaining the conversion relationship between the image information and the real-world information, we selected 14 lines in the scene. The actual manual measurement of the lines and the distance calculated using the pixel information are compared and analyzed. The analysis results are shown in Table 1. The calculations of *Error* and *Accuracy* are shown in Formula (13), and (14), respectively.

$$Error = Manual\ result - System\ result, \tag{13}$$

$$Accuracy = \left| \frac{Error}{Manual\ result} \right|. \tag{14}$$

Table 1 shows that among the result of 14 lines, the 13-th line exhibited the highest distance accuracy (99.7%), and the 6-th line displayed the lowest accuracy (96.0%). The average accuracy is 98.3%. The result shows that the accuracy of the actual distance obtained by the image information and the introduced calibration method can reach at least 95%. The speed of the vehicle is obtained by the ratio of the actual real-world distance to the time consumed. This also means that when the 2-dimension image trajectory of the vehicle in the video is acquired correctly and the time is known, the accuracy of the average speed can be achieved to 95% or above.

5.2 Traffic flow and vehicle composition

5.2.1 Detailed traffic flow

This section analyzes the detailed traffic flow obtained in the system and the composition of different vehicle types. The experiment selected three different traffic scene for analysis, a crossroad, a T-junction and a road scene, each video being one-hour long. At the same time, we did manual marking when collecting video, and classified statistics on different moving paths and different types of vehicles. The experimental results were displayed in Table 2, Table 3, and Table 4. The screenshots from videos corresponding to the results are shown in Fig. 7a–c, respectively. The *Error* is calculated as Formula (15).

$$Error = \frac{System\ result - Manual\ result}{Manual\ result}. \tag{15}$$

As shown in Table 2 and Fig. 7a, the overall error detected at the cross was -1.0%. Generally, we found the larger the sample volume, the lower the absolute number of the error. For an example, at the road route D to B, the vehicle numbers was counted 284. The errors detected using the framework was 3.5%. Thus the errors detected at each mouth were quite similar, indicating that the results obtained using our framework are highly reliable.

The overall error determined at the T-Junction was 7.2% (Table 3 and Fig. 7b). The correlation between sample volumes and errors rate was similar compared to that detected from the cross, suggesting that the detection result using our framework will be remarkably accurate when the sample volume is large enough.

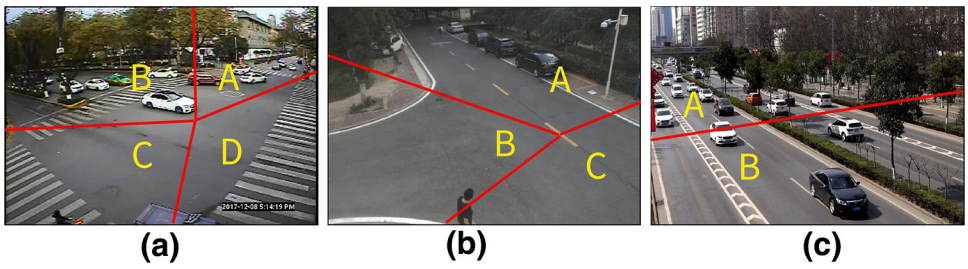

Fig. 7 Experiment traffic scenes. **a** Intersection. **b** T-junction. **c** Road

Table 1 Calibration analysis result

Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Manual result (cm)	392	212	395	213	383	201	383	198	320	303	321	300	322	299
System result (cm)	385	210	393	205	381	209	387	211	322	282	327	292	323	302
Error (%)	7	2	2	8	2	−8	−4	−3	−2	5	−6	8	−1	7
Accuracy(%)	98.2	99.1	99.5	96.3	99.5	96.0	99.0	98.5	99.3	98.3	98.1	97.3	99.7	97.7

Table 2 The results obtained from the system and related manual detection from Fig. 7a, and error of the results

	Route	A → B	A → C	A → D	B → A	B → C	B → D	C → A	C → B	C → D	D → A	D → B	D → C	Total
Manual	Car	59	189	64	62	143	148	228	176	63	80	230	93	1535
	Truck	3	8	2	0	12	11	6	8	0	3	23	7	83
	Bus	6	20	0	5	10	25	18	22	0	1	31	0	138
	Total Manual	68	217	66	67	165	184	252	206	63	84	284	100	1756
	Car	55	220	59	75	123	178	210	142	54	73	238	107	1534
	Truck	2	12	1	0	9	5	10	7	0	3	17	10	76
	Bus	7	14	1	4	15	19	9	18	1	1	39	0	128
	Total System	64	246	61	79	147	202	229	167	55	77	294	117	1738
Error (%)	Car	−6.7	16.4	−7.8	20.9	−13.9	20.3	−7.9	−19.3	−14.3	−−8.8	3.5	15.1	−0.1
	Truck	−33.3	50.0	−50.0	0	−25.0	−54.5	66.7	−12.5	0	0	−26.1	42.9	−8.4
	Bus	16.7	−30.0	/	−20.0	50.0	−24.0	−50.0	−18.2	/	0	25.8	0	−7.2
	Total Error	−5.9	13.4	−7.6	17.9	−10.9	9.8	−9.1	−18.9	−12.7	−8.3	3.5	17.0	−1.0

Table 3 The results obtained from the system and related manual detection from Fig. 8b, and error of the results

	Route	A → B	A → C	B → A	B → C	C → A	C → B	Total
Manual	Car	74	128	31	120	173	23	549
	Truck	3	0	4	3	1	0	11
	Bus	5	12	2	6	10	0	35
	Total Manual	82	140	37	129	184	23	595
System	Car	80	136	38	107	191	26	578
	Truck	2	0	3	3	3	0	11
	Bus	4	9	2	8	6	1	30
	Total System	86	145	43	118	200	27	619
Error (%)	Car	8.1	6.3	22.6	−10.8	10.4	13.0	5.3
	Truck	−33.3	0	−25.0	0	50.0	0	0
	Bus	−20.0	−25.0	0	33.3	−40.0	–	−14.3
	Total Error	7.5	3.6	16.2	−8.5	8.7	17.4	7.1

Table 4 The results obtained from the system and related manual detection from Fig. 8c, and error of the results

	Route	A → B	B → A	Total
Manual	Car	1534	897	2431
	Truck	71	114	185
	Bus	168	48	216
	Total manual	1773	1059	2832
System	Car	1592	928	2520
	Truck	52	107	159
	Bus	183	61	244
	Total system	1827	1096	2923
Error (%)	Car	3.8	3.4	3.7
	Truck	−26.7	−6.1	−14.0
	Bus	8.9	27.1	12.9
	Total error	14.7	3.5	3.2

Table 4 and Fig. 7c demonstrated that the total number of vehicles had reached 2832, and the total error was only 3.2%. This result was not only because the environmental complexity of the road is low, which also shows that the proposed system has higher accuracy when the number of vehicles is larger.

From the perspective of total errors of the three road scenarios, the overall error at the crossroad was the lowest, whereas the T-junction exhibited the highest error. The high error of the T-junction was likely due to the video angle, which was constant for the vehicle counting in a fixed direction. In contrast, at the cross, the errors at the different directions could be either positive or negative, thus the total error was low after addition of the errors detected from these directions. Comparing between the results of the road and the T-junction, the errors detected from the road was

generally lower because the complexity of road is lower, which might result from fewer occlusion problems of the camera and occlusion would enhance detection errors and tracking errors. In the T-junction experiment, the error of the relevant direction of the mouth C was high, as it was the closest to the camera. The vehicle moved to a large proportion in the image, and the occlusion of the image was severe. Therefore, camera angle also plays a significant role in the accuracy of object detection. Although the errors caused by occlusion still exist, the counting method is not significantly affected by them because the counting condition of the counting method is that the starting point and the end point fall in two different areas. If a trajectory breaks or is missing during the target movement, the error can be eliminated by the counting frame, thereby achieving a higher counting accuracy.

5.2.2 Vehicle composition

Vehicle composition is also an important traffic parameter in traffic control systems. The composition of the vehicle represents the proportion of different types of vehicles to the total number of the vehicles. Based on the experiment results in Sect. 5.2.1, this section analyzes the accuracy of the vehicle composition. The results of the vehicle composition at the intersection are shown in Table 5. The vehicle composition results at the T-junction are shown in Table 6. The Table 7 is the composition result of road vehicles. In Tables 5, 6, and 7, label number represents the number of manually counted vehicles, label rate represents the result of manual statistical vehicle composition, system number represents the number of vehicles obtained by the system, and system rate represents the vehicle composition result obtained by the system. The error is calculated as Formula (16).

$$Error = |Manual\ rate - System\ rate|. \tag{16}$$

Based on the statistical results of truck in Table 6, vehicle composition error obtained by the system can be as small as 0, and the error of truck in Table 7 is the highest, which is 1.1%. The average error of Table 5 is 0.6%, the average error of Table 6 and Table 7 are both 0.7%. The experimental results indicate that the system has very good stability and high accuracy in the acquisition of vehicle composition information, which can reach more than 98%.

Table 5 Vehicle composition analysis result of intersection

Intersection	Car	Truck	Bus	Total
Manual number	1535	83	138	1756
Manual rate (%)	87.4	4.7	7.9	–
System number	1534	76	128	1738
System rate (%)	88.3	4.4	7.3	–
Error (%)	0.9	0.3	0.6	–

Table 6 Vehicle composition analysis result of T-junction

T-junction	Car	Truck	Bus	Total
Manual number	549	11	35	595
Manual rate (%)	92.3	1.8	5.9	–
System number	578	11	30	619
System rate (%)	93.4	1.8	4.8	–
Error (%)	1.1	0	1.1	–

Table 7 Vehicle composition analysis result of road

Road	Car	Truck	Bus	Total
Manual number	2431	185	216	2832
Manual rate (%)	85.8	6.5	7.7	–
System number	2520	159	244	2923
System rate (%)	86.2	5.4	8.4	–
Error (%)	0.4	1.1	0.7	–

5.3 VISSIM-based simulation

The first purpose of this section is to explain how to develop and evaluate the signal control scheme through the traffic parameters estimated by our system. The second is to prove that the system is rich in traffic parameters and has important significance and superiority in traffic control. We used VISSIM software to simulate the realistic road network environment and input traffic parameters to evaluate the signal timing scheme. Taking the traffic flow information by system at the intersection scene in Sect. 5.1 as an example, the Webster method introduced in Sect. 4.1 is used to calculate the signal timing scheme. There are three lanes in each direction of the intersection, which are straight lane, left turn lane, and right turn lane. Each lane is 3.5 m wide, and no right signal control. Four-phase signal control is used to configure the intersection. Areas A and C straight route as phase1, areas A and C turn-left route as phase 2, areas B and D straight route as phase 3, and areas B and D turn-left route as phase 4. Taking the maximum traffic flow of each of the four phases, the maximum traffic volumes of the phases 1, 2, 3, and 4 are 315, 206, 368, and 100 respectively. The saturated traffic flow of a lane is given according to the actual situation of the road, which is set at 1200. Next, the calculated flow ratio Y is equal to 0.48, L is 41s, and timing period C is equal to 79s. According to the flow ratio of each phase, Fig. 8b shows the green time of the four phases is calculated as 17s, 28s, 20s, and 14s, respectively. The intersection and phases are shown in Fig. 8a. The signal timing scheme is shown in Fig. 8c.

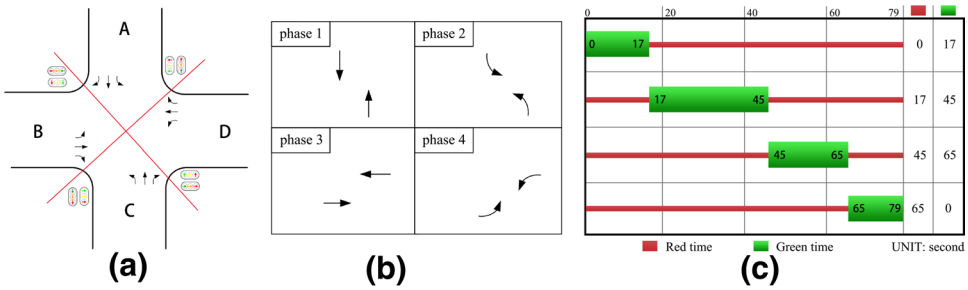

Fig. 8 Traffic scenes environment. **a** The intersection road type. **b** The intersection phase type. **c** Signal timing scheme

Table 8 Simulation result of the signal timing scheme

No.	Simulation time (s)	Average queue length (vehicle)	Maximum queue length (vehicle)	Average delay (s per vehicle)	Average stop number
1	3600	5.8	39.4	24.92	0.76
2	3600	7.1	75.55	26.98	0.81

The intersection is constructed in the VISSIM, and the traffic flow parameters obtained by the system are input into the traffic environment configuration module of VISSIM. The traffic parameters input to the traffic environment configuration module includes the number of vehicles per lane, the vehicle types, the vehicle composition, and the average speed of the vehicle passing through the intersection. The signal timing scheme is then introduced into the intersection through the signal control module provided by VISSIM. Finally, the simulation is performed using the VISSIM micro-simulation module, and the simulation results are shown in Table 8. The VISSIM vehicle input of experiment number 1 was not divided into different types, and the experiment number 2 used the detailed traffic flow input.

Table 8 shows that VISSIM can provide us with sufficient evaluation results for traffic signal timing. In the experiment number 1, it can be seen that the average delay of the vehicle is 24.92 seconds per car, and the average number of stopping per vehicle is 0.76 times. However, in the experiment number 2, the results of these traffic control evaluation indicators are significantly worse than the experiment 1. It means that if only used simple traffic parameter input, the calculated traffic control scheme is not in line with actual needs. The experimental results also showed that the proposed system can be used to calculate the signal timing scheme of traffic scenes using the traffic parameters and evaluate the scheme. It also showed that the proposed system is very comprehensive to integrate traffic parameter acquisition, signal timing calculation and evaluation, and provides a very good experimental framework and platform for traffic control optimization and research.

6 Conclusion

This paper proposed an intersection signal timing system based on traffic video. The traffic video was initially processed by using advanced techniques such as image processing and deep learning to obtain the detailed traffic parameters such as information of traffic flow and vehicle speed. The Webster method was then used to calculate the signal timing scheme of the traffic scene in the video. Finally, VISSIM was used to simulate and obtain the evaluation results of the signal timing scheme. The experimental results showed that the accuracy of the detailed traffic flow information obtained by the proposed system can reach more than 90%. Additionally, the accuracy of vehicle composition of different vehicle types can be achieved more than 98%, and the vehicle speed accuracy can reach more than 95%. The high accuracy of traffic parameters achieved by the proposed system can improve the reliability and adaptability of the signal timing. This paper provides a convenient and reliable experimental framework for research problems such as traffic control scheme optimization, and also provides a more realistic solution for the actual engineering of intersection signal control. The strength of the proposed system is that it is a single camera-based intelligent transportation system, and can estimate a variety of traffic parameters. The limitation of the proposed system is that although the traffic parameters can be obtained in real time, the signal control cannot be adjusted in real time. Therefore, the first priority of our future work is making a dynamic and automatic signal timing system based on real-time acquisition of traffic parameters, which makes the system more practical. Another application-oriented direction is to explore the area-traffic signal control system for road networks.

Acknowledgements This work is supported by the Funds for Shaanxi Key R&D Program (No.2018ZDXMGY-047) and the Funds for China Key R&D Program (No.SQ2019YFB160023).

References

Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596

Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision

Chiou SW (2018) A data-driven bi-level program for knowledge-based signal control system under uncertainty. Knowl Based Syst 160(NOV.15):210–227

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. Comput Vis Pattern Recognit 1:886–893

Davani SG (2016) A clustering approach for controlling PTZ cameras in automated video surveillance. In: IEEE international symposium on multimedia

Farhadi A (2018) Yolov3: an incremental improvement

Felzenszwalb P, Ramanan D (2005) A discriminatively trained, multiscale, deformable part model. Comput Vis Pattern Recognit, pp 1–8

Girshick R, Donahue J, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587

Guo J, Kong Y, Wei Y (2017) A model and genetic algorithm for area-wide intersection signal optimization under user equilibrium traffic. Math Comput Simul

Haiying L (2018) Vehicle motion segmentation using rigid motion constraints in traffic video. Sustainable Cities and Society, pp S2210670717310089

Han B (2015) Learning multi-domain convolutional neural networks for visual tracking

Hwang MC (2018) Value-based deep reinforcement learning for adaptive isolated intersection signal control. Intell Transp Syst IET 12(9):1005–1010

Jones MJ (2001) Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society conference on computer vision and pattern recognition

Kampel M (2011) Camera auto-calibration using pedestrians and zebra-crossings. In: IEEE international conference on computer vision workshops

Kaur H, Sahambi JS (2016) Vehicle tracking in video using fractional feedback Kalman filter. IEEE Trans Comput Imaging 2(4):550–561

Kim H (2019) Multiple vehicle tracking and classification system with a convolutional neural network. J Ambient Intell Humaniz Comput

Lee EC (2019) Comparison of facial expression recognition performance according to the use of depth information of structured-light type RGB-D camera. J Ambient Intell Humaniz Comput

Liu HC (2015) A method of expressway traffic parameter self-adaptive estimation based on hybrid particle filtering. J Highw Transp Res Dev

Lo HK (2016) Multi-stage stochastic program to optimize signal timings under coordinated adaptive control. Transp Res Part C Emerg Technol

Lu K, Du P, Huang W (2017) A novel traffic signal split approach based on explicit model predictive control. Math Comput Simul

Mahalle PN (2019) Fuzzy priority based intelligent traffic congestion control and emergency vehicle management using congestion-aware routing algorithm. J Ambient Intell Humaniz Comput, pp 1–18

McLauchlan P, Beymer D, Mali J (1997) A real-time computer vision system for measuring traffic parameters. In: CVPR, p 495

Matas J (2013) Robust scale-adaptive mean-shift for tracking

Noyce D (2013) Modeling reservation-based autonomous intersection control in vissim. Transp Res Rec J Transp Res Board 2381(2381):81–90

Roshandeh AM, Levinson HS, Zhou B (2014) New methodology for intersection signal timing optimization to simultaneously minimize vehicle and pedestrian delays. J Transp Eng 140(5):04014009

Sikora T (2017) High-speed tracking-by-detection without using image information. In: IEEE international conference on advanced video and signal based surveillance

Sinha P (2017) Autocalib: automatic traffic camera calibration at scale. In: ACM international conference

Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

Wang X (2016) Object detection from video tubelets with convolutional neural networks

Wang Z, Wu Y (2010) Multi-stage stochastic program to optimize signal timings under coordinated adaptive control. In: Control and decision conference (CCDC)

Xie M (2012) A robust o(n) solution to the perspective-n-point problem. IEEE Trans Pattern Anal Mach Intell 34(7):1444–1450

Yang MH (2017) Correlation particle filter for visual tracking. IEEE Trans Image Process PP(99):1–1

Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330–1334

Zhang Z (2018) Vehicle trajectory clustering based on 3D information via a coarse-to-fine strategy. Soft computing a fusion of foundations methodologies and applications