Quan CHEN, Ph.D. student[1]
E-mail: quanchenseu@foxmail.com
Hao WANG, Prof.[1]
(Corresponding author)
E-mail: haowang@seu.edu.cn
Changyin DONG, Ph.D.[1]
E-mail: dongcy@seu.edu.cn
[1] Jiangsu Key Laboratory of Urban ITS
Jiangsu Province Collaborative Innovation Center
of Modern Urban Traffic Technologies
School of Transportation, Southeast University
No.2 Dongnandaxue Road, Nanjing 211189, PR China

Intelligent Transport Systems
Original Scientific Paper
Submitted: 8 Nov. 2021
Accepted: 1 Mar. 2022

EMPIRICAL ANALYSIS OF VEHICLE TRACKING ALGORITHMS FOR EXTRACTING INTEGRAL TRAJECTORIES FROM CONSECUTIVE VIDEOS

ABSTRACT

This study introduces a novel methodological framework for extracting integral vehicle trajectories from several consecutive pictures automatically. The framework contains camera observation, eliminating image distortions, video stabilising, stitching images, identifying vehicles and tracking vehicles. Observation videos of four sections in South Fengtai Road, Nanjing, Jiangsu Province, China are taken as a case study to validate the framework. As key points, six typical tracking algorithms, including boosting, CSRT, KCF, median flow, MIL and MOSSE, are compared in terms of tracking reliability, operational time, random access memory (RAM) usage and data accuracy. Main impact factors taken into consideration involve vehicle colours, zebra lines, lane lines, lamps, guide boards and image stitching seams. Based on empirical analysis, it is found that MOSSE requires the least operational time and RAM usage, whereas CSRT presents the best tracking reliability. In addition, all tracking algorithms produce reliable vehicle trajectory and speed data if vehicles are tracked steadily.

KEYWORDS

video observation; integral trajectory extracting; vehicle tracking.

1. INTRODUCTION

With the development of traffic investigation techniques, empirical vehicle data support numerous up-to-date researches. Li et al. predicted safety and operation impacts of lane changes in oscillations with empirical vehicle trajectories [1]. Based on traffic data collected from Los Angeles County in 2010, the effects of traffic conditions and road characteristics on air pollutant emissions at the level of traffic analysis zone were investigated [2]. Wang et al. discussed the stability of CACC-manual heterogeneous vehicular flow with partial CACC performance degrading [3]. Based on cellphone location and license plate recognition data, Liu et al. dealt with urban transport network flow estimation [4]. Wang et al. presented a crash prediction method based on vehicle trajectory data extracted from intersection videos collected in Fengxian, China by an unmanned aerial vehicle [5]. Zhao et al. proposed a driving behaviour rule extraction algorithm based on the driver’s long-term driving experience in the processes of perception, interaction and vehicle control of road traffic information [6]. Considering travel time, travel time reliability and distance, Sun et al. proposed a multi-criteria user equilibrium model [7]. Based on bus speed, acceleration and emissions data collected from four fuel types in China, a mean distribution deviation method was proposed to identify bus pollutant emissions [8]. Gu et al. utilised unmanned aerial vehicle video data for in-depth analysis of drivers’ crash risk at interchange merging areas [9]. Guo et al. obtained crash data from 367 freeway diverge areas in a three-year period and modelled a novel random parameters multivariate tobit model for evaluating risk factors on crash rates of different collision types [10]. Based on vehicle trajectory data analysis, Wang et al. proposed a combined usage of microscopic traffic simulation and extreme value theory for safety evaluation [11]. Li et al.

developed a model to predict vehicle trajectories in the straight-line and non-free flow state using historical trajectories and external parameters [12].

It can be found that the above named researches relied on empirical vehicle data, directly or indirectly. However, it is difficult for traditional traffic investigation, such as manual counting, floating car observation or earth coil collection, to obtain vehicle trajectory and speed data throughout observation areas. In recent years, with the advance of camera equipment, video observation has shown enormous potential in traffic surveys, but extraction of vehicle data from videos is still a challenge nowadays.

Many researchers were concerned about that problem and proposed their solutions. Kim et al. constructed frameworks for detecting vehicles from videos based on data-driven features [13]. Eliker et al. studied the reference flight trajectory generation and planning problems for quadcopter unmanned aerial vehicles [14]. Chen et al. presented a novel methodological framework for vehicle trajectory extraction from aerial videos and compared the extracted vehicle trajectories with manual calibrated data to testify the performance [15]. Feng et al. researched a method for vehicle trajectory construction from videos under mixed traffic conditions [16]. Lu et al. put forward a point-based tracking algorithm for trajectory extracting in traffic jams and complex weather conditions. [17]. In general, most findings focused on valid methods of extracting video vehicle data from one camera. However, the observation range of cameras is limited to ensure vehicles in videos are visible and clear. If the observation area is out of range, more than one camera ought to be used. In this case, an inevitable problem is how to stitch videos from different cameras and obtain integrated vehicle trajectories. To solve the problem, this paper proposed a methodological framework for extracting integral vehicle trajectories from several consecutive pictures automatically. The framework contains camera observation, eliminating image distortions, video stabilising, stitching images, identifying vehicles and tracking vehicles.

Another thing worth mentioning is that majority of current researches introduced their own methodologies for trajectory extracting. Partial studies compared extracted data with real data to verify the effectiveness of adopted methods. In fact, data accuracy is only one aspect of methodology evaluation. Reliability, operational time, random access memory (RAM) usage are all important indicators, especially for vehicle tracking, which is usually the key point in the whole methodological framework. Unfortunately, few investigations discussed this issue.

There are several classical object tracking algorithms. Camshift is a robust method of finding local extrema in the density distribution of a data set [18]. Sparse optical flow attempts to figure out where some points in an image have moved to in another image [19]. Regrettably, these algorithms expose latent problems when tracking vehicles. Camshift is unstable and may capture another nearby vehicle. Moreover, white vehicles can hardly be tracked by Camshift. In observation videos taken from high altitude, sometimes it is not easy to find trackable feature points, as well as corners, to apply sparse optical flow, especially in pure colour vehicles. Even worse, any occlusion, such as a road ramp or an indicator, could probably stop the tracking.

Recently, some novel tracking algorithms have attracted attention. Boosting algorithm is an online AdaBoost feature selection algorithm. The algorithm selects the most discriminating features for tracking depending on the background [20]. CSRT is based on the Discriminative Correlation Filter with channel and spatial reliability [21]. KCF uses a circulant structure of tracking-by-detection with kernels, which utilises properties of circulant matrix to improve processing speed [22]. Median flow detects the forward-backward error and selects reliable trajectories in video sequences [23]. MIL applies multiple instance learning instead of traditional supervised learning during tracking-by-detection to avoid incorrectly labelled training examples from slight inaccuracies in the tracker, which may result in drifting [24]. MOSSE trains a minimum output sum of squared error filter to adapt changes of the target object appearance in tracking [25]. TLD decomposes the long-term tracking task into tracking, learning and detection, which localises all appearances that have been observed so far and corrects the tracker if necessary [26].

Multiple tracking methods based on different theories cause selection hesitation. Even though most tracking methods do not limit their application scenes distinctly, diminutive object, limited clarity, lack of notable features and complex background of traffic videos probably cause trouble for vehicle tracking. Therefore, it is worth comparing the above-mentioned methods from various aspects and finding out the best one for traffic information collection. By testing, it is found that the TLD tracking region might shift to another vehicle, which has a similar shape and colour, and cause inevitable chaos. In this case, other six algorithms, including boosting, CSRT, KCF, median flow, MIL and MOSSE, are selected for further analysis based on empirical videos.

The paper is organised as follows. Video collection and processing methods are introduced in Section 2. Tracking algorithms comparison is presented in Section 3. Major findings of the paper are summarised in Section 4. The framework of the paper is shown as *Figure 1*.

2. VIDEO COLLECTION AND PROCESSING METHODS

Video collection and processing methods contain following steps: camera observation, eliminating image distortions, video stabilising, stitching images, identifying vehicles and tracking vehicles. Detailed descriptions are as follows.

2.1 Camera observation

For the sake of comparing the aforementioned tracking algorithms, a field observation was conducted in South Fengtai Road near Fenghuanghemei residential block, in Nanjing, Jiangsu Province, China. The observation area is shown in *Figure 2a*, which is a part of a city expressway and contains a weaving area in both directions. Complex road

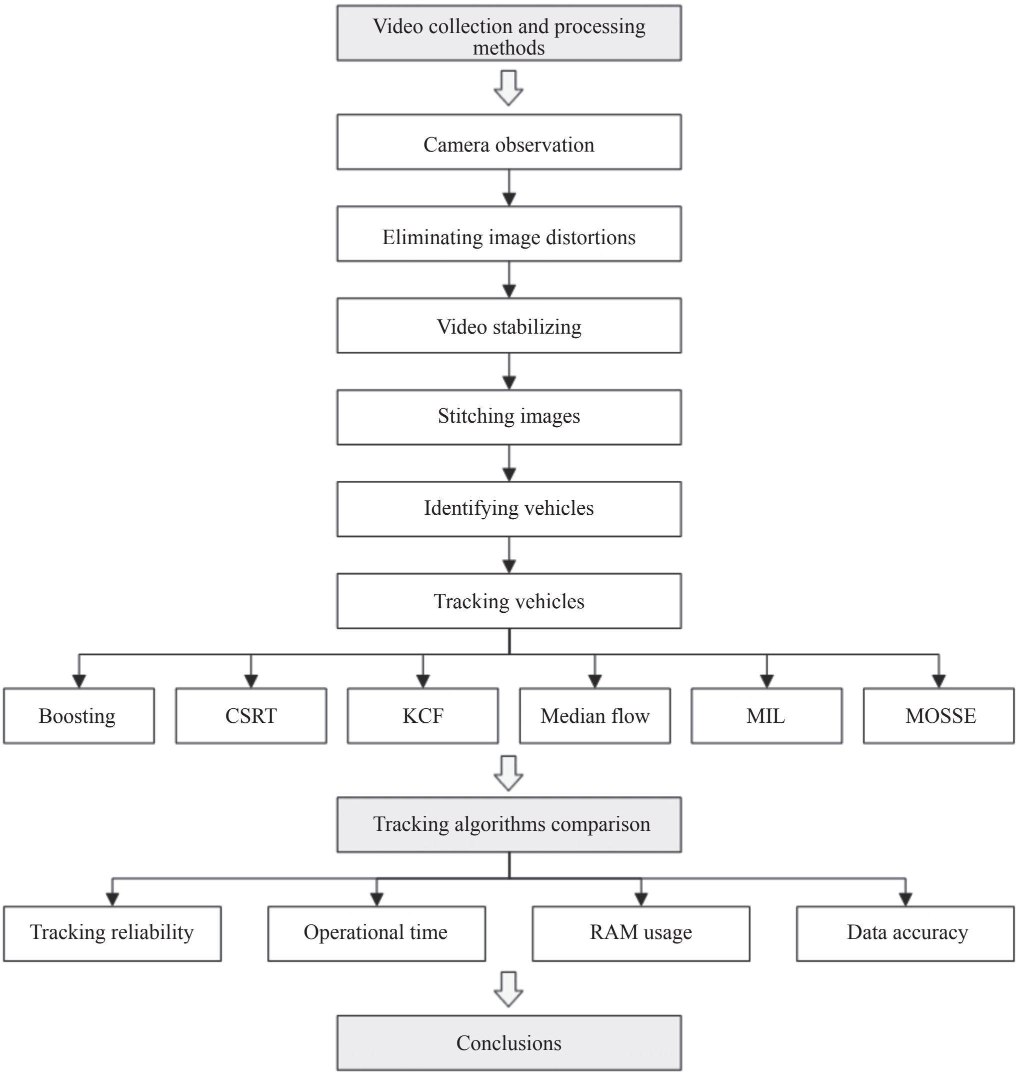
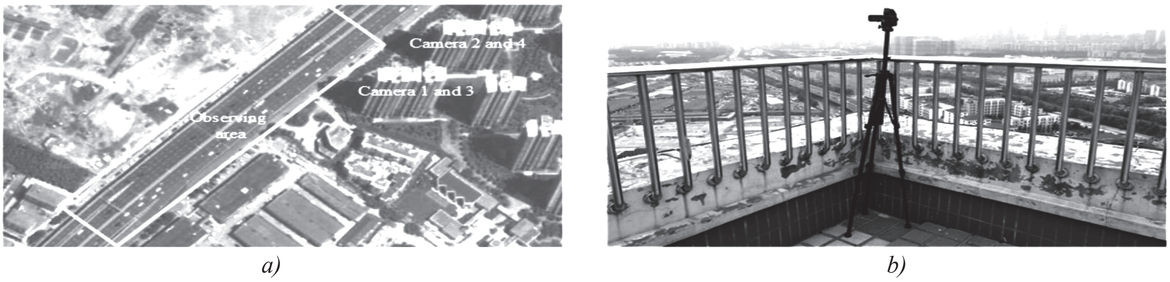


Figure 1 – The framework of the paper



a)

b)

Figure 2 – Observation positions (a) and one of the cameras (b)

structure and frequent vehicle lane changing behaviors bring challenges to vehicle identification and tracking. Therefore, it is helpful to find potential deficiencies of different tracking algorithms.

To cover the 460 meters long observation area, four cameras were set on the tops of two buildings just beside the road, which both had 34 floors and were about 100 meters above the road surface, as shown in *Figure 2b*.

2.2 Eliminating image distortions

Owing to the optical lenses used in cameras, distortions are inevitable, so images fail to show the actual positions of vehicles. Radial distortions and tangential distortions are main reasons for image distortions. Radial distortions arise as a result of the shape of the lens, whereas tangential distortions arise from the assembly process of the camera as a whole. To eliminate image distortions, *Equation 1* is adopted [27]

$$\begin{bmatrix}x_p\\y_p\end{bmatrix}=\left(1+k_1r^2+k_2r^4+k_3r^6\right)\begin{bmatrix}x_d\\y_d\end{bmatrix}+\begin{bmatrix}2p_1x_dy_d+p_2\left(r^2+2x_d^2\right)\\p_1\left(r^2+2y_d^2\right)+2p_2x_dy_d\end{bmatrix} \tag{1}$$

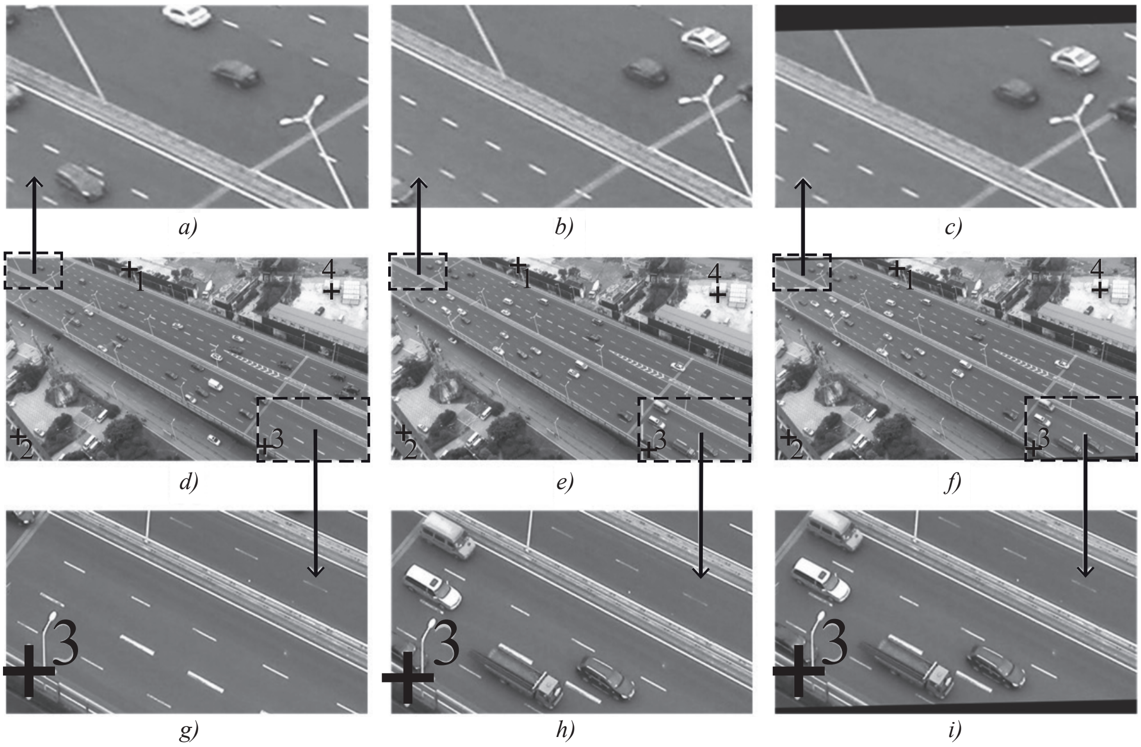
where k_1, k_2 and k_3 are parameters of radial distortion, p_1 and p_2 are parameters of tangential distortion, $\begin{bmatrix}x_d\\y_d\end{bmatrix}$ is the distortion coordinate of any point in the image and $\begin{bmatrix}x_p\\y_p\end{bmatrix}$ is the undistorted coordinate of the point. A calibration board like that in [27] is made to calculate the values of k_1, k_2, k_3, p_1 and p_2 of the cameras used in this research. In this case, undistorted coordinates of points in images can be calculated and undistorted images are available.

2.3 Video stabilising

Because the videos were shot from the rooftops of the buildings, as *Figure 2* shows, cameras were often shaken by the strong wind despite the fact that tripods were used. *Figure 3d* is the first frame of a camera video and *Figure 3e* is a subsequent frame from the same video. In order to distinguish the difference between them, the regions marked with dotted lines in *Figures 3d and 3e* are enlarged. It is worth noting that the positions and scales of the top left regions in *Figures 3d and 3e* are identical, and the bottom right regions in *Figures 3d and 3e* are also the same. *Figures 3a and 3b* are the enlarged views of the top left regions in *Figures 3d and 3e* respectively. *Figures 3g and 3h* are the enlarged views of the bottom right regions in *Figures 3d and 3e* respectively.

By comparison, it can be found that the visual angles of *Figures 3a and 3b* are different. For example, the distance between the image's top left corner and the left endpoint of the isolation barrier in *Figure 3a* is larger than that in *Figure 3b*. Besides, the nearest lane line at bottom left of the isolation barrier in *Figure 3a* has four visible sections, whereas five sections of the same lane line can be observed in *Figure 3b*. Analogously, the lane lines in *Figures 3g and 3h* manifest the difference between them. Though the differences seem to be insignificant in the whole view, meter-level errors arise since the length of one section of the lane line is about 2 meters and the interval between lane line sections is about 3 meters. Therefore, specific processes are necessary.

Obviously, the discrepancies arise because *Figures 3d and 3e* are recorded by the shaking camera at different visual angles. Based on the structure of the camera, images are projections of points in the physical world into the camera plane. The process of changing the visual angle of the subsequent frame to eliminate the discrepancies is a kind of image reprojection from one plane to another. As *Figure 4* shows, from the perspective of projective geometry, the reprojection can be regarded as mapping a convex quadrilateral to another, which is called perspective transformation [28].



a) b) c)

d) e) f)

g) h) i)

Figure 3 – Video stabilising: d) is the first frame of the video, e) is a subsequent frame of the video, f) is the processed subsequent frame, a)–c) are the enlarged views of the top left regions in d)–f) respectively, and g)–i) are the enlarged views of the bottom right regions in d)–f) respectively

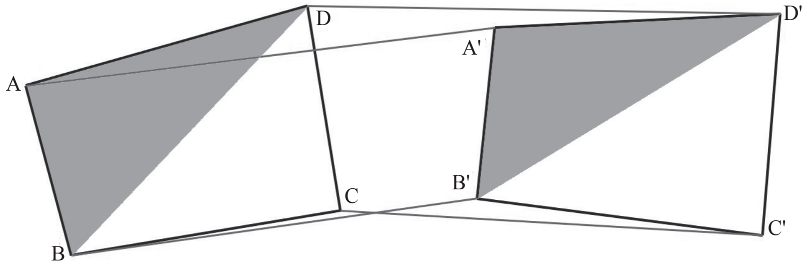


Figure 4 – Perspective transformation from ABCD to A'B'C'D'

According to the deduction in [27], the mathematic model of perspective transformation is as follows.

$$\begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \quad (2)$$

where $\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & 1 \end{bmatrix}$ is the projection matrix, $\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$ is the parameter of rotation, $\begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix}$is the parameter of translation, $[k_{31} \quad k_{32}]$ is the parameter of perspective, $\begin{bmatrix} x_o \\ y_o \end{bmatrix}$ is the coordinate of the point in the first frame and $\begin{bmatrix} x_s \\ y_s \end{bmatrix}$ is the coordinate of the same point in the subsequent frame.

Since coordinates of one point in the first and subsequent frame can provide two equations, an equation based on x coordinates and one based on y coordinates, at least four points are necessary for the eight unknowns in *Equation 2*. In order to avoid equivalent equations, any of the three points, which are in this paper referred to as mark points, should not lie in a straight line. In addition, the mark points near image edges show better results in general. Besides, mark points must be fixed in the physical world and visible throughout the videos. Examples of mark points are indicated with crosses in *Figure 3d*. A valid way to record positions of mark points is to click on them with a mouse and record the positions of the mouse pointer. When the camera shakes, mark points change positions in videos as well. Therefore, tracking mark points is essential to reduce the workload and manual errors. However, the artificially selected mark points are seldom applicable for computer tracking. The points focused on tracking should be nearly unique and parameterizable, and contain enough information to be picked out from one frame to the next. Since visible colour changing in two different directions usually appears near these points, they are called corners in computer vision. [29] provided a method to find corners and [30] raised improvements. Sometimes, the peak of colour changing does not occur at the centre of a pixel. To solve the problem, a common method is to fit the curve of image colour values and find the peak with mathematical calculations. The work is called subpixel corner detection, which was explained in detail in [31, 32].

When it comes to video stabilising described in this paper, the nearest subpixel corner around each mark point is chosen as focus. Sparse optical flow [19] is found to be reliable when tracking these subpixel corners since they are uncovered. Positions of the subpixel corners in subsequent frames are recorded, as shown in *Figure 3e*. Based on the coordinates of four subpixel corners in the first frame and the subsequent frame, the projection matrix in *Equation 2* can be worked out. If the projection matrix is applied to change positions of all points in the subsequent frame, the subsequent frame will be projected to the visual angle of the first frame, as *Figure 3f* shows. To distinguish the result, identical regions in *Figure 3f* are enlarged as *Figures 3c and 3i*. By comparison, it can be found that vehicles in *Figure 3c* are corresponding to those in *Figure 3b*, whereas the visual angle of *Figure 3c* is the same as that of *Figure 3a*, which is validated by positions of the isolation barrier left endpoint or visual sections of the lane lines. The black parts in *Figures 3c and 3i* present the portions of *Figure 3f* that are less than *Figure 3e*. Similarly, *Figure 3i* shows the identical scene as *Figure 3h* but analogous visual angle is as in *Figure 3g*. On the basis of above explained video stabilising process, all subsequent frames can be projected to the visual angle of the first frame and the effect of camera shaking will be eliminated.

2.4 Stitching images

To ensure the consistency of tracking vehicles, video images from 4 observation cameras are stitched together to compose the whole bird's-eye view [33], as shown in *Figure 5*. In detail, quadrangle areas in *Figures 5a, 5b, 5d and 5e* are projected to corresponding positions in the whole bird's-eye view image (*Figure 5c*) based on projection matrices calculated by *Equation 2* with positions of peak points of the quadrangle areas.

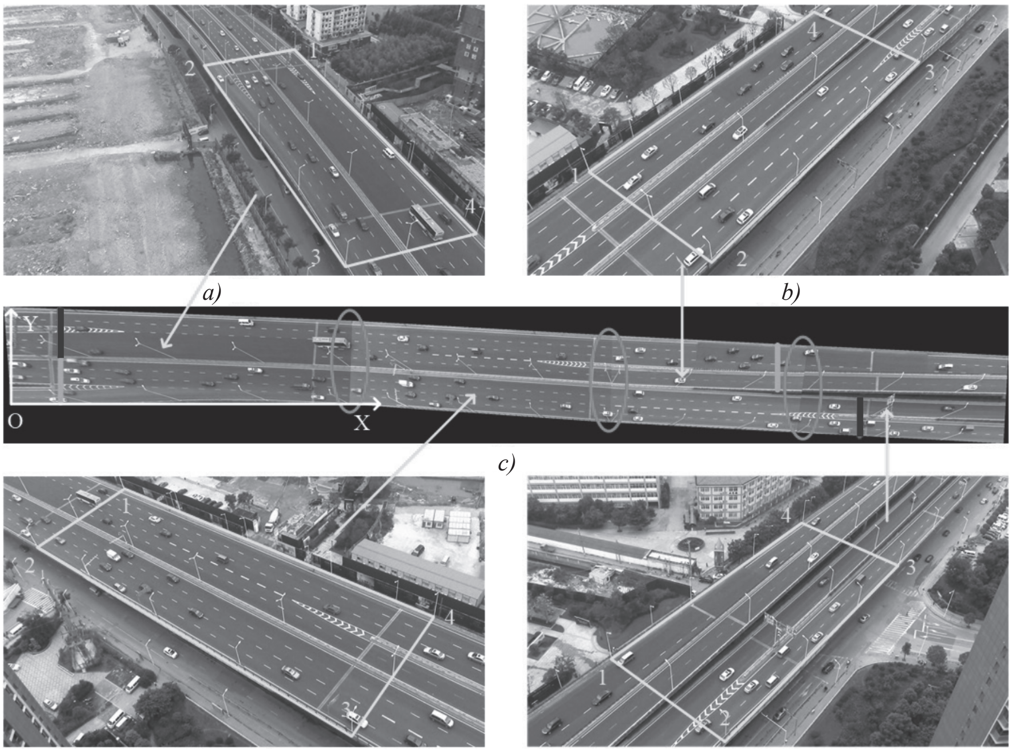
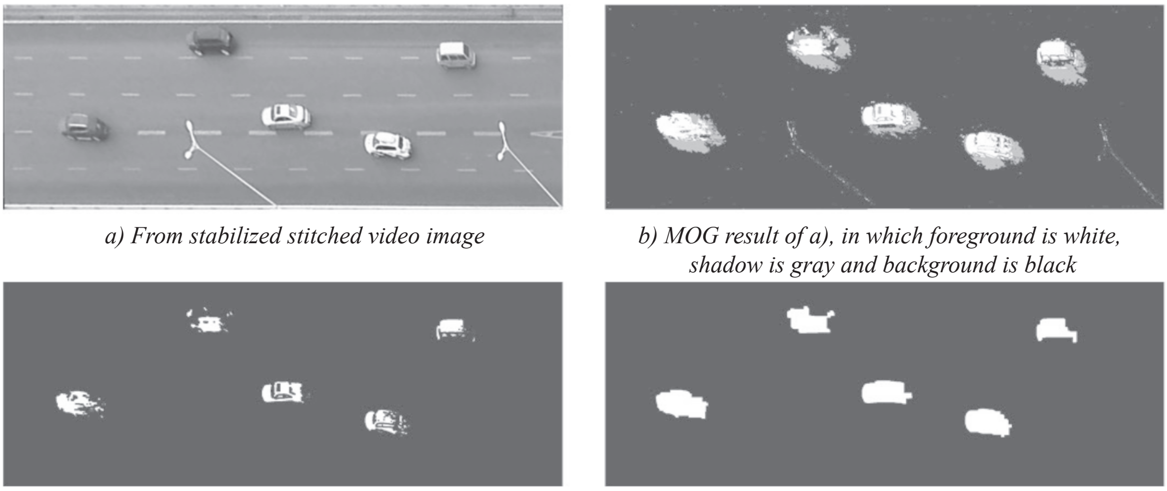


Figure 5 – Observed scene: a), b), d), e) are images from four cameras and c) represents the whole scene



a) From stabilized stitched video image

b) MOG result of a), in which foreground is white, shadow is gray and background is black

c) The erosion result of foreground in b)

d) The dilatation result of c)

Figure 6 – Identifying vehicles

2.5 Identifying vehicles

Based on stabilised stitched video images, the mixture of Gaussians (MOG) model, a statistical model marrying the average distance method and the codebook method [34, 35], is adopted to differentiate between background, shadow and foreground, as shown in *Figure 6b*. The foreground part (white part in *Figure 6b*) is regarded as potential vehicles and chosen for further analysis. It is eroded to remove noise (*Figure 6c*) and dilated to fill the blanks from MOG and erosion (*Figure 6d*). In addition, contours and shapes of the foreground part are extracted to help recognise vehicles. Based on aforementioned procedures, 95% vehicles can be identified.

2.6 Tracking vehicles

If an identified vehicle passes the light grey in *Figure 5c*, its position will be fed to tracking algorithms as the initial input. The vehicle will be tracked until it leaves the dark grey lines in *Figure 5c*. If one tracked vehicle is lost in the process, it will be deleted in time after the last position is recorded to avoid interference with other tracked vehicles.

3. TRACKING ALGORITHMS COMPARISON

The difference of tracking algorithms is analysed with respect to tracking reliability, operational time, random access memory (RAM) usage and data accuracy. To exclude the effect of irrelevant factors, a control group is included, in which the tracking of vehicles is ignored, whereas other functions, such as eliminating image distortions, video stabilising, stitching images and identifying vehicles are all reserved. The tests were conducted on a computer equipped with an Intel Core i5-6300HQ CPU @ 2.30GHz, a NVIDIV GeForce GTX 960M GPU, an 8.00 Gb RAM and a Windows 10 64bit system.

3.1 Tracking reliability

All above mentioned algorithms provide a rectangle to represent the most likely bounding box for the target during tracking. If the bounding box is detached from the tracked vehicle, the tracking of a vehicle is lost, as shown in *Figure 7*.

This section records the quantities of track losses in different areas when varied algorithms are applied. The results are shown in *Figures 8b-8g*, where numbers of colour bars indicate quantities of lost vehicles. To distinguish prominent interferences, a binarization diagram of the observation area highlighting zebra lines, lane lines, lamps, guide boards and stitching seams is shown in *Figure 8a* for further comparison. In addition, green lines and red lines in *Figure 8a* indicate where the tracking starts and ends respectively.

Considering the operational time of partial tracking algorithms, which will be introduced in Section 3.2, 10-minute stitched observation videos are intercepted for further analysis. It is found that dark coloured vehicles (DCV for short), such as black or dark grey ones, are usually difficult to be identified or tracked because their colours are close to those of road surface and shadow. Therefore, light coloured vehicles (LCV for short), such as

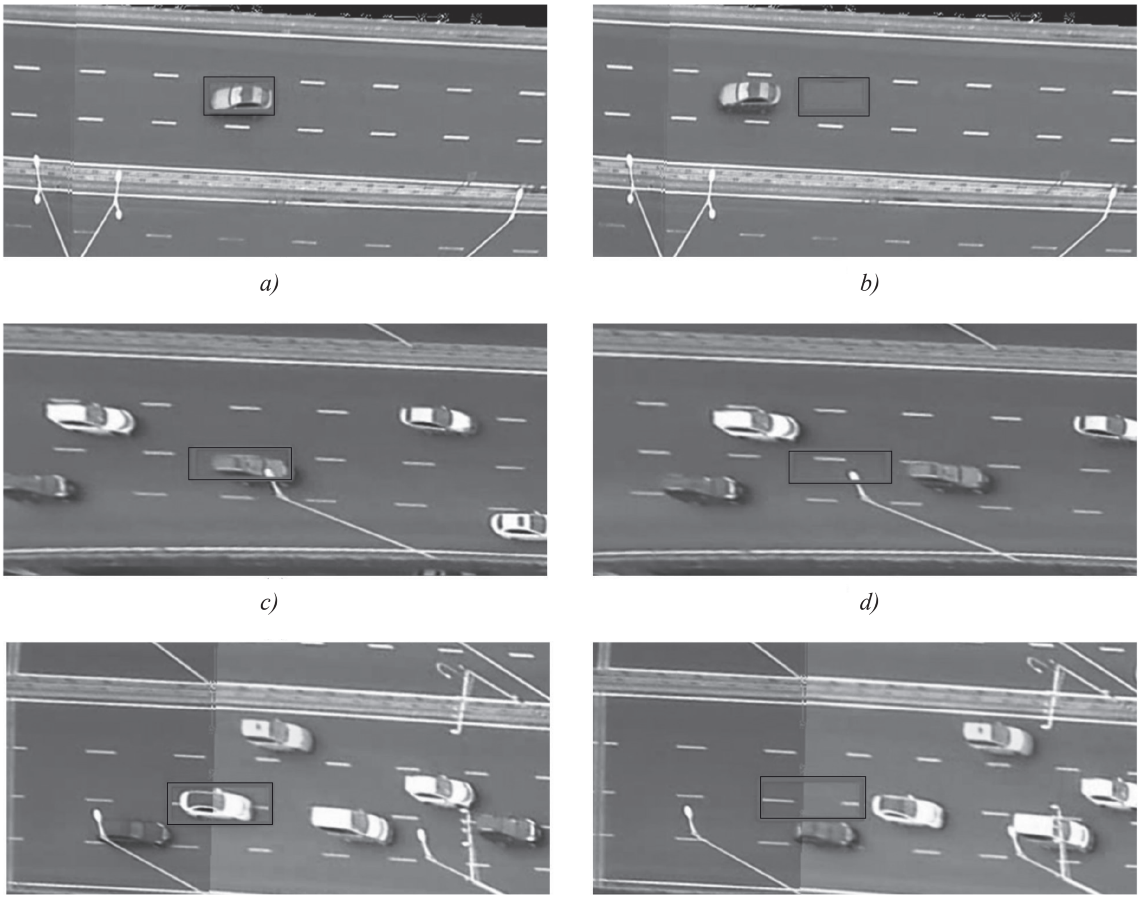


Figure 7 – Tracks are lost because of the lane line (a, b), the lamp (c, d) and the seam (e, f) respectively

white, silver, red, yellow or blue ones, and DCVs are analysed separately. During the 10-minute stitched observation video, 941 LCVs and 230 DCVs are observed in total. Interference factors which can be covered by vehicles, such as zebra lines and lane lines, are classified as background factors. Oppositely, interference factors which can cover vehicles, such as lamps and guide boards, are classified as foreground factors. In addition, image stitching seams are regarded as another kind of factors. Detailed track loss information is shown in *Table 1*.

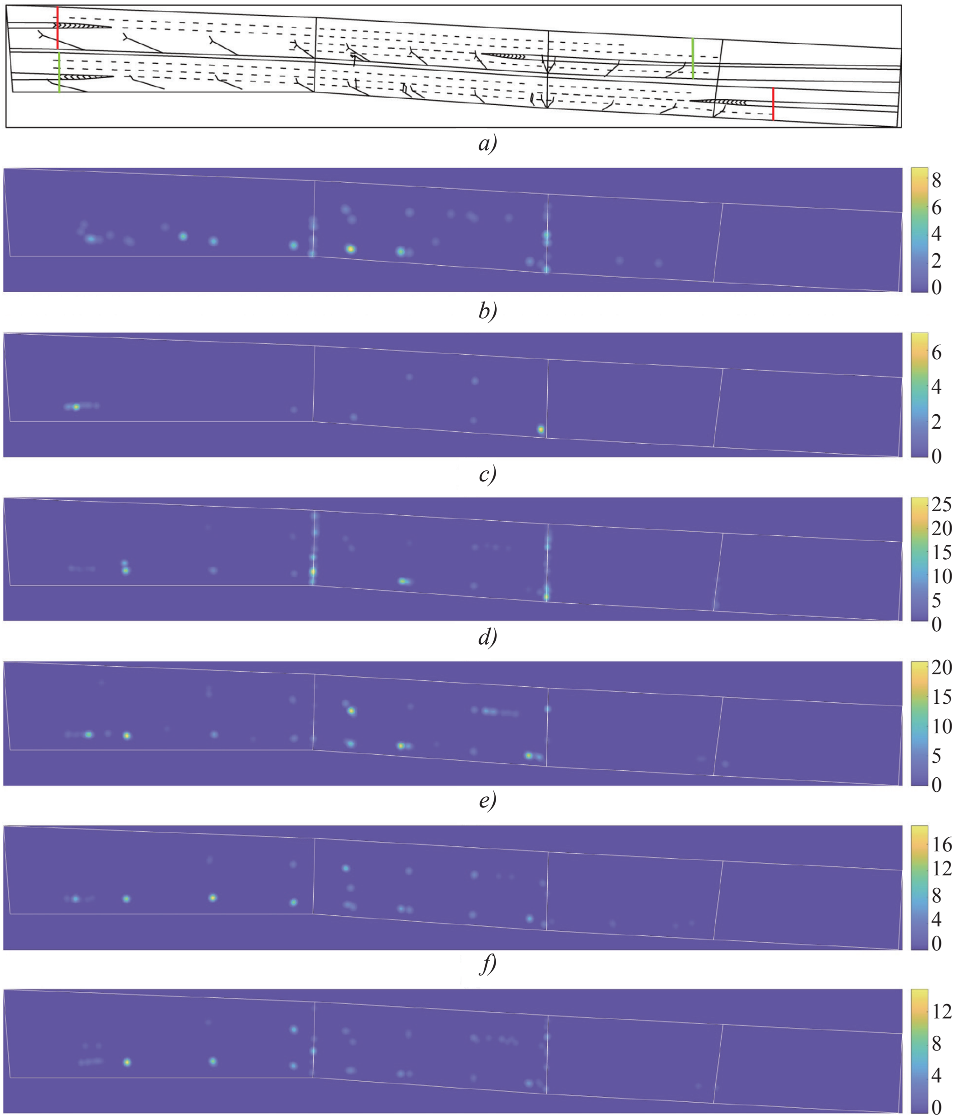
Table 1 – Track loss quantities of tracking algorithms

	Boosting	CSRT	KCF	Median Flow	MIL	MOSSE
Track loss quantities of LCVs caused by background	0	16	22	25	0	18
Track loss quantities of LCVs caused by foreground	28	0	46	132	24	26
Track loss quantities of LCVs caused by seams	0	0	155	0	0	21
Track loss quantities of DCVs caused by background	23	0	0	42	19	0
Track loss quantities of DCVs caused by foreground	17	18	67	71	153	64
Track loss quantities of DCVs caused by seams	33	0	101	20	0	10
Total amount of track loss	101	34	391	290	196	139

According to *Table 1*, CSRT has the fewest track losses in total, whereas KCF has the most. For visualised comparison, track loss probability is calculated based on *Equation 3*

$$p = \frac{m}{n} \tag{3}$$

where p is track loss probability, m is the quantity of track losses of LCVs or DCVs, and n is the amount of tracked LCVs or DCVs. For example, the track loss probability of LCVs affected by foreground is 28/941=2.98 %. All track loss probability is shown in *Figure 9*.



a)

b)

c)

d)

e)

f)

g)

Figure 8 – The observation area binarization diagram (a) and track loss quantities in different areas of boosting (b), CSRT (c), KCF (d), median flow (e), MIL (f) and MOSSE (g)

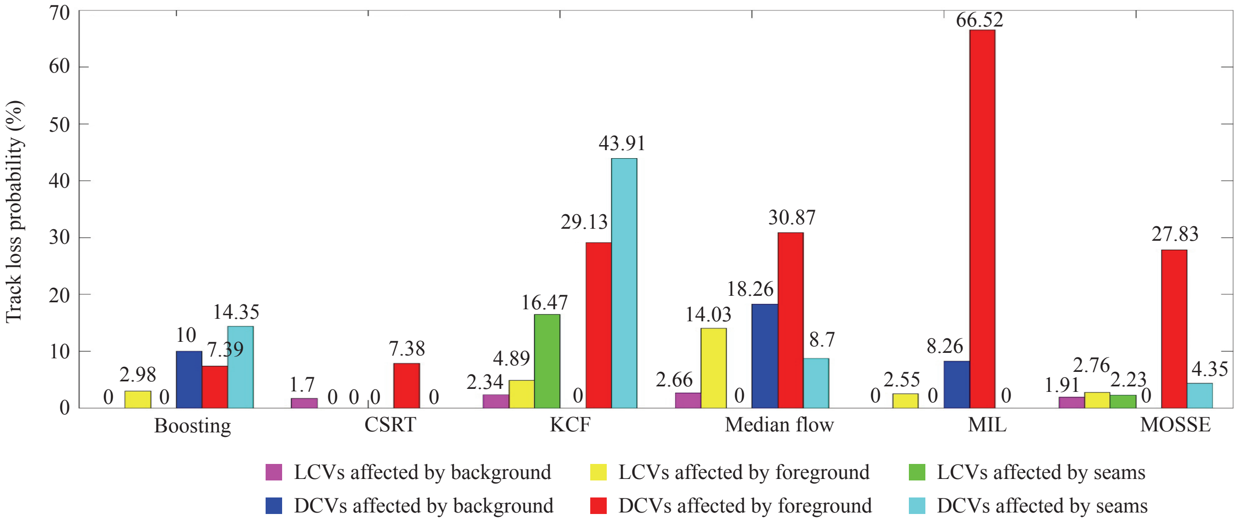


Figure 9 – Track loss probability of tracking algorithms

As *Figure 9* shows, all algorithms track LCVs better than DCVs, generally. Boosting, CSRT, MIL, MOSSE are reliable when tracking LCVs. KCF is more likely to lose track of LCVs at seams. When it tracks DCVs, the problem becomes even worse. Median flow is interfered with the foreground when tracking LCVs. All algorithms face challenges when DCVs are occluded. In this case, boosting and CSRT perform relatively well, whereas MIL loses tracks of about two-thirds of DCVs. Boosting, median flow and MIL are also unstable when DCVs are affected by background. However, CSRT, KCF and MOSSE overcome the problem if DCVs are identified. Boosting, median flow and MOSSE may lose tracks of DCVs at seams, though they are better than KCF to some extent. It also can be found that if any kind of algorithm has trouble in tracking LCVs through the foreground or at seams, it will have even worse problems when tracking DCVs.

3.2 Operational time

The study reveals that different algorithms differ greatly in operational time. It is worth noting that there are about 50 vehicles tracked together on average. As *Table 2* shows, none means no tracking algorithm takes effect and the time cost is mainly due to eliminating image distortions, video stabilising, stitching images and identifying vehicles. Tracking time, which shows the actual operational time of tracking algorithms, is calculated by subtracting the total time of none from that of each algorithm. Obviously, MOSSE has the fastest running speed, whereas MIL needs the longest time. Boosting and CSRT take almost the same amount of time, which is longer than that of KCF or median flow. If the tracking time of MOSSE is assumed as 1, expenditures of other algorithms range from 25.05 to 367.52.

3.3 RAM usage

As for the RAM usage, different algorithms also show individual features. As *Table 3* shows, extra average RAM usage is equal to average RAM usage subtracting that of none. MOSSE becomes the focus again for the least extra average RAM usage. CSRT costs a little more. Boosting needs the most RAM, whereas KCF, median flow and MIL consume more RAM than CSRT and less RAM than boosting. If extra average RAM usage of MOSSE is assumed as 1, those of the others range from 1.83 to 60.58.

3.4 Data accuracy

For the sake of data accuracy, a rectangular plane coordinate system in the road surface is established as the white arrows in *Figure 5c*. The origin is the no. 2 point in *Figure 5a* and the x axis goes through the no. 3 point in *Figure 5a*. A portable high precision GPS device, whose positional error is 8 mm, was used to measure GPS coordinates of the quadrangle areas peak points in *Figures 5a, 5b, 5d and 5e*. The image scale can be calculated by dividing pixel distance of the quadrangle areas peak points

Table 2 – Operational time of tracking algorithms

Tracking Algorithms	Boosting	CSRT	KCF	Median Flow	MIL	MOSSE	None
Total Time [h:mm:ss]	5:40:01	6:19:44	1:48:42	2:40:36	15:42:02	0:50:10	0:47:44
Tracking Time [h:mm:ss]	4:52:17	5:32:00	1:00:58	1:52:52	14:54:18	0:02:26	0:00:00
Tracking Time Relative Rate	120.11	136.44	25.05	46.38	367.52	1	/

Table 3 – RAM usage of tracking algorithms

Tracking Algorithms	Boosting	CSRT	KCF	Median Flow	MIL	MOSSE	None
Average RAM Usage [Mb]	1217	512	566	625	914	502	490
Extra Average RAM Usage [Mb]	727	22	76	135	424	12	0
Extra Average RAM Usage Relative Rate	60.58	1.83	6.33	11.25	35.33	1	/

into their physical distance. Since vehicles move in the plane of the quadrangle areas approximately, their physical displacement can be calculated with their pixel displacement multiplied by the image scale. The vehicle position in the road surface coordinate system is denoted as (x, y). Since the frame rates of observation videos are 25 Hz and the data intervals are 0.04 s, vehicle speed can be calculated with *Equation 4*

$$v_t = \frac{\sqrt{(x_{t+\Delta t} - x_{t-\Delta t})^2 + (y_{t+\Delta t} - y_{t-\Delta t})^2}}{2dt} \tag{4}$$

where v_t is the vehicle speed at time t, Δt is the time interval of data, $(x_{t+\Delta t}, y_{t+\Delta t})$ is the vehicle position at $(t+\Delta t)$ s, and $(x_{t\text{-}\Delta t}, y_{t\text{-}\Delta_t})$ is the vehicle position at $(t\text{-}\Delta t)$ s.

To acquire empirical data, a car equipped with the portable GPS device went through the observed section several times. Trajectory and speed data of the vehicle were collected from videos by different tracking algorithms. The data from steady tracking were selected to be compared with the data from the GPS device on the car.

Since the GPS device used the WGS84 coordinate system, GPS data were transformed to points in the road surface coordinate system by *Equation 5*. GPS coordinates and road coordinates of special fixed points in videos, such as stitching mark points, were measured to compute the parameters. where (x_r, y_r) is the position of point in the road surface coordinate system, (x_g, y_g, z_g) is the position of point in the WGS84 coordinate system, Δx, Δy, Δz are distance of translation and α, β, γ are angles of rotation.

Another fact worth noting is that the sampling frequency of the GPS device is 10 Hz, which means the data interval of GPS is 0.1 s and it differs from those of the videos. To calculate the deviation between them, 0.2 s is taken as the least common multiple data interval. For convenience, GPS data are marked as $g(1)$, $g(2)$, $g(3)$ and so forth. Video data are marked as $o(1)$, $o(2)$, $o(3)$ and so forth. To find the best data match, GPS data marked as $g(2i+i_g)$ are matched with video data marked as $o(5i+i_o)$ for calculating deviations, where i, i_g and i_o are natural numbers, i ranges from 1 to the number of data with 0.2 s interval, i_g is the order of the first GPS data adopted and ranges from 1 to 2, i_o is the order of the first video data adopted and ranges from 1 to 5. Therefore, there are 10 kinds of possible match, including $g(2_i+1)$ with $o(5i+1)$, $g(2i+1)$ with $o(5i+2)$, $g(2i+1)$ with $o(5i+3)$, $g(2i+1)$ with $o(5i+4)$, $g(2i+1)$ with $o(5i+5)$, $g(2i+2)$ with $o(5i+1)$, $g(2i+2)$ with $o(5i+2)$, $g(2i+2)$ with $o(5i+3)$, $g(2i+2)$ with $o(5i+4)$ and $g(2i+2)$ with $o(5i+5)$. The data matches are shown in *Figure 10*.

During numerical calculation, mean deviation values of x, y and v are computed based on *Equations 6–8* and the comprehensive mean deviation value M is calculated with *Equation 9*. The minimum value of M is worked out with *Equation 10*, and the data match with M_{min} is chosen as the best one. In other words, the particular data match of $g(2i+i_g)$ with $o(5i+i_o)$ resulting in M_{min} presents minimum deviation between GPS data and video data. In addition, the 85th percentile of the deviation value, which is greater than 85% of the data and less than the other 15%, and the standard deviation are calculated for more details.

$$\overline{dx}_{io,ig} = \frac{1}{q}\sum_{i=0}^{q} \left| x_{o(5i+io)} - x_{g(2i+ig)} \right| \tag{6}$$

$$\overline{dy}_{io,ig} = \frac{1}{q}\sum_{i=0}^{q} \left| y_{o(5i+io)} - y_{g(2i+ig)} \right| \tag{7}$$

$$\overline{dv}_{io,ig} = \frac{1}{q}\sum_{i=0}^{q} \left| v_{o(5i+io)} - v_{g(2i+ig)} \right| \tag{8}$$

$$M_{io,ig} = \frac{1}{3}\left[\overline{dx}_{io,ig} + \overline{dy}_{io,ig} + \overline{dv}_{io,ig} \right] \tag{9}$$

$$\begin{aligned} &M_{min} = \\ &min\{M_{1,1}, M_{2,1}, M_{3,1}, M_{4,1}, M_{5,1}, M_{1,2}, M_{2,2}, M_{3,2}, M_{4,2}, M_{5,2}\} \end{aligned} \tag{10}$$

where q is the number of data with 0.2 s interval, i is a natural number and ranges from 1 to q, i_o is the order of the first video data adopted and ranges from 1 to 5, i_g is the order of the first GPS data adopted and ranges from 1 to 2, $\overline{dx}$ is the mean deviation value of x, $\overline{dy}$ is the mean deviation value of y, $\overline{dv}$ is the mean deviation value of v, M is the comprehensive mean deviation value of x, y and v, and M_{min} is the minimum value of M.

The results of data error are shown in *Table 4*. Though there are some differences between dx, dy and dv, the gaps are narrow.

Furthermore, the study analyses the effect of image stitching seams on data accuracy, which are marked with blue ellipses in *Figure 5c*. If the vehicle passes a seam at time t, data from one second

$$\begin{bmatrix} x_r \\ y_r \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos a \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{5}$$

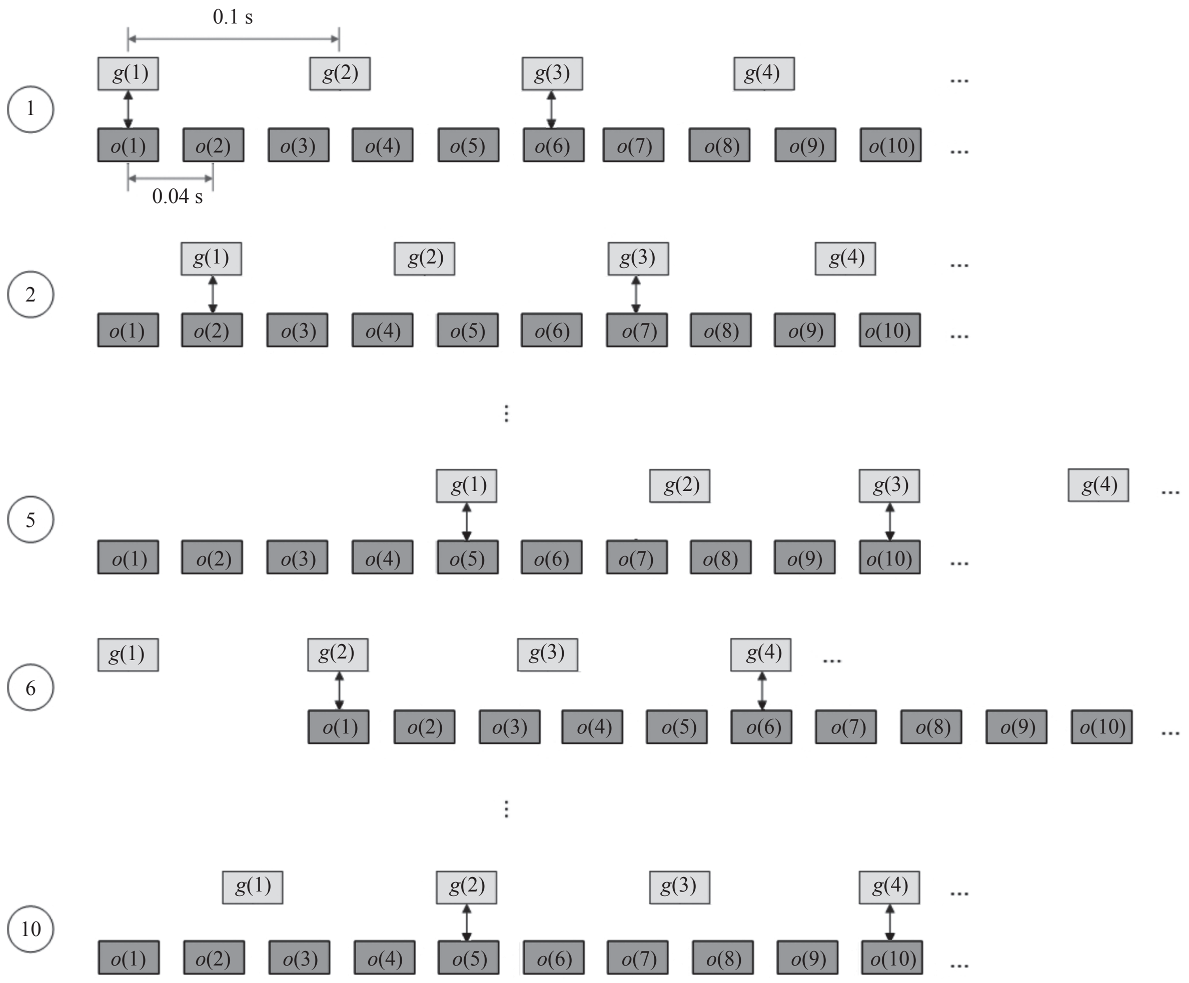


Figure 10 – The matches of GPS data (light grey) and video data (dark grey)

Table 4 – Position and speed error of integral video vehicle data

	Boosting	CSRT	KCF	Median flow	MIL	MOSSE
Mean of *dx* [m]	0.68	0.61	0.96	0.70	0.61	0.92
Standard deviation of *dx* [m]	0.42	0.48	0.41	0.43	0.46	0.44
The 85th percentile of *dx* [m]	1.22	1.18	1.47	1.16	1.17	1.54
Mean of *dy* [m]	0.65	0.47	0.61	0.54	0.76	0.64
Standard deviation of *dy* [m]	0.39	0.35	0.39	0.38	0.33	0.37
The 85th percentile of *dy* [m]	1.10	0.92	1.08	1.04	1.15	1.07
Mean of *dv* [m/s]	0.48	0.56	0.48	0.31	0.45	0.41
Standard deviation of *dv* [m/s]	0.34	0.57	0.35	0.23	0.35	0.31
The 85th percentile of *dv* [m/s]	0.85	0.96	0.93	0.55	0.88	0.67

before *t* to one second after *t* are collected as data near seams. The rest are categorised as data of the vehicle staying away from seams. Increases in position and speed error of vehicle data near seams compared with data away from seams are calculated and shown as *Table 5*. In *Table 5*, the minimum and maximum of each line are listed. In contrast with *Table 4*, the results in *Table 5* do not exceed the range of potential random errors. In other words, no obvious deviation increase is observed when

Table 5 – Increases in position and speed error of vehicle data near seams compared with data away from seams

	Boosting	CSRT	KCF	Median flow	MIL	MOSSE	Minimum	Maximum
Mean of *dx* [m]	0.26	0.03	0.67	-0.01	0.25	0.68	-0.01	0.68
Standard deviation of *dx* [m]	0.33	0.38	0.15	0.22	0.31	0.21	0.15	0.38
The 85th percentile of *dx* [m]	0.70	0.37	0.90	0.27	0.55	1.00	0.27	1.00
Mean of *dy* [m]	-0.46	-0.39	-0.48	-0.46	-0.37	-0.4	-0.48	-0.37
Standard deviation of *dy* [m]	0.38	0.29	0.35	0.33	0.32	0.35	0.29	0.38
The 85th percentile of dy [m]	0.01	0.05	-0.06	0.02	0.03	0.01	-0.06	0.05
Mean of *dv* [m/s]	-0.09	-0.36	-0.10	-0.08	0.00	-0.04	-0.36	0.00
Standard deviation of *dv* [m/s]	-0.23	-0.91	-0.18	-0.05	-0.11	-0.16	-0.91	-0.05
The 85th percentile of *dv* [m/s]	-0.03	0.06	-0.32	-0.16	-0.35	-0.48	-0.48	0.06

considering the vehicle passing seams. Therefore, image stitching seams have little influence on data accuracy.

Based on the above analysis, it can be concluded that if the vehicle is tracked steadily, there will be little difference between the above algorithms in the error of position or speed. In this case, all algorithms produce reliable trajectory and speed data.

4. CONCLUSIONS

In this paper, an integral framework and main algorithms of vehicle information collection from several consecutive videos are introduced. A practical and programmable method of video stabilising is explained in detail. As key points, six novel tracking algorithms, including boosting, CSRT, KCF, median flow, MIL and MOSSE, are compared in terms of tracking reliability, operational time, RAM usage and data accuracy based on empirical observation videos. According to data analysis, it is found that MOSSE has the best running efficiency, the least RAM demand and medium reliability. CSRT shows optimum reliability in the cost of time, and its RAM usage is more than that of MOSSE and less than those of the others. In addition, if observation videos are from the same camera, image stitching can be ignored. If so, boosting and KCF could have better performance. Even though MIL performed well when tracking LCVs, long time cost and high RAM usage may limit its application. In general, MOSSE is worth trying especially when there are few DCVs. If high-performance computers are used, CSRT is probably preferred for its prominent reliability. In addition, if vehicles are tracked steadily, all tracking algorithms can extract reliable trajectory and speed data.

In the study, it is found that different weather and sunlight influence vehicle identification and tracking, especially in respect of road surface colour and object shadow. Therefore, the difference of algorithms in special environments and possible improvement will be our research contents in the near future.

ACKNOWLEDGEMENTS

The research was sponsored by the National Key Research and Development Program of China (No. 2019YFB1600200), the Natural Science Foundation of Jiangsu Province (No. SBK2021041144), China Postdoctoral Science Foundation (No.2020M681466) and Jiangsu Planned Projects for Postdoctoral Research Funds (No. BK20210249).

陈全，博士研究生[1]
电子邮箱：quanchenseu@foxmail.com
王昊，教授[1]
（通讯作者）
电子邮箱：haowang@seu.edu.cn
董长印，博士后[1]
电子邮箱：dongcy@seu.edu.cn
1 城市智能交通江苏省重点实验室
现代城市交通技术江苏高校协同创新中心
东南大学交通学院
中华人民共和国江苏省南京市东南大学路2号，
211189

从连续视频中提取完整轨迹的车辆跟踪算法实测分析

摘要

本研究介绍了一种新的方法论框架，用于从几张连续的图片中自动提取车辆的完整轨迹。该框架由摄像机观测、图像畸变消除、视频稳定、图像拼接、车辆识别和车辆跟踪部分组成。本文以中国江苏省南京市凤台南路四个路段的观测视频为例，对该框架进行了验证分析。作为关键点，本文比较了六种典型的跟踪算法，包括*Boosting*、*CSRT*、*KCF*、*Median Flow*、*MIL*和*MOSSE*，在跟踪可靠性、运行时耗、内存开销和数据精确度方面的差异，并考虑了多种干扰因素的影响，包括车辆颜色、斑马线、车道线、路灯、指示牌和图像拼接接缝。通过实测分析，我们发现*MOSSE*的时间和内存开销最低，而*CSRT*则表现出最佳的跟踪可靠性。此外，我们还发现如果车辆被稳定地跟踪，所有跟踪算法都能获得可靠的车辆轨迹和速度数据。

关键词

视频观测；完整轨迹提取；车辆跟踪

REFERENCES

[1] Li M, Li Z, Xu C, Liu T. Short-term prediction of safety and operation impacts of lane changes in oscillations with empirical vehicle trajectories. *Accident Analysis & Prevention*. 2020;135: 105345. doi: 10.1016/j.aap.2019.105345.

[2] Xu C, Zhao J, Liu P. A geographically weighted regression approach to investigate the effects of traffic conditions and road characteristics on air pollutant emissions. *Journal of Cleaner Production*. 2019;239: 118084. doi: 10.1016/j.jclepro.2019.118084.

[3] Wang H, Qin Y, Wang W, Chen J. Stability of CACC-manual heterogeneous vehicular flow with partial CACC performance degrading. *Transportmetrica B: Transport Dynamics*. 2019;7(1): 788-813. doi: 10.1080/21680566.2018.1517058.

[4] Liu Z, Liu Y, Meng Q, Cheng Q. A tailored machine learning approach for urban transport network flow estimation. *Transportation Research Part C: Emerging Technologies*. 2019;108: 130-150. doi: 10.1016/j.trc.2019.09.006.

[5] Wang C, Xu C, Dai Y. A crash prediction method based on bivariate extreme value theory and video-based vehicle trajectory data. *Accident Analysis & Prevention*. 2019;123: 365-373. doi: 10.1016/j.aap.2018.12.013.

[6] Zhao YC, et al. Driving rule extraction based on cognitive behavior analysis. *Journal of Central South University*. 2020;27(1): 164-179. doi: 10.1007/s11771-020-4286-1.

[7] Sun C, et al. Multi-criteria user equilibrium model considering travel time, travel time reliability and distance. *Transportation Research Part D: Transport and Environment*. 2019;66: 3-12. doi: 10.1016/j.trd.2017.03.002.

[8] Wang C, Sun Z, Ye Z. On-road bus emission comparison for diverse locations and fuel types in real-world operation conditions. *Sustainability*. 2020;12(5): 1798. doi: 10.3390/su12051798.

[9] Gu X, et al. Utilizing UAV video data for in-depth analysis of drivers' crash risk at interchange merging areas. *Accident Analysis & Prevention*. 2019;123: 159-169. doi: 10.1016/j.aap.2018.11.010.

[10] Guo Y, Li Z, Liu P, Wu Y. Modeling correlation and heterogeneity in crash rates by collision types using full Bayesian random parameters multivariate Tobit model. *Accident Analysis & Prevention*. 2019;128: 164-174. doi: 10.1016/j.aap.2019.04.013.

[11] Wang C, et al. A combined use of microscopic traffic simulation and extreme value methods for traffic safety evaluation. *Transportation Research Part C: Emerging Technologies*. 2018;90: 281-291. doi: 10.1016/j.trc.2018.03.011.

[12] Li XC, Fan LK, Chen T, Guo CS. Vehicle lane-changes trajectory prediction model considering external parameters. *Promet – Traffic&Transportation*. 2021;33(5): 745-754. doi: 10.7307/ptt.v33i5.3718.

[13] Kim EJ, et al. Extracting vehicle trajectories using unmanned aerial vehicles in congested traffic conditions. *Journal of Advanced Transportation*. 2019; 9060797. doi: 10.1155/2019/9060797.

[14] Eliker K, Zhang GQ, Grouni S, Zhang WD. An optimization problem for quadcopter reference flight trajectory generation. *Journal of Advanced Transportation*. 2018; 6574183. doi: 10.1155/2018/6574183.

[15] Chen XQ, et al. High-resolution vehicle trajectory extraction and denoising from aerial videos. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(5): 3190-3202. doi: 10.1109/TITS.2020.3003782.

[16] Feng RY, Fan CY, Li ZB, Chen XQ. Mixed road user trajectory extraction from moving aerial videos based on convolution neural network detection. *IEEE Access*. 2020;8: 43508-43519. doi: 10.1109/ACCESS.2020.2976890.

[17] Lu SN, Song HS, Hua C, Wang GF. A point-based tracking algorithm for vehicle trajectories in complex environment. *International Conference on Intelligent Systems Design and Engineering Applications*. 2014. p. 69-73. doi: 10.1109/ISDEA.2014.24.

[18] Bradski G. Computer vision face tracking for use in a perceptual user interface. *Proceedings of IEEE Workshop on Applications of Computer Vision*. 1998. p. 214-219.

[19] Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. *Proceedings of the International Joint Conference on Artificial Intelligence*. 1981;2: 674-679.

[20] Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting. *Proceedings of the British Machine Vision Conference*. 2006;6: 1-10. doi: 10.5244/C.20.6.

[21] Lukežič A, et al. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*. 2018;126: 671-688. doi: 10.1007/s11263-017-1061-3.

[22] Henriques JF, Caseiro R, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. *Proceedings of the European Conference on Computer Vision*. 2012;4: 702-715. doi: 10.1007/978-3-642-33765-9_50.

[23] Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: Automatic detection of tracking failures. *IEEE*

International Conference on Pattern Recognition. 2010. p. 2756-2759. doi: 10.1109/ICPR.2010.675.

[24] Babenko B, Yang MH, Belongie S. Visual tracking with online multiple instance learning. *IEEE Conference on Computer Vision and Pattern Recognition*. 2009. p. 983-990. doi: 10.1109/CVPR.2009.5206737.

[25] Bolme DS, Beveridge JR, Draper BA, Lui YM. Visual object tracking using adaptive correlation filters. *IEEE Conference on Computer Vision and Pattern Recognition*. 2010. p. 2544-2550. doi: 10.1109/CVPR.2010.5539960.

[26] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on *Pattern Analysis and Machine Intelligence*. 2012;34(7): 1409-1422. doi: 10.1109/TPAMI.2011.239.

[27] Kaehler A, Bradski G. *Learning OpenCV 3*. O'Reilly Media, Inc. California, US; 2017. p. 644-671.

[28] Semple J, Kneebone G. Algebraic projective geometry. Oxford, UK: Oxford University Press; 1979.

[29] Harris C, Stephens M. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*. 1988. p. 147-151.

[30] Shi J, Tomasi C. Good features to track. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994. p. 593-600. doi: 10.1109/CVPR.1994.323794.

[31] Lucchese L, Mitra SK. Using saddle points for subpixel feature detection in camera calibration targets. *Asia-Pacific Conference on Circuits and Systems*. 2002;2: 191-195. doi: 10.1109/APCCAS.2002.1115151.

[32] Chen D, Zhang G. A new sub-pixel detector for x-corners in camera calibration targets. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 2005. p. 97-100.

[33] Chen Q, Wang H, Dong CY. Modeling lane-changing behaviors in merging areas of urban expressways in Nanjing, China. *Transportation Research Record*. 2020;2674(7): 480-493. doi: 10.1177/0361198120923361.

[34] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of the IEEE International Conference on Pattern Recognition*. 2004;2: 28-31. doi: 10.1109/ICPR.2004.1333992.

[35] Zivkovic Z, Heijden FV. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*. 2006:27(7): 773-780. doi: 10.1016/j.patrec.2005.11.005.