Soft Computing (2020) 24:1283–1291
https://doi.org/10.1007/s00500-019-03965-w

METHODOLOGIES AND APPLICATION

A high accurate vehicle speed estimation method

Shengnan Lu[1,2] · Yuping Wang[2] · Huansheng Song[3]

Published online: 9 April 2019

Abstract
In this paper, we present a novel approach for accurate vehicle speed estimation from video sequences. Common methods usually track sets of distinguishing features; however, feature extraction is a difficult task in dynamic environments. Herein, we propose a novel analysis method without feature extraction. Initially, a frame difference method is applied to a region of interest, from which projection histograms are obtained and a group of key bins are selected to represent the vehicle motion. Then, all the possible speeds are tested one by one, and the extreme value of the testing function is selected for the corresponding speed. The proposed system was tested on three data sets containing 2054 vehicles, where the ground truth of speed is obtained by a radar speed detector. The experiment results show that the proposed system has an average error of 0.3 km/h, with 99.4% of the estimation speed within the error of range (− 2 km/h, 2 km/h). The system turns out to be robust, accurate and real time for practical use.

Keywords Vehicle speed estimation · Camera calibration · Projection histogram · Speed enumeration

1 Introduction

As an important parameter, vehicle speed is widely used for automatic analysis of urban traffic in recent years. This case is due, in part, to the use of various sensing modalities, including radar, lidar, loop detectors and visual surveillance. As cameras are becoming cheaper, easier to install and of higher quality than ever before, video-based system has become the main means for vehicle speed estimation.

Over the few past decades, with the development of video sensing and computational technologies, vehicle speed estimation using computer vision has been an extremely active research area. In the meantime, computing power has greatly increased, and many advanced hardware platforms have emerged, such as multicore processing and graphical processing unit (GPU) (Banz et al. 2011; Homm et al. 2010), which make it possible to carry out vehicle speed measurement in real time. In addition, new analysis techniques based on vehicle speed estimation have enabled new applications such as vehicle behavior analysis, traffic condition prediction and autonomous driving.

In recent years, several video-based methods have been proposed for vehicle speed estimation. Many of such methods tracked sets of distinguishing features extracted from vehicle regions, and vehicle speed was estimated by comparing the trajectories of the tracked features to known real-world distances. A variety of features, such as blobs (Madasu and Hanmandlu 2010; Maduro et al. 2008), corners (Song et al. 2014; Dogan et al. 2010), edges (Zhiwei et al. 2007; Dailey et al. 2000), image patches (Grammatikopoulos et al. 2005), license plates (Luvizon et al. 2016; Llorca et al. 2016) or a combination of such features (Palaio et al. 2009) are often used for vehicle tracking. Unfortunately, due to a lack of vehicle height information, these features on vehicles have a limited capability to determine the lane, as shown in Fig. 1. The trajectory, indicated by the red line, is not actually located on the surface of the road but "floating" above the road. The length of trajectory is obviously shorter than the lane determination distance. For larger vehicles like the truck or motor bus, the lane determination will surely fail. As a result, vehicle speed estimation will be inaccurate using these trajectories.

In this paper, instead of obtaining motion vector by tracking features on vehicles, we use the shadow beneath the vehicle, which is considered very close to the ground

Communicated by V. Loia.

✉ Shengnan Lu
lushengnan@xsyu.edu.cn

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

[2] Tulane University, New Orleans, US

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

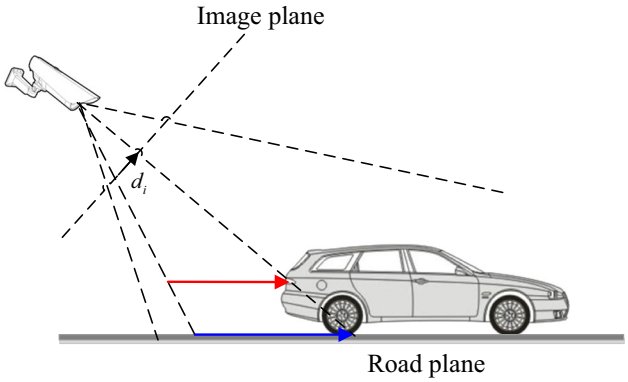


Fig. 1 Vehicle speed estimation scheme

truth speed. In order to estimate vehicle speed without modeling 3D space, or requiring simple camera calibration, a few assumptions are made about the scene in the beginning:

1. The input video is captured by a single camera, fixed and located overhead so that the shadow beneath the vehicle is clearly visible, as shown in Fig. 2;
2. The movement of vehicles is in constant manner;
3. Each lane lies in a plane;
4. The lane line on the road is a straight line;
5. The shadow beneath the vehicle is at approximately the same distance as the ground one.

In reality, vehicle feature extraction is prone to be affected by the moving status of vehicle and dynamic environment, such as rainy and snowy days, illumination variations and shadow effects, which will render it be difficult, resulting in low accuracy of speed estimation. In this paper, we adopt a new analysis method without feature extraction. The approach is based on enumeration, which can avoid the above problems and improve the accuracy of speed estimation.

Figure 3 illustrates the flowchart of the proposed approach. First, a region of interest (ROI) containing the moving vehicle is set. A frame difference method is applied

Fig. 2 Sample image from our system

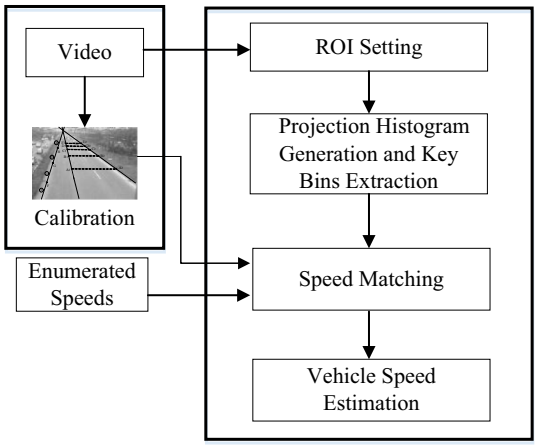


Fig. 3 Flowchart of the proposed system

to this region, which contains the motion information of vehicles. The projection histograms are then obtained from the difference images, and a group of key bins are selected to represent the vehicle movement tendency. Because the vehicle speeds are discrete, all the possible speeds are enumerated as a discrete set, and are tested one by one. Finally, the extreme value of the testing function is selected with the corresponding speed as final output. Actually, this method is the inverse process of traditional one for vehicle speed estimation. Instead of extracting features, we consider the feature differences among the images with different speeds and construct the mapping relationship with extreme value corresponding to the vehicle speed.

A preliminary version of the system (Lu et al. 2015) described here was published at International Journal of Smart Home. The system described here differs from that version in several aspects, such as camera calibration from one dimension to two dimension, which is more accurate and efficient for speed computation. The new system was also evaluated more thoroughly.

The rest of this paper is organized as follows. The related work is discussed in Sect. 2. Both camera calibration and speed estimation are introduced in Sects. 3 and 4, respectively. Finally, experimental results are presented in Sects. 5 and the conclusions are given in Sects. 6.

2 Related work

2.1 Vehicle speed estimation and measurement

In the past few years, many video-based methods for vehicle speed estimation were proposed. Many approaches for this task include motion vehicle segmenting using background subtraction (Jeyabharathi and Dejey 2016), frame differences (Zhiwei et al. 2007), tracking the whole vehicle or local features of vehicles, such as blobs (Madasu

and Hanmandlu 2010; Maduro et al. 2008), corners (Song et al. 2014; Dogan et al. 2010), edges (Zhiwei et al. 2007; Dailey et al. 2000), image patches (Grammatikopoulos et al. 2005) and license plates (Luvizon et al. 2016; Llorca et al. 2016). The speed is estimated from the displacement of vehicle between two features using inverse perspective mapping with a flat world assumption for the road. Then, a scale factor is obtained by camera calibration (Llorca et al. 2016). Most of the approaches are applied to the traffic scenes with a single fixed traffic camera covering two lanes or more, with low focal lengths, detecting vehicles at a large distance. However, some exceptions are found in recent works (Dogan et al. 2010; Cinzburg et al. 2015; Lin et al. 2008). Optical flow vectors are used to transform into space magnitudes after camera calibration in Dogan et al. (2010) and Lan et al. (2014), with side view images and traffic images, respectively. The work in Cinzburg et al. (2015) just relies on a laptop and a consumer web camera for traffic video capturing. The situation with one blurred image is considered in Lin et al. (2008) to estimate the relative movement of the vehicle.

Accordingly, a successful vision-based vehicle speed estimation system should meet the following requirements: tracking vehicle stably under challenging conditions, with accurate camera calibration, and with real-time operation and low cost. Methods based on blobs, edges and image patches are sensitive to conditions such as illumination variation, shadow and perspective. Our previous work (Song et al. 2014) developed a point tracking approach for traffic jams and complex weather conditions. Luvizon et al. (2016) proposed a blob tracking method based on a particle filter, similar to the one proposed by Maduro et al. (2008). Zhiwen et al. (2007) detected each blob feature such as Laplacian and color after background subtraction. In practice, tracking vehicle stably under complex traffic environments is still a challenge in computer vision. In this paper, we adopt a new analysis strategy without feature extraction instead.

Moreover, there is an important issue that has been overlooked. In practice, it is hard to track features on the car when touching the road. As a result, the plane on which the features of vehicle move is not the road plane, but a false one parallel to the road plane. A common way to solve this problem is to compute a correction factor that depends on the height of vehicle features over the road. The work from Luvizon et al. (2016) relies on vehicle license plate as the features, and they set a constant factor S for solving the problem of the actual license plate above the road plane. However, the constant factor S will bring larger errors when the vehicle license plate location is high. A similar method was proposed by Ginzburg et al. (2015), which is also based on vehicle license plate tracking, where they computed the compensation factor using the knowledge of the license plate such as the standard dimensions. In our work, we will use the shadow beneath the vehicle, which is considered to be very close to the ground truth speed.

3 Camera calibration

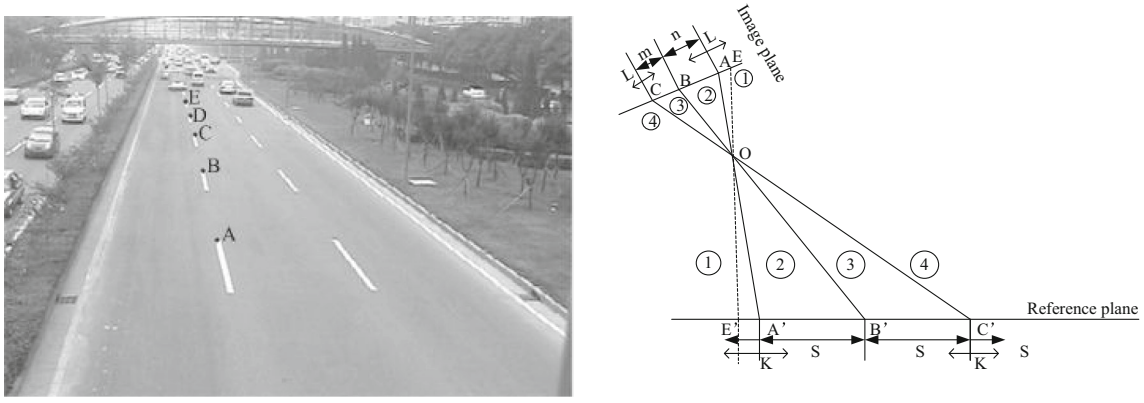
Specifically, we assume that each road lane lies on an approximately flat plane. We call the plane the reference plane. This assumption makes it possible for us to map the image plane to the reference plane, which will map every point in the image plane to the location in the real world.

In our previous work, one-dimensional calibration was proposed with the fixed points on lane markings in Song et al. (2014). We relied on the fact that the locations of the points on the lane markings are visible in the image and known priori, as shown in Fig. 4a; a mapping from pixels in the image to coordinates in the real world can be established based on the known length of pavement markings and lane width. Figure 4b shows the projective model of one-dimensional calibration in Song et al. (2014). However, this work assumed that the distance in the reference plane mapped from pixels in each row is approximately equal; unfortunately, this assumption cannot be guaranteed when the lane is far away from the center line of the camera in the horizontal direction. This is because one-dimensional calibration just calculated projective relation between the rows in the image plane and the distances in the reference plane. In order to improve the calibration accuracy, a two-dimensional method is proposed with further details below.

3.1 2D calibration based on 1D calibration

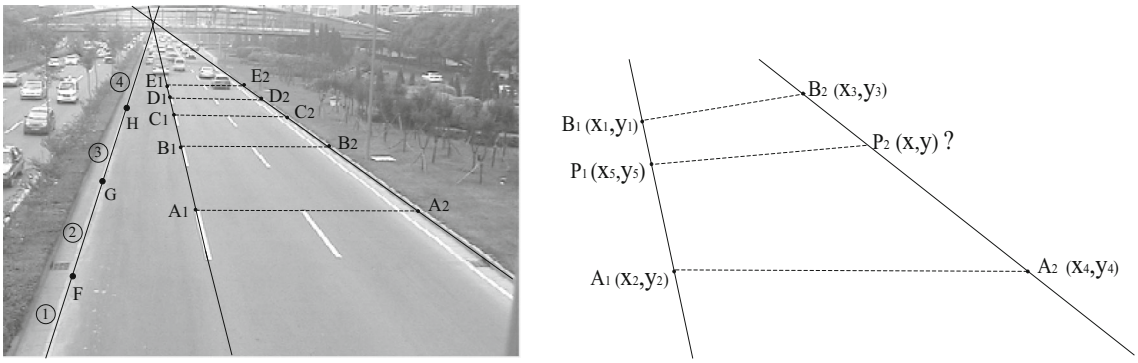
Assume that we have obtained two mapping relationships from one-dimensional calibration mentioned above: R–D form (from the rows in the image plane to the distance in the reference plane) and D–R form (from the distance in the reference plane to the rows in the image plane). The two-dimensional calibration method will provide the relations between the pixels in the image plane and the distance in the reference plane, which is based on the one-dimensional calibration results.

Similar to one-dimensional calibration, some visible calibration markings on the road are also needed in the two-dimensional calibration. We manually select five pairs of points on the road, as shown in Fig. 5a, where we consider that the segments A_1A_2, B_1B_2, C_1C_2, D_1D_2 and E_1E_2 are all equal and parallel in the real world. However, they may not be equal and parallel in the image plane because of an inclined visual angle. So, we need to assign the same distance to every point on the same segment. Note



(a) (b)

Fig. 4 One-dimensional calibration method. **a** The location of lane markings in the image. **b** Projective model of one-dimensional calibration



(a) (b)

Fig. 5 Two-dimensional calibration method. **a** The location of lane markings in the image. **b** Method of the lane line coordinates calibration

that the middle lane line is just toward the camera in Fig. 5a, and the distances of points on the middle lane line in the reference plane are regarded as the real distance. That is, the distance in the reference plane for A_1A_2 is obtained from the coordinates of point A_1 in the image plane and the R–D form. With the same method, we can get the distances of other segments B_1B_2, C_1C_2, D_1D_2 and E_1E_2 in the reference plane.

Then, how can we calculate the rest points in the image, which is not on the calibration segments? As shown in Fig. 5b, A_1B_1 is not equal to A_2B_2, and if we want to get the point P_2, which is not with point P_1 in the same row, we can make use of the proportions of segments. From the proportions of $B_1P_1/B_1A_1 = B_2P_2/B_2A_2$, the point P_2 in the image plane can be obtained by

$$\begin{cases} x = x_3 - \dfrac{(x_1 - x_5)(x_3 - x_4)}{(x_1 - x_2)} \\ y = y_3 - \dfrac{(y_1 - y_5)(y_3 - y_4)}{(y_1 - y_2)} \end{cases} \quad (1)$$

Then, all the points on the segment P_1P_2 will be assigned the same distance in the reference plane.

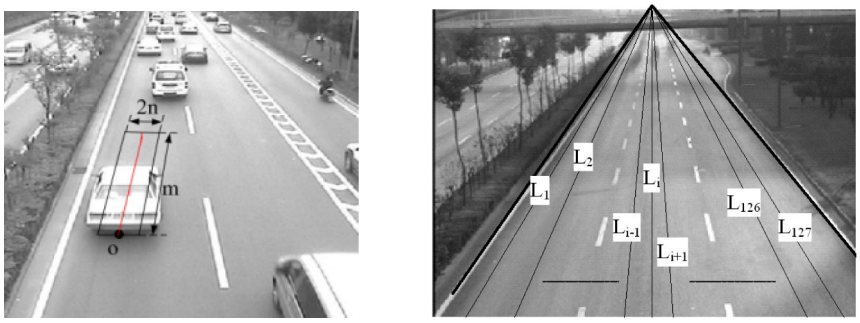
According to the method above, we can obtain the distances of all the points which are just on the segment. Some points may be omitted, so we need to scan every point in the image. An adjacent value-filling method is adopted for the point with no value. We scan every point line by line, and the point with no value will be filled with the first nonzero value on the right. Then, the relation between the pixels in the image plane and the distance in the reference plane is built, which is saved as P–D form. Meanwhile, a D–P form is derived from the P–D form, which contains the mapping relationships from the distance in the reference plane to the point in the image plane.

4 Speed estimation

4.1 Region of interest setting

In this paper, we will obtain motion information of vehicles without feature extraction and tracking. The first step is to set a region of interest, which is known as tracking zone in

Fig. 6 ROI setting



(a) Tracking line and tracking zone setting **(b)** Detection line

the following text, which will limit further processing to an area of target vehicle. This process is done in a few steps:

First, select the initial tracking point manually at the rear of the vehicle, and the best position is on the beneath shadow of the moving vehicle, which is denoted as point $O(x, y)$ in Fig. 6a.

Second, set the tracking line and tracking zone. As shown in Fig. 6b, the road region has been divided along the traffic direction into 128 subregions by 127 detection lines. Tracking line is determined by these pre-defined detection lines, and the nearest one to the point O is selected, shown as the red line in Fig. 6a. Then, a rectangle region of $m*2n$ pixels is generated dynamically, which starts from point O, and centered along the determined tracking line. The region is shown as tracking zone in Fig. 6a.

We expect that the tracking zone will contain the moving distance of vehicle in several consecutive frames, so the value of m is mainly determined by the frame numbers, and the value of $2n$ is less than the width of the vehicle. Here, $m = 90$ and $n = 8$ are applied in detail below.

4.2 Image motion detection

Image motion detection begins with a frame difference method, and we extract consecutive 10 video frames in the tracking zone, as shown in Fig. 7. Then, the difference value of the tracking zone is obtained between frames at time $k + n$ and k using the frame difference method. Here, in order to obtain obvious moving features of the vehicle, we perform an interval frame difference method, where $n = 2$. Consequently, there will be 8 difference images obtained from the 10 consecutive frames.

Second, we use a horizontal projection histogram method on these 8 difference images. We accumulate the 16 pixel values in each row, generating a histogram with 90 bins (for 90 rows). This histogram is shown as Fig. 8, which includes 8 projection histograms of 8 difference images. It can be seen that there is a pattern in the projection histograms, which keeps shifting steadily with vehicle movement along moving direction. We need to find a key bin with almost fixed relative position and value in this pattern, which will represent the vehicle movement tendency. Here, we use a 2D array $G[k][r]$ to represent the projection value of the row r in the kth difference image.

In our previous work, the maximum of bins is selected as the key bin, however, the maximum is susceptible to noise, which will lead to unreliable location of the key bin. To solve this problem, we present a novel method for the key bin selection—Max ratio. Using a definite proportion of the sum of 90 bin values in the first difference image as a

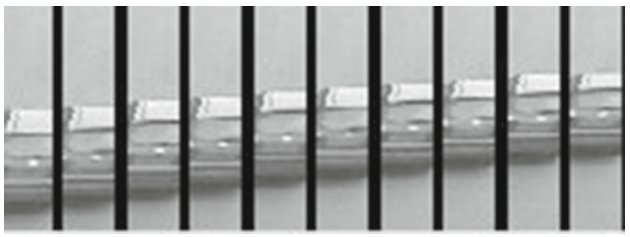


Fig. 7 Ten video frames in tracking zone

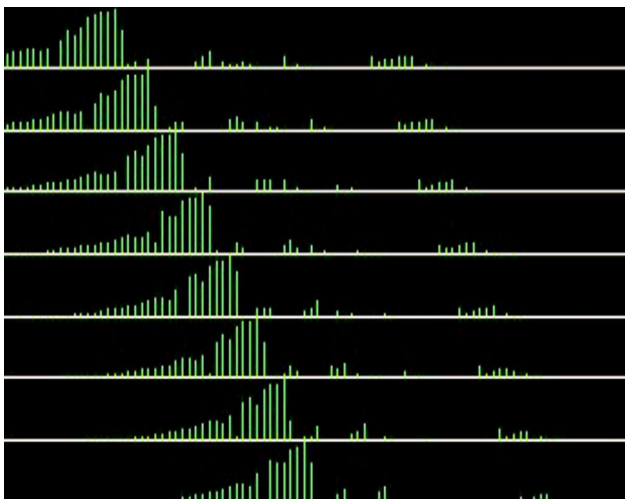


Fig. 8 Projection histograms of eight difference images

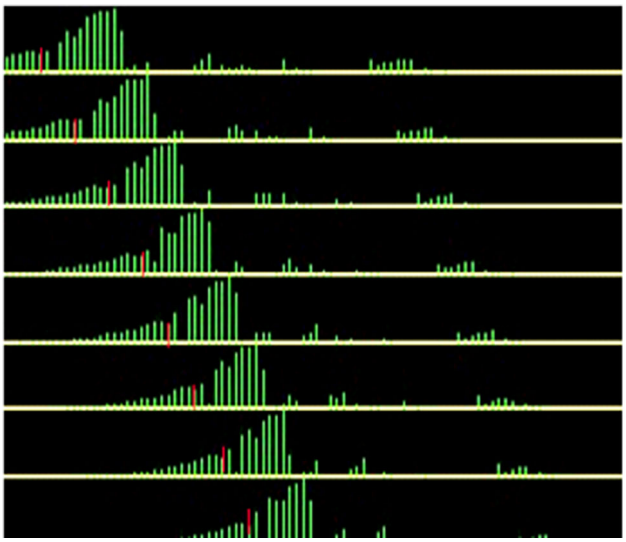


Fig. 9 The key bin selection using Max-ratio method

threshold, the key bin is selected according to the accumulation of every bin value one by one in the following difference images. Once the accumulation value meets the threshold, the current bin is selected as the key one. In our test, we assign 0.05 to the definite proportion. The key bin is selected as in Fig. 9. Then, the row of key bin in the image can be obtained by the initial position of point O and the line of the key bin in the tracking zone. Consequently, we will get 8 rows and they are saved in an array $R_I[8]$.

4.3 Speed estimation

In the last step, we have obtained the rows of 8 key bins. However, we cannot estimate vehicle speed just from each displacement between key bins. The selection process of the key bins is rough, which is hard to represent the movement of vehicles accurately. In this section, we will present a new algorithm for speed computation.

The main idea of this method is to enumerate all the possible speeds as a discrete speed set $\{v^{(i)}\} = \{1, \ldots, 180\}$, considering the tolerance as 1 km/h. For each speed, we track the point O among 8 frames in the reference plane, and then 8 new positions will be generated. According to these positions, we will find 8 corresponding bins in Fig. 8. Repeating the process above, 180 groups of bin values will be obtained. Then, we need to find which group is the closest one to the 8 key bins, and the corresponding speed will be the real speed of target vehicle. Note that we select the point O as the initial tracking point, which is on the beneath shadow of the target vehicle. It can represent the real speed of the moving vehicle. Our tests also show that the selection of the tracking point is crucial to the result of speed estimation.

To reduce the processing time when repeating to test for the best speed, the speed computation is divided into two

```
1: function Speed coarse estimation(R_I[1,2,…,8])
2:   for each m∈{1,2,…,8} do
3:     R_W[m] ←R-D(R_I[m]);
4:   end for
5:   for each i∈{1,2,3,…,36} do
6:     for each n∈{0,1,…,7} do
7:       s_n←5iΔtn+s_0;
8:       j←D-R(s_n);
9:       k←R_I[n+1];
10:      S[i]←S[i] +(G[n][k]- G[n][j])²;
11:    end for
12:  end for
13:  for each i∈{1,2,3,…,36} do
14:    if min(S[i]) then
15:      s←5*i;
16:    end if
17:  end for
18:  return s;
19: end function
```

Fig. 10 Routine to speed coarse estimation

parts: coarse speed estimation and fine speed estimation. The first step of coarse estimation enumerates speeds from 5 to 180 with an interval of 5. In accordance with the above method, a best speed will be found. Then based on the coarse speed, the fine estimation is performed around the best speed, which will enumerate speed with an interval of 1. At last, a final speed will be determined.

The function of speed coarse estimation is to roughly estimate the vehicle speed with a certain interval. The routine is shown in Fig. 10. It receives parameters as the rows of the key bins. It returns an enumerated speed. First, we compute the rows of the key bins in the real world, denoted by $R_W[8]$. The displacement in meters is calculated according to the equation at step 7 of Fig. 10, where 5 is the speed interval, i is the number of enumerated speed, s_0 is the row of point O in the reference plane, Δt is the frame interval, and n is the number of frames. From this equation, we can get 8 positions along the tracking line, and these positions are turned into the rows in the tracking zone. Accordingly, the projection value of each row can be obtained from $G[k][r]$, as shown in step 9. Finally, comparing these 8 projection values [denoted as $G[n][j]$] with the key bin values [denoted as $G[n][k]$] to find the closest one, which is thought as the real vehicle speed. We use the sum of squared differences to measure the distance between $G[n][j]$ and $G[n][k]$, and the minimum value corresponds to the real vehicle speed, which can be obtained by $5*i$, as shown in step 15.

After the coarsely estimated speed is obtained, we perform the fine estimation around this speed v. The principle of fine estimation is the same as coarse estimation mentioned above. The only difference is that the interval of enumerated speed is 1. Based on the speed v, 30 enumerated speeds from $v - 15$ to $v + 14$ are tested one by one, and a final speed will be determined. Here, we adopt a compute strategy implementing different compute process in different phase (coarse and fine estimation), which can improve the efficiency and accuracy of calculation, and can also guarantee the real time and precision of the system.

5 Experiments

A measurement system was built for evaluating the proposed method. We used a Pentium 4 3.2-GHz central processing unit (CPU) with 4-GB random access memory (RAM) computer, and the system was developed using Visual C++ on a raw video format. In the next sections, our testing scenario and the performance of speed estimation are described.

5.1 Testing scenarios

To demonstrate the robustness and validity of the proposed method, we have devised a test scenario at the south of 2nd ring road of Xi'an in China, using a radar speed detector at the roadside, which is captured by a low-cost CMOS image sensor with the resolution of 720 × 288, at 25 frames per second. The speed ground truth is measured by the radar speed detector. As shown in Table 1, the videos are classified as 3 sets with different weather conditions. The radar speed detector is on the right side of the road, and the ground truth speed sometimes failed to properly assign a speed to a vehicle when several vehicles entering the speed acquisition area at the same time. So we will ignore these vehicles with overlap. The "No. valid" column in Table 1 indicates the number of vehicles which had a valid assigned speed.

5.2 Speed estimation evaluation

The performance of the proposed approach is evaluated by comparing the speed estimated by our system with the ground truth speeds obtained by the radar speed detector. According to the precision of the radar speed detector, the tolerance is set to ± 2 km/h. If the difference between the speed detected by the proposed method and the one detected by radar is lower than or equal to the tolerance, the speed can be thought to be correct. Speed error distribution is given regarding the valid vehicles. They are divided into 3 parts, depending on whether the estimated speed is below, inside or above the tolerance. Figure 11 shows the distribution of speed estimation error. The maximum estimation error is − 4 km/h and 3 km/h, with 99.4% of the estimations being inside of the tolerance. The average error is 0.31 km/h with a standard deviation 0.82 km/h.

Table 1 Data set information

Data set	Time	No. vehicle	No. valid
1	78	2091	1273
2	15	547	219
3	32	894	562

In order to verify that the proposed method is better than our previous version (Lu et al. 2015), we performed tests on the same 3 data sets. The results are shown in Table 2. The accuracy ratio for vehicle speed is calculated based on the correct detection count for speed (DCV) and the total target count (TTC). Within the error tolerance, all 2054 vehicles were tested and the speeds of 2042 were correctly detected. The speed detection accuracy ratio can reach 99.4%, which indicates that two-dimensional calibration has a higher precision than one-dimensional calibration.

In order to verify if feature extraction from the vehicle is a good choice for speed estimation, we also compared our system with a point-based tracker (Song et al. 2014), which is our previous work for vehicle tracking. Obviously, the accuracy ratio of the proposed system is much higher than the method with a point-based tracker. That happens because in this case the features have a larger variance in their heights, which will make the same vehicle to have different motion vectors. Besides, this feature is not significant enough in complex traffic environment, leading to tracking error.

Moreover, we compared our system with a method mentioned in reference (Wu and Juang 2012) for vehicle speed estimation as shown in Table 3. The two methods are tested by the same speed detector with different error tolerance. It can be seen that the proposed method shows excellent accuracy for vehicle speed estimation.

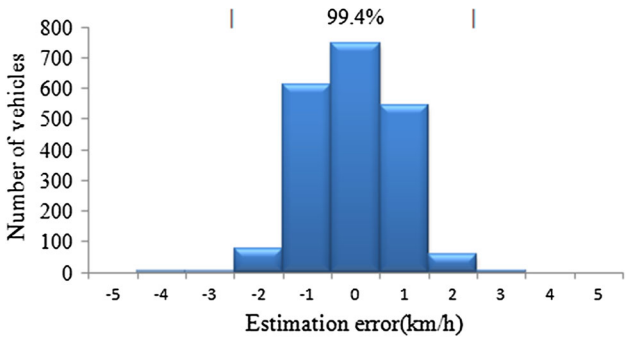


Fig. 11 Speed estimation error distribution

Table 2 A comparison with pervious version

Data set	Proposed system			Previous version		
	1	2	3	1	2	3
DCV/TTC (*)	1270/1273	217/219	555/562	1245/1273	213/219	542/562
Accuracy ratio (%)	99.8	99.3	98.8	97.8	97.4	96.5

(*) DCV is the correct detection count for speed; (*) TTC is the total target count

Table 3 A comparison with other approaches

Data set	Proposed system			A point-based tracker (Song et al. 2014)			Wu and Juang (2012)
	1	2	3	1	2	3	
DCV/TTC	1270/1273	217/219	555/562	999/1273	181/219	418/562	471/491
Accuracy ratio (%)	99.8	99.3	98.8	78.5	82.6	74.3	95.5
Error tolerance (km/h)	± 2			± 2			± 5

6 Conclusion

Pervious approaches for speed estimation generally require feature extraction, which is a difficult work for the moving object in dynamic environment. In this paper, we abandon the idea of feature tracking for speed estimation but adopt a new analysis strategy without feature extraction instead. Actually, this proposed method is the inverse process of the traditional method for vehicle speed estimation. We consider the feature differences among the images with different speeds and construct the mapping relation with the extreme value corresponding to the vehicle speed. Besides, we make use of the shadow beneath the vehicle as the tracking point, which is considered as the ground truth speed. In our experiment, we have shown that the proposed system is compared with our previous version, our previous work which uses point feature for vehicle tracking, and an approach tracking the vehicle as a whole, our system performs better than these methods. The estimated speed has an average error of 0.3 km/h, with 99.4% of the estimation speed within (− 2 km/h, 2 km/h) error range.

In the future work, we intend to optimize the location of tracking point and study how to select the tracking point automatically. We also aim to evaluate this system in a variety of scenarios, including nighttime, rainy and cloudy scenarios, and comparisons with other approaches. Another topic for future work is to construct a more diverse and comprehensive data set for speed estimation, with the ground truth speed by a higher precision speed detector.

Funding This study was funded by National Natural Science Foundation of China (Grant No. 61572083), funded by Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 18JK0617). This study was also funded by the Doctoral Scientific Research Foundation of Xi'an Shiyou University (0106-134010003).

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Banz C, Blume H, Pirsch P (2011) Real-time semi-global matching disparity estimation on the GPU. In: Proceeding of the IEEE ICCV workshops, November, pp 514–521

Cinzburg C, Raphael A, Weinshall D (2015) A cheapest system for vehicle speed detection. Preprint. arXiv:1501.06751

Dailey D, Cathey F, Pumrin S (2000) An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Trans Intell Transp Syst 1(2):98–107

Dogan S, Temiz MS, Kulur S (2010) Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. Sensors 10(5):4805–4824

Grammatikopoulos L, Karras G, Petsa E (2005) Automatic estimation of vehicle speed from uncalibrated video sequences. In: Proceeding of the modern technologies, education and professional practice in geodesy and related fields, pp 332–338

Homm F, Kaempchen N, Ota J, Burschka D (2010) Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection. In: Proceeding of the IEEE IV symposium, June, pp 1006–1013

Jeyabharathi D, Dejey D (2016) Vehicle tracking and speed measurement system (VTSM) based on novel feature descriptor: diagonal hexadecimal pattern (DHP). J Vis Commun Image Represent 40:816–830

Lan J, Li J, Hu G et al (2014) Vehicle speed measurement based on gray constraint optical flow algorithm. Optik 125:289–295

Lin HY, Li KJ et al (2008) Vehicle speed detection from a single motion blurred image. Image Vis Comput 26(10):1327–1337

Llorca DF, Salinas C, Jimenez M et al (2016) Two-camera based accurate vehicle speed measurement using average speed at a fixed point. In: Proceeding of the IEEE ITSC, pp 2533–2538

Lu S, Song H, Xu X (2015) An enumeration method applied in intelligent transportation system. Int J Smart Home 9(2):143–150

Luvizon DC, Nassu BT, Minetto R (2016) A video-based system for vehicle speed measurement in urban roadways. IEEE Trans Intell Transp Syst 18:1393–1404

Madasu V, Hanmandlu M (2010) Estimation of vehicle speed by motion tracking on image sequences. In: Proceeding of the IEEE intelligent vehicles symposium, pp 185–190

Maduro C, Batista K, Peixoto P et al (2008) Estimation of vehicle velocity and traffic intensity using rectified images. In: Proceeding of the IEEE ICIP, pp 777–780

Palaio H, Maduro C, Batista K et al (2009) Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking. In: Proceeding of the IEEE ICRA, pp 283–286

Song H, Lu S, Ma X et al (2014) Vehicle behavior analysis using target motion trajectories. IEEE Trans Veh Technol 63(8):3580–3591

Wu BF, Juang JH (2012) Adaptive vehicle detector approach for complex environments. IEEE Trans Intell Transp Syst 13(2):817–827

Zhiwei H, Yuanyuan L, Xueyi Y (2007) Models of vehicle speeds measurement with a single camera. In: Proceeding of the international conference on computational intelligence security workshops, pp 283–286

Zhiwen H, Yuanyuan L, Xueyi Y (2007) Models of vehicle speeds measurement with a single camera. In: Proceeding of the international conference on computational intelligence security workshops, pp 283–286