Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Viewpoint Independent Vehicle Speed Estimation from Uncalibrated Traffic Surveillance Cameras

Haili Mao, Chengxi Ye, Mingli Song, Jiajun Bu and Na Li
College of Computer Science
Zhejiang University
Hangzhou, China
{hailimao,yechengxi}@gmail.com {brooksong,bjj,nli}@zju.edu.cn

***Abstract*—We present here a prototype of an algorithm for vehicle speed estimation. Different from previous approaches, our algorithm requires no road markers and fewer manual calibrations. Based on specific projection rules, we find a relation between the in-camera coordinate and the real world coordinate. A non-linear regression is employed to estimate the model parameters. This model enables us to estimate the real world position of the vehicles directly from a video sequence taken by a surveillance camera. The algorithm shows its ability to produce accurate estimations in our experiments.**

***Index Terms*—Vehicle speed estimate, projection model, tracking, camera calibration, surveillance camera, Kalman filter**

I. INTRODUCTION

Nowadays, most of the traffic speed sensors deployed are magnetic based appliances or radar systems, infrared sensors [7]–[9]. However, it might be impractical or too expensive to install them at all locations. However, it is commonly noticed that a large number of CCTV cameras are already deployed for traffic surveillance and management. This common phenomenon motivates us to investigate the possibility of using computer vision techniques to build more easy-to-set (and use) traffic surveillance systems. And the idea of this paper is to make full use of these videos from these un-calibrated cameras and make them become additional speed sensors which can increase the capability of traffic management system at low costs.

In recent years, several image-based speed estimation methods have been proposed. Generally speaking, most of the methods try to calibrate the cameras and build a homograph matrix which describes the mapping from the in-camera coordinate to the real world coordinate. Vehicle speed is then estimated from the real world coordinates.

Many researchers have created systems that use information available in the image. Jung & Ho [1] placed four known poles in the surveillance scene. Zhu [11] sent a known length, width and height vehicle to drive through the scene. The method in [3] requires painted markers in the image with the required geometry. However, these methods are not suitable for calibrating multiple cameras that may move daily.

Schoepflin & Dailey [5] proposed an unsupervised calibration method based on a known distance perpendicular to the road and activity map generated from traffic flow on a plane. However, the model is a little bit complex to use, because there are too many unknown parameters. Later, they [6] use the distances between lane markers to calibrate cameras and a cross correlation technique to estimate the mean speed. But lane marker distances varies on roads with different speed limits and in some small lane or at the road cross, even no lane marker can be identified from surveillance images.

In this paper, we present a novel algorithm to calibrate traffic management cameras and use the calibration results to process video sequences for vehicle speed estimation. Different from previous approaches, our algorithm can produce results even with no road markers; fewer or no manual calibrations are required. Our algorithm requires simply video sequences from surveillance cameras. Based on specific projection rules, we find a mapping from the in-camera coordinate to the real world coordinate. A non-linear regression is employed to estimate the parameters in the mapping. This model frees the users from manual calibration if the camera is set at certain positions and directions. The algorithm then enables the user to estimate the real world position of the vehicles directly from a video sequence taken by a surveillance camera. The algorithm shows its ability to produce accurate estimation of vehicle speed in experiments.

II. SYSTEM OVERVIEW

Our vehicle speed estimation system is mainly consisted of seven modules as shown in Figure 1:

1) A "Foreground and Background Detection" module to perform FG/BG classification of each image pixel.

2) An "Extract Top and Bottom positions from FG" module that uses the result (FG mask) of the "FG/BG Detection" module to extract top and bottom 2-D vehicle position coordinates.

3) An "Estimate Camera Parameters and Vanish Point" module to collect a sufficient amount of vehicle coordinates and use them to estimate vanish point and camera parameters.

4) A "New Vehicle Detection" module to calculate each connected components of the foreground mask, each component is considered as a blob. We track each blob in the current and the previous frames. If the blob can be successfully tracked across multiple successive frames, then add this new blob into the tracked blob list.

5) A "Vehicle Tracking" module is initialized by the "New Vehicle Detection" module. This module tracks each vehicle from the tracked list.

6) A "Vehicle Position Data in Ground Plane" module that translates vehicle coordinates in image plane to ground plane.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

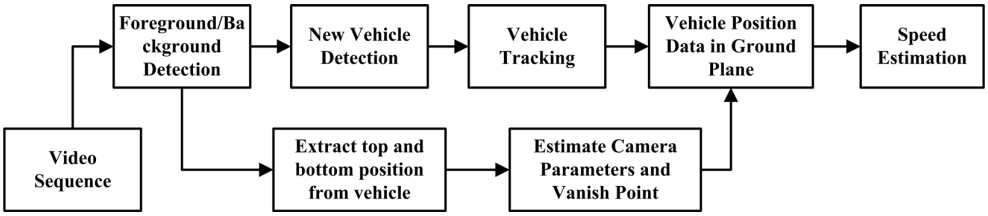

Fig. 1. Speed estimation system pipeline

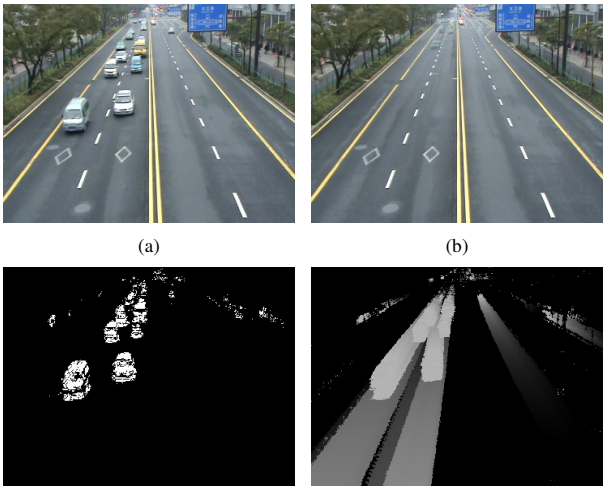

Fig. 2. Dynamic matrix. (a).Input video (b).Background (c).Subtraction image (d).Dynamic matrix

7) A "Speed Estimation" module that use vehicle ground plane coordinates as the input to estimate vehicle speed with Kalman filter.

We will discuss these modules in later sessions.

III. FOREGROUND AND BACKGROUND DETECTION

For a given video sequence, the system will firstly generate the background dynamically, according to the environment and lighting conditions.

We use the method proposed in [10] to generate the dynamic background. The basic idea of this algorithm comes from an assumption that the pixel value in the moving object's position changes faster than those in the real background. It produces a dynamic matrix to analyze the change detection result by the frame-to-frame difference method, where the motion state of each pixel is stored in the matrix. Only those pixels whose values do not change much in a period of time can be updated into the background.

In our system, we set time length $\lambda = 180$ to record the pixel's moving state, which means the pixel doesn't change much within 180 frames will be updated into the background. The update weight of input frame is 0.03.

Figure 2 shows an example of the dynamic matrix. Figure 2 (a) is the current input video. Figure 2(b) and (c) show the background and subtraction image respectively. Figure 2(d) displays the dynamic matrix which shows the motion of each pixel over the past frames.

We can obtain an initial background within a few seconds if the traffic flow is low. The situation is common especially if the traffic is controlled by traffic lights. Otherwise the convergence might be slower. The algorithm is much faster for our real-time use compared with mixture Gaussian distribution model [2]. Although the latter method can achieve similar results, it needs to consider the last N values taken by the pixel, and find the K Gaussian and weights that fit these samples best. The GMM algorithm is too computationally intensive for real-time use.

Once we have the background model, the foreground objects can be obtained with a subtraction. A problematic issue we have to face here is the shadow. We address the problem here by converting the sequence from RGB color space to HSV space; shadows are detected in the latter space. Our idea is based on the observation that since shadows are formed by occlusions and occlusions in most cases only change the luminance while keep the hue channel unchanged in some degree.

The foreground mask can then be obtained, however, with errors due to noise and algorithm accuracy. We address the issue with morphological operations like dilation and erosion, to further eliminate the inaccuracy in areas near the windows and the contours. The operations can fill in the holes in the masks or even join undesirable separate parts. Later, a threshold for the size of the foreground masks is set to eliminate the wrong masks or insignificant ones. We continue the following work with modified masks

IV. CAMERA CALIBRATION

Our model assumes that all vehicles lie in a ground plane. Based on the projection model, the length and position of a car in the image plane will follow some projection rules. We estimate the parameters in this model by a nonlinear regression method.

A. Project Model

For convenience we borrow the model from [6] and is briefly introduced as follows.

The configuration of the system is set as in Figure 3. We set up two real world coordinate systems in the model. The first

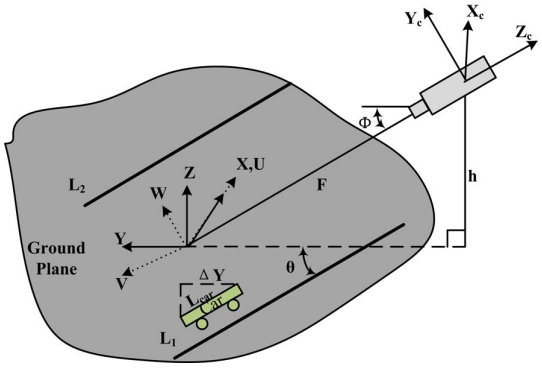

Fig. 3. Camera and roadway geometry

Fig. 4. The result of the nonlinear regression, the vanish point is estimated to lie on the marked line

is the camera coordinate system, which is fixed at the camera position. The three axis point to the right, up and back direction of the camera respectively. The second is the ground plane coordinate system and is fixed where the Z_c axis intercepts the ground plane. The three axis of this coordinate system are parallel to the X_c, parallel the projection of Z_c on the ground plane and perpendicular to the ground plane respectively. Thus, given the tilt angle of the camera and the angle θ, we can change from one system to another system with a simple affine transform. This transform can be expressed as follows:

$$\begin{aligned} X_c &= X \\ Y_c &= Y\sin(\phi) \\ Z_c &= -Y\cos(\phi) - F \end{aligned} \tag{1}$$

The third coordinate system is called the image coordinate system. Under the pinhole camera assumption, its relation with the camera coordinate system can be denoted as follows:

$$\begin{aligned} u &= -f\frac{X_c}{Z_c} \\ v &= -f\frac{Y_c}{Z_c} \end{aligned} \tag{2}$$

Furthermore, our goal is to find its correspondence with the ground plane coordinate system. Substitute (1) into (2) yields:

$$\begin{aligned} u &= f\frac{X}{Y\cos(\phi)+F} \\ v &= f\frac{Y\sin(\phi)}{Y\cos(\phi)+F} \end{aligned} \tag{3}$$

In particular, the vanish point in the image coordinate system can be obtained analytically as:

$$\begin{aligned} u_0 &= \lim_{Y\to\infty}\frac{fX}{Y\cos(\phi)+F} \\ &= \lim_{Y\to\infty}\frac{f\frac{X}{Y}}{\cos(\phi)+\frac{F}{Y}} \\ &= f\cot(\theta)\sec(\phi) \end{aligned} \tag{4}$$

$$\begin{aligned} v_0 &= \lim_{Y\to\infty}\frac{fY\sin(\phi)}{Y\cos(\phi)+F} \\ &= \lim_{Y\to\infty}\frac{f\sin(\phi)}{\cos(\phi)+\frac{F}{Y}} \\ &= f\tan(\phi) \end{aligned} \tag{5}$$

Along with (3), we can now obtain a simpler expression for X and Y as follows:

$$\begin{aligned} X &= \frac{F\tan(\phi)u}{v_0 - v} \\ Y &= \frac{F\sec(\phi)v}{v_0 - v} \end{aligned} \tag{6}$$

From the second equation of (5), we obtain:

$$\frac{Y}{F\sec(\phi)} = \frac{v}{v_0 - v} \tag{7}$$

B. Parameter estimation

Now if we follow the prior knowledge that the vehicle length is fixed in all the sequence. That means for every vehicle at time t, if we denote the front and rear of the vehicle in the image coordinate as v_{ft} and v_{rt} , then:

$$\frac{\Delta Y}{F\sec(\phi)} = \frac{v_{ft}}{v_0 - v_{ft}} - \frac{v_{rt}}{v_0 - v_{rt}} \tag{8}$$

Note that the left side of equation (7) is a constant. By moving this term to the right, we reach the equation:

$$\frac{v_{ft}}{v_0 - v_{ft}} - \frac{v_{rt}}{v_0 - v_{rt}} - Const = 0 \tag{9}$$

In our analysis, we obtain the $\{v_{ft}, v_{rt}\}$'s with a back ground subtraction algorithm. And then a nonlinear regression technique is used to estimate the model parameters. Note that nonlinear regressions may not reach reasonable solutions if the model or equation is not formulated in specific forms. In our experiment, we find it's best to incorporate the data in the lower half of the video to calculate the parameters. The coordinate of the vanish point and the parameter Const are then accurately obtained. As an example, the vanish point of the scene in

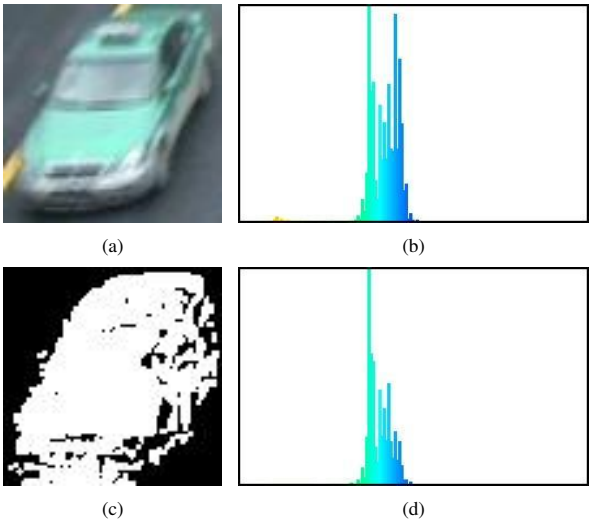

Fig. 5. (a).Target image (b).Hue histogram from target image (c).FG mask (d).Hue histogram with FG mask

Figure 4 is estimated and marked. Here, the Levenberg-Marquardt algorithm [4] is employed in the non-linear solver, which can be found in Figure 4. Note that, $Const = \frac{\Delta Y}{F\sec(\phi)}$, in our system, we assume ΔY can be known a prior, for example, if $\theta = 0$, ΔY is simply the average car length. Otherwise, $\Delta Y = Aver\_car\_length * \cos\theta$. Finally, $F\sec(\phi)$ can be obtained and this is substitute in (5) in the next steps.

V. Vehicle Tracking

We use the Mean-Shift algorithm to track every blob in our tracked blob list. A basic implementation of the Mean-Shift would contain the following 3 steps:

1) Calculation of an initial histogram which identifies the object being tracked;

2) Applying the initial histogram onto every new frame from the input stream using a technique called back projection, yielding a single channel (grayscale) image where each pixel contains the bin size of the initial histogram for the color of the corresponding pixel in the new frame (this is sometimes referred to as the "probability" of the pixel to be part of the tracked object represented by the initial histogram);

3) Searching the back-projection to find the region with the highest intensity which corresponds to the area where the tracked object most probably resides.

Since the color of a single car is mostly the same, the Mean-Shift algorithm is a sound alternative for tracking. Traditional Mean-Shift algorithm uses the hue histogram in a box as the feature (where histogram is calculated from a rectangle which bound the entire foreground mask). However, this implementation contains some hue information from background. We initialized the Mean-Shift algorithm on the foreground masks, which we've already obtained in previous steps. Note that this implementation improves the tracking accuracy by masking out the background.

VI. Vehicle Speed Estimation

In the previous section, we obtain the trajectory of every vehicle in the surveillance area by tracking. We can then transform every point in the trajectory to the 3-D ground plane coordinate system. We use position of the bottom edge of the car to represent the position of the car. And the real world position of the car is calculated according to the projection model, using the second equation of formula (5), and notes that $S = \frac{Y}{\cos\theta}$. The velocity is calculated as $v = \frac{\Delta S}{\Delta t}$. However, due to the algorithm accuracy, even little error in S can lead to large oscillations in the velocity. A simple solution is to set an entry line l_1 and exit line l_2 in the surveillance area and the time from entry to exit is used to calculate the speed. However, this model can only produce the average speed of the car as results, which might not suffice in some specific situations where there are abrupt changes in the velocity. To address this problem, we employ the Kalman filter to take acceleration into consideration.

The classic discrete Kalman filter assumes that we have a state model in which the next state vector $X(k+1)$ is a linear combination of the previous state estimate $X^+(k)$ plus some zero-mean Gaussian noise vector $u(k)$, where the noise vector has a covariance matrix Q .

$$X(k+1) = A(k)X^+(k) + u(k) \tag{10}$$

$A(k)$ is the transfer matrix. We denote predictions by the superscript $-$ and the same quantities after updates using measurement with the superscript $+$.

Here we summarize the state of the car by two position parameters, x and y, and two velocities, v_x and v_y . They are the elements of state vector $X(k)$, and the corresponding transfer matrix is:

$$X(k) = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

We assume that the obtained measurement $Y(k)$ that are linear combinations of the state variable $X(k)$ with some added zero-mean Gaussian measurement noise $w(k)$ which have a covariance matrix R. Here the measurement is a 2-dimensional vector $Y(k)$ given by:

$$Y(k) = MX^+(k) + w(k) \tag{12}$$

In our system, measurement matrix M and measurement vector Y is:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, Y = MX = \begin{bmatrix} x \\ y \end{bmatrix} \tag{13}$$

All that remains is to combine the above expressions into the generalized forms of the update equations. First we calculate $X^-(k+1) = A(k)X^+(k)$ on the basis of equation (9). Next, we obtain the covariance matrix $P^-(k+1)$ for $X^-(k+1)$ as follows:

$$P^-(k+1) = A(k)P^+(k)A(k)^T + Q \tag{14}$$

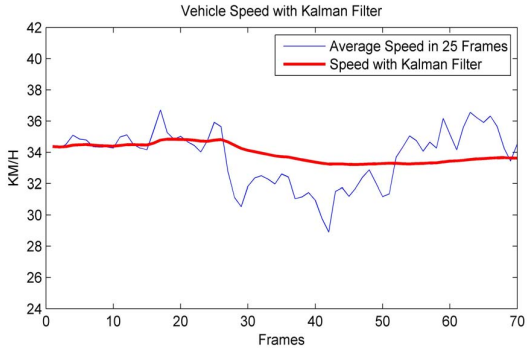

Fig. 6. Vehicle speed with Kalman filter

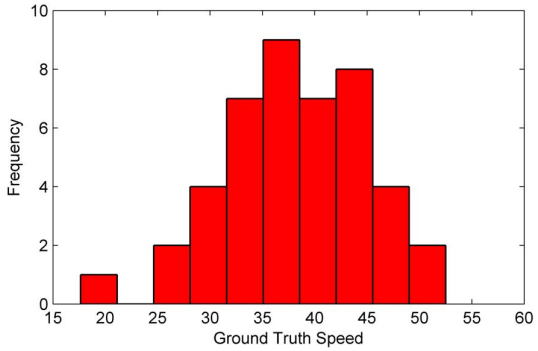

Fig. 7. Ground truth speed

We then calculate the Kalman gain K , which tells us how to optimally incorporate differences between the state prediction $X^-(k+1)$ and any state $Y(k+1)$ that we were able to measure.

$$K = P^-(k+1)M^T(MP^-(k+1)M^T + R)^{-1} \quad (15)$$

$$X^+(k+1) = X^-(k+1) + K(Y(k+1) - MX^-(k+1)) \quad (16)$$

Finally, we obtain the new estimate covariance $P^+(k+1)$ as:

$$P^+(k+1) = (I - KM)P^-(k+1)(I - KM)^T + KRK^T \quad (17)$$

The above five equations form the cycle of two-phase Kalman estimator. In each cycle, we input the car position coordinate in ground plane to the Kalman filter and get the estimated results of the car velocities. Here we show the result of speed estimation using the Kalman filter.

Here we summarize the state of the car by two position parameters, x and y, and two velocities, v_x and v_y. They are the elements of state vector , and the corresponding transfer matrix is: In Figure 6, the blue curve indicates the average speed in the last second; the red curve indicates the speed given by the Kalman filter. This shows the Kalman filter produces more stable results. In our system, the variance Q/R is set to 0.01.

VII. Result

Images from roadside cameras are used with appropriate constraints to obtain speed estimates suitable for comparison both with "ground truth". The speed estimates presented here are made using two practical constraints. The first constraint is that only vehicles whose projection is over ten pixels in length are used. The second constraint is that only vehicles identified in the lower two thirds of the image are used for speed estimation.

The ground truth speed is obtained by placing calibration lines on the road and measuring "by hand" the time for individual vehicles to pass the sequential road markings in sequential images. This ground truth individual vehicle speed can then be compared to the estimates derived from the algorithm.

Figure 7 shows the histogram of ground truth speed of 44 vehicles from both directions on both lanes in 1 minute and 30 seconds period. The video is captured in the urban area where the speed limit is 50km/h.

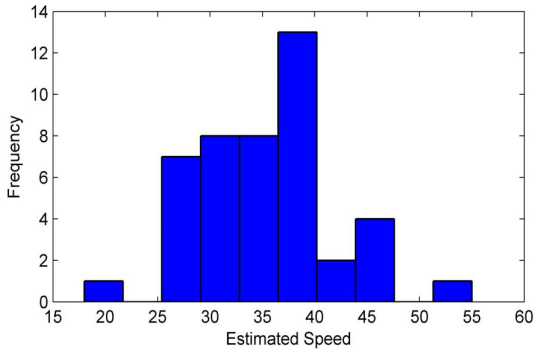

Fig. 8. Estimated speed

Figure 8 shows the histogram of the estimated speed. Note that the estimated is different from the true speed because of the change in vehicle speed and algorithm accuracy. We use the estimated speed at 2/3 of the tracking period for comparison. The Kalman filter is promised to produce stable result at this moment. What's more, the vehicles are also near enough for our proposed algorithm.

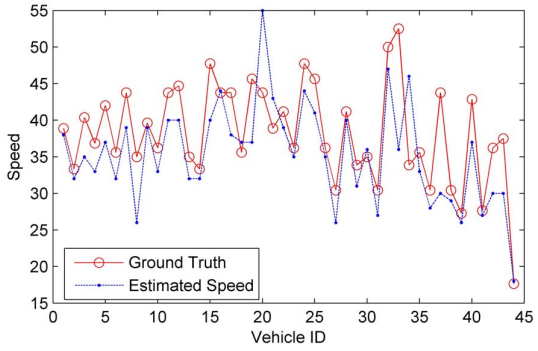

Fig. 9. Speed of each vehicle

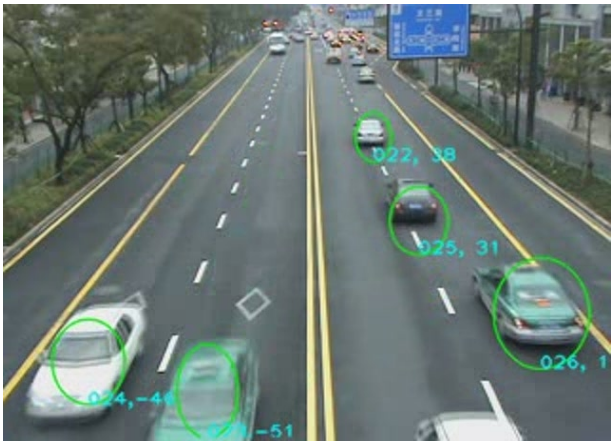

Fig. 10. Screen shot of our system

From Figure 7 and Figure 8 we can see there is only a small difference between the estimated speed and the true speed. The error mean is found to be 2.91km/h. Figure 9 shows the difference of estimated speed and the true speed for every vehicle and the errors are fewer than 10% for most vehicles.

Figure 10 shows a screen shot of our vehicle speed estimated system. Three are two numbers at the bottom of each vehicle, the first number is ID and the second number is the estimated speed. The complete video sequence is available at "http://eagle.zju.edu.cn/home/cv/mao/result.AVI".

We found the proposed algorithm can produce sound estimations for the vehicle speed while requiring fewer inputs and manual calibrations compared with previous approaches.

ACKNOWLEDGMENT

This paper is supported by National Science Foundation of China (Grant No. 60873124), Key Projects in the National Science & Technology Pillar Program (No. 2008BAH26B00) and Zhejiang Key Laboratory of Service Robot (No. 2008E10004).

REFERENCES

[1] Young-Kee Jung and Yo-Sung Ho. Traffic parameter extraction using video-based vehicle tracking. pages 764–769, 1999.

[2] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. European Workshop Advanced Video Based Surveillance Systems*, volume 1, 2001.

[3] A.H.S. Lai and N.H.C. Yung. Lane detection by orientation and length discrimination. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 30(4):539–548, Aug 2000.

[4] J.J. Mor. The Levenberg-Marquardt algorithm: implementation and theory. *Lecture notes in mathematics*, 630:105–116, 1977.

[5] T.N. Schoepflin and D.J. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):90–98, June 2003.

[6] T.N. Schoepflin and D.J. Dailey. Algorithms for calibrating roadside traffic cameras and estimating mean vehicle speed. pages 277–283, 30 2007-Oct. 3 2007.

[7] C. Sun, S.G. Ritchie, California, Institute of Transportation Studies, Partners for Advanced Transit, Highways (Calif, and Dept. of Transportation. Individual vehicle speed estimation using single loop inductive waveforms. *Journal of Transportation Engineering*, 125(6):531–538, 1999.

[8] Y. Wang and N.L. Nihan. Freeway traffic speed estimation with single-loop outputs. *Transportation Research Record: Journal of the Transportation Research Board*, 1727(-1):120–126, 2000.

[9] D.L. Woods, B.P. Cronin, R.A. Hamn, Texas, Texas Transportation Institute, Dept. of Transportation, Office of Research, and Technology Transfer. *Speed Measurement with Inductance Loop Speed Traps*. Texas Transportation Institute, Texas A&M University System; Available through the National Technical Information Service, 1994.

[10] T. Yang, S.Z. Li, Q. Pan, and J. Li. Real-time and accurate segmentation of moving objects in dynamic scene. In *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, pages 136–143. ACM New York, NY, USA, 2004.

[11] Zhigang Zhu, Bo Yang, Guangyou Xu, and Dingji Shi. A real-time vision system for automatic traffic monitoring based on 2d spatio-temporal images. pages 162–167, Dec 1996.