Vehicle 3D Estimation Based on Time Series Images and Prior Knowledge

Haoyin Zhou, Tao Zhang, Changshui Zhang, Peng He
Department of Automation, Tsinghua University, Beijing 100084, China
(*Tel: +86-10-62797710, zhouhaoyin@gmail.com*)

Abstract: Vehicle 3D estimation is important in intelligent transportation systems. To simplify system structure and improve system accuracy, an algorithm based on one-camera system and prior knowledge is presented. The experimental result shows that the algorithm can achieve satisfied accuracy.

Keywords: prior knowledge, 2D image, 3D estimation, shadow removal

I. INTRODUCTION

In intelligent transportation systems, it is usually to use images captured by video cameras to identification information of vehicles. The most usually method to get vehicle boundary box is to use 2 or more video cameras for 3D estimation. This kind of system considers vehicles as moving objects with no prior knowledge. Hence they usually have low accuracy for 3D estimation. In the method proposed in this paper, we want to take fully use of prior knowledge and estimate 3D information of vehicles based on single video camera.

We get some helpful prior knowledge, including the physical coordinates of 3 or more feature points in the image, physical coordinates of the video camera, the driving direction of vehicles, the range of length and width of a certain kind of vehicle, and the position of light.

The mapping between image coordinates and physical coordinates can be acquired from the physical coordinates of feature points. So vehicle area that is got by motion detection can be projected to physical coordinate. When the vehicle is moving along the street, the length direction of the vehicle can be confirmed. Intuitively, the length of the vehicle can be got by measuring the move area along the driving direction.

However, because of the height and shadow, the vehicle area got by motion detection is not precise. To remove the impact of height information, we can do the measurement in different frames of the same video. In this process, some height information can be got.

Shadow removal is a great problem in CV. If we get the prior information of the position of the light and the average length of this type of vehicle, it is easy to know the approximate location of the shadow. And a suitable location of the shadow segmentation can be found by calculate variance of a slide window along the length direction. It is the same to estimate the width of a vehicle as the length estimation. In the end, the result of 3D estimation is given.

II. METHOD

1. Motion Detection

Motion Detection is to detect the moving areas in each video frame from the background. When a car is running on the road and a static camera captured the entire scene saved as a piece of video, what we do is to find the moving car area in every video frame. The called "moving area" means that there are one or more moving objects in these labeled moving areas. Effective and efficient motion detection is the basement of tracking of moving objects [1].

Gaussian Background Model has been widely used for robustly modeling complicated backgrounds, especially those with small repetitive movements (such as leaves, bushes, rotating fan, ocean waves, rain). According to Gaussian Background Model, the distribution of each pixel's lightness of a background image meets Gaussian distribution, which means for an image B [2]:

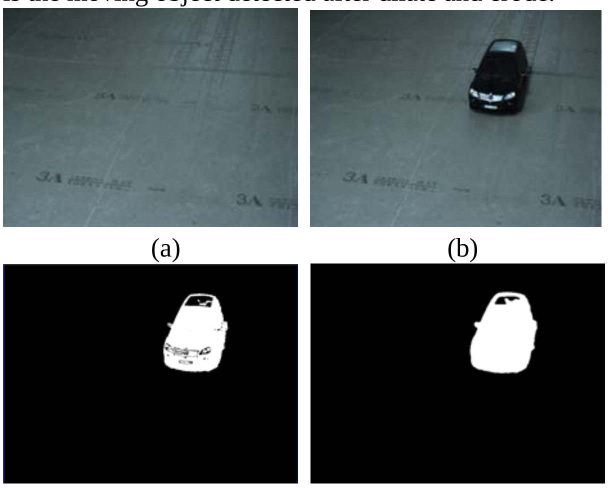
$$I_B(x,y) \sim N(u,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\sigma > 0) \qquad (1)$$

Thus, each pixel has two parameters which are average u and variance σ^2. Meanwhile, the background image is changing slowly with time. The parameters of each pixel in the background image should be updated:

$$u(t+1,x,y) = \alpha \cdot u(t,x,y) + (1-\alpha) \cdot I_{t+1}(x,y) \quad (2)$$

Where $u(t,x,y)$ is point (x,y)'s average in the time t, $I_{t+1}(x,y)$ is the pixel of point (x,y) in $t+1$ frame, and α is the updating speed. The changing of σ^2 is so little to be ignored [3].

According to the model above, we use the first frame without moving to build the background model and update the model with the latest input frame. The moving areas can be got by the current frame subtracting background model. The pixels with high difference belong to moving areas. Then we filtrate the noise in the image by morphologic processing and Gaussian filter. For visual effect, we label background pixel black and object pixel white, producing a binary image. The result shows that the Gaussian background model achieves better performance. Fig. 1(a) is the Gaussian background model. Fig. 1(b) is some frame with a moving object. Fig. 1(c) is got by Fig. 1(b) subtracting background model. Fig. 1(d) is the moving object detected after dilate and erode.



(a) (b)

(c) (d)

Fig.1 (a) Gaussian background, (b) Frame with vehicle, (c) Subtract result, (d) Dilate & erode

2. Convert to physical coordinate

Image coordinate can be converted to physical coordinate with some necessary prior knowledge. The system geometry is shown below. Here we take the left-bottom point as the origin, and direction of vehicle as y-axis. The z-axis is perpendicular to the ground.

At first, we measure the physical coordinate of the following points: 2 edge points of the view field, which are named (x1, y1, 0) and (x2, y2, 0); point which is in the center of the image (x4, y4, 0); camera optical center points (x0, y0, z0).

If the focus length of the camera is f, the camera imaging plane equation is:

$$\begin{aligned}&(x0-x4)(x-x0-\frac{f(x4-x0)}{\sqrt{x4-x0)^2+(y4-x0)^2+(h)^2}})\\&+(y0-y4)(y-y0-\frac{f(y4-y0)}{\sqrt{x4-x0)^2+(y4-x0)^2+(h)^2}})\\&+(h)(z-h+\frac{fh}{\sqrt{x4-x0)^2+(y4-x0)^2+(h)^2}})=0\end{aligned} \tag{3}$$

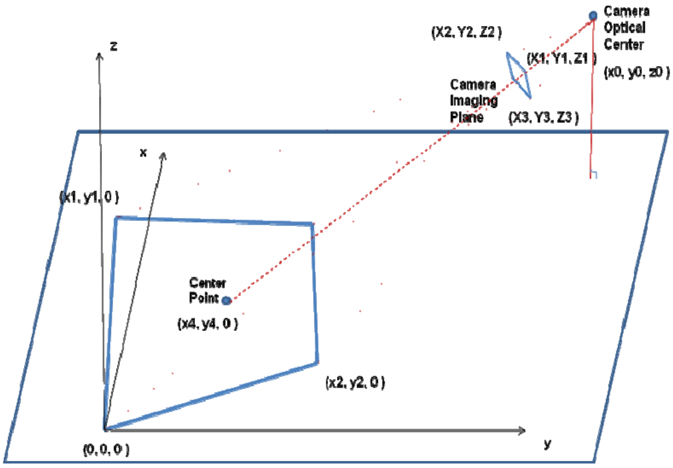


Fig 2 Schematic diagram of the system geometry

The 3 straight line equations that separately get pass the 3 edge points of the view field and camera optical center point are similar, take the line that get pass (x1, y1, 0) and (x0, y0, z0) for example:

$$\begin{cases}x=(x1-x0)t+x0\\y=(y1-y0)t+y0\\z=(z1-z0)t+z0\end{cases} \tag{4}$$

With the plane equation and 3 straight line equation, the physical coordinate of the camera imaging plane edge point (X1, Y1, Z1), (X2, Y2, Z2), (X3, Y3, Z3) are got.

If pixel size of the image is 780*560, the physical coordinate of (i, 0) and (0, j), which is image coordinate, in the camera imaging plane is:

$$\begin{aligned}U1&=\frac{i}{780}(X2-X1)+X1\\V1&=\frac{i}{780}(Y2-Y1)+Y1\\W1&=\frac{i}{780}(Z2-Z1)+Z1\end{aligned} \tag{5}$$

$$\begin{aligned}U2&=\frac{j}{560}(X2-X1)+X1\\V2&=\frac{j}{560}(Y2-Y1)+Y1\\W2&=\frac{j}{560}(Z2-Z1)+Z1\end{aligned} \tag{6}$$

Physical coordinate and image coordinate are both describing the position of the same point. When (i, j) = (i, 0) + (0, j) in image coordinate, the physical coordinate of (i, j) is

$$\begin{aligned} U &= U1 + U2 \\ V &= V1 + V2 \\ W &= W1 + W2 \end{aligned} \tag{7}$$

The equation of the straight line that gets pass the camera optical center and (U, V, W) is:

$$\begin{cases} x = (U - x0)t + x0 \\ y = (Y - y0)t + y0 \\ z = (W - z0)t + z0 \end{cases} \tag{8}$$

The intersection of this line and z=0 plane is:

$$\begin{aligned} x &= (U - x0)\frac{-z0}{W - z0} + x0 \\ y &= (v - y0)\frac{-z0}{W - z0} + y0 \\ z &= 0 \end{aligned} \tag{9}$$

Here we find the mapping between image pixels to real world point. So the virtual grid can be drawn in images. Every grid is 10cm*10cm in physical world; the sides of grids are separately along the vehicle length and width direction.

3. Shadow Cutting

We use slide window and histogram for shadow cutting. At first, convert the image from RGB space to HSV space, which is more suitable for dividing shadow and vehicle body. Secondly, we create narrow and long window along length direction. And calculate histogram of H component of HSV. Thirdly, compare the difference of histogram of adjacent window, as shown in figure 4, and find some local maximum line. The algorithm of comparison is as follows [4]:

$$\begin{aligned} d(H_1, H_2) &= \frac{\sum_I (H_1'(I) \cdot H_2'(I))}{\sqrt{\sum_I (H_1'(I)^2) \cdot \sum_I (H_2'(I)^2)}} \\ H_k'(I) &= \frac{H_k(I) - 1}{N \cdot \sum_j H_k(J)} \end{aligned} \tag{10}$$

With time-series images, we can find local maximum lines in every image. We can find some stable lines after converted to physical coordinate and fitting,

Here the prior knowledge of statistical average vehicle width can be used, and choose the nearest line as shadow cutting line, Fig. 8 shows the result of shadow cutting.

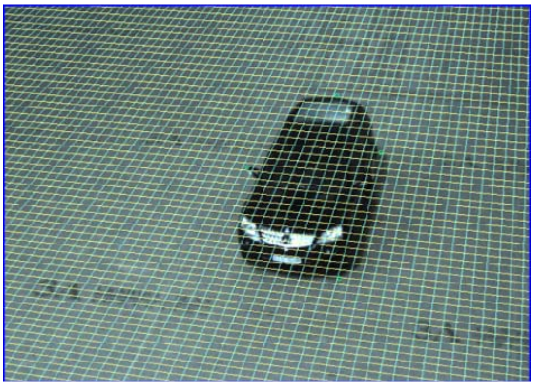


Fig 3 Virtual grid

Fig 4 Histogram calculate area

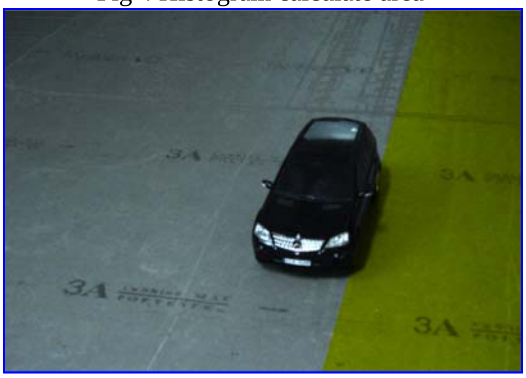
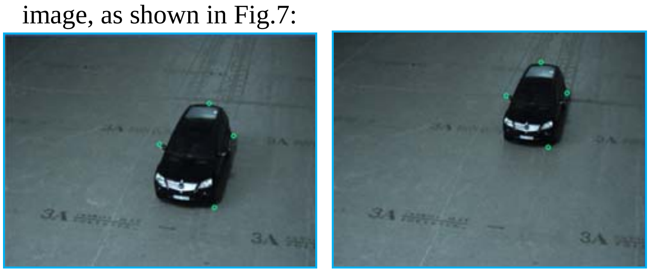


Fig 5 Shadow cutting line

4. Get Real Size

After shadow cutting, we can get the 4 edge point of the vehicle, from which the look length and look width can be got.

The look length of the vehicle can be got in every image, as shown in Fig.7:



(a) (b)

Fig.6 (a)Edge points, (b)Edge points

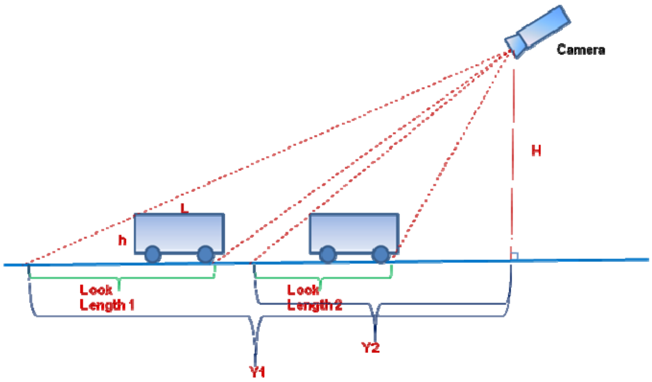


Fig 7 System geometry

With time-series images, a lot of look length can be got by measurement

$$\begin{cases} L + \dfrac{Y1}{H} h = L1 \\ L + \dfrac{Y2}{H} h = L2 \\ L + \dfrac{Y3}{H} h = L3 \\ \ldots\ldots \end{cases} \quad (11)$$

Define $A = \begin{bmatrix} 1 & Y1/H \\ 1 & Y2/H \\ 1 & Y3/H \\ \ldots & \end{bmatrix}, X = \begin{bmatrix} L \\ h \end{bmatrix}, b = \begin{bmatrix} L1 \\ L2 \\ L3 \\ \ldots \end{bmatrix}$

To get a least squares solution of the over determined equations, we can calculate the solution of equation:

$$A^T A x = A^T b \quad (12)$$

Because of $Yi \neq Yj (i \neq j)$, the rank of matrix A is 2. So we can get

$$rank(A^T A) = rank(A) = 2 \quad (13)$$

Here is the lease square solution:

$$x = (A^T A)^{-1} A^T b \quad (14)$$

III. RESULT

We test our algorithm in 5 different kinds of different vehicles, of which the result is shown in table 1, 2 and 3.

Table 1. 3D estimation result：length

Vehicle type	Real Length	Estimate Length	Length Error
Car	48.00	50.31	-2.31
SUV	46.00	45.35	0.65
Mini-Car	36.00	34.47	1.53
Truck	60.00	58.23	1.77
Bus	72.00	69.78	2.22

Table 2. 3D estimation result：width

Vehicle type	Real Width	Estimate Width	Width Error
Car	20.00	22.03	-2.03
SUV	21.00	21.40	-0.4
Mini-Car	19.00	20.19	-1.19
Truck	35.00	35.49	-0.49
Bus	24.50	25.53	-1.03

Table 3. 3D estimation result：hight

Vehicle type	Real Height	Estimate Height	Height Error
Car	14.00	9.80	4.2
SUV	17.50	16.46	1.04
Mini-Car	14.00	9.64	4.36
Truck	33.00	31.06	1.94
Bus	28.00	26.78	1.22

In table 1, 2 and 3, most estimate error is less than 2 cm, except the height of car and mini-car. That is because the real top point does not influence the look length and look width of the vehicle in images.

IV. CONCLUSION

Vehicle information extraction is being widely studied. This paper comes up with an algorithm for vehicle 3D estimation with prior knowledge. With time series images, 3D information can be extracted from the 2d information in every image.

The shortcoming of the algorithm is that it cannot accurate estimate vehicle information when it changes moving direction.

REFERENCES

[1] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in Proc. IEEE Workshop Applications of Computer Vision, 1998, pp. 8–14.

[2] C. Stauffer andW. Grimson, "Adaptive background mixture models for real-time tracking," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, 1999, pp. 246–252.

[3] H. Z. Sun, T. Feng, and T. N. Tan, "Robust extraction of moving objects from image sequences," in Proc. Asian Conf. Computer Vision, Taiwan, R.O.C., 2000, pp. 961–964.

[4] M. Kiger, "A shadow handler in a video-based real-time traffic monitoring system," in Proc. IEEE Workshop Applications of Computer Vision, CA, 1992, pp.11-18.