Image and Vision Computing Journal (to appear)

VISATRAM: A real-time vision system for automatic traffic monitoring

Zhigang Zhu*, Guangyou Xu, Bo Yang, Dingji Shi, Xueyin Lin

Department of Computer Science and Technology

Tsinghua University, Beijing 100084, China

Abstract

This paper presents a novel approach to automatic traffic monitoring using 2D spatio-temporal images. A TV camera is mounted above a highway to monitor the traffic through two slice windows, and a panoramic view image and an epipolar plane image are formed for each lane. Our real-time vision system for automatic traffic monitoring, VISATRAM, is an inexpensive system with a PC486 and a frame grabber. The system can not only count vehicles and estimate their speeds, but also classify them using 3D measurements. The system has been tested with real road images under various light conditions, including shadows in daytime and lights at night.

Keywords: Intelligent vehicle/highway system, traffic monitoring, spatio-temporal image, epipolar plane image, panoramic view image

---

* The author is currently on a leave in the Computer Science Department, University of Massachusetts at Amherst, MA 01003. Email: zhu@cs.umass.edu, or zhuzhg@mail.tsinghua.edu.cn. This work was supported by China Advanced Research Project during 1993-1997. An earlier version of this paper is presented in the 1996 IEEE Workshop on Application of Computer Vision [15].

1. Introduction

Automatic traffic monitoring plays an important role in an Intelligent Vehicle/Highway System (IVHS). A vision-based approach is promising since it requires no pavement reconstruction and has more potential advantages, such as larger detection areas, and more flexibility than inductive loops. However, traffic flow raises interesting but difficult problems for image processing. Various light conditions create a need for robust algorithms, which require a large amount of computational power to meet the real-time operations of a traffic monitoring system. Many research efforts have been made in this area, and there exist several commercial products (e.g., [1,2]); but there are room for significant improvement in performance and capability of visual traffic monitoring systems.

A successful and widely used vision system for real traffic monitoring applications must meet the following four basic requirements.

- Easy installation and calibration. This is important for on-site setup, reconfiguration and operation by non-expert personnel.
- Environmental adaptation. The real system should work in different light conditions, including heavy shadows under strong sunlight, dim illumination in the evening, vehicle headlights at night and abrupt light changes. The basic problem is the background updating and vehicle separation.
- Accurate vehicle speed and size estimation is needed for applications such as intersection control, traffic surveillance, speed trap detection, vehicle classification and other special studies.
- Real-time operation and low cost. These are a key factor for the wide use of an on-line traffic monitoring system.

1.1. Related works

D'Agostino [3] discussed the potentials of a commercial machine vision system for traffic monitoring and control. The basic requirements are low cost and robust performance, which have not been fully met till now. For the system described in [3], problems of shadows and nighttime operation had not been solved. For example, the system cannot classify vehicles during

nighttime. Vehicle speed was estimated by setting up two inspection zones. The system needs expensive special image processing hardware.

A background updating method for road traffic scenes is discussed in [4]. This system extracts vehicles using a subtraction-spatio differentiation method to remove the adverse effects of vehicle shadows. The background updating method is based on the change ratio between the road surface brightness of the current input image and that of the old background image. Since every full frame of a video sequence must be processed, a specially designed HITACHI-IP/200 parallel image processor was used for real-time detection of a three-lane flow of vehicles. In addition, the system must determine if vehicle(s) appearing in two successive frames are the same. The problem of estimating vehicle speed was not addressed.

In other systems, vehicle shapes are modeled using complex models [5-8], which cannot be processed in real-time with low-cost hardware. Sullivan [5] used a wire-frame model for the vehicle. Based on an estimated pose and full calibration parameters of the camera, the model is back-projected to the image and edges of the model are then matched to lines in the image. The system requires full calibration of both intrinsic and extrinsic parameters, which is tedious and sensitive to noise.

Yuan et al [6] extract a vehicle and estimate its length, width, height and the number of units of the vehicle from a single perspective image captured by a camera placed at the roadside. The ultimate goal of their system is to classify vehicles into many categories, therefore reducing the gap between the requirement and the availability. However their approach encounters the general problems in image segmentation, and the methods to identify the roof, side and front of a vehicle are quite ad hoc. In a more recent article [7], vehicles and their wheels are extracted by using a more sophisticated image segmentation method and a deformable model of the vehicle. All of these approaches may achieve more profound goals, but real-time implementation with low-cost hardware is still difficult. Moreover, with all these methods, the entire vehicle is assumed to be fully visible in a single image, which is not always true.

Ferrier et al [8] obtained real-time performance by tracking the occluding contour using intensity/motion information. Initial calibration of a projection relationship between an image and the ground plane enable metric information to be derived from the image positions and velocities without full calibration. A tracking technique used in their paper is designed to be

resilient against the vibration of the camera. Real-time performance was achieved on a SUN IPX with a Datacell S2200 image capture board. An underlying assumption is that the tracked outline of the vehicle is roughly plane shape. This weak perspective viewing condition can only be satisfied if the camera is far enough above the road being viewed. In order to calibrate the camera, markers should be placed and measured for locations lacking any existing road markers. Effects of headlights at night were not discussed in the paper.

Kilger [9] at Simens AG showed that real-time traffic monitoring is possible with low-cost hardware. A bounding box, especially the width of this box was used as the geometric model for robust classification of vehicles in a real-time application under the difficult illumination conditions typically found on a sunny day with heavy shadows. Shadows are separated from vehicles by investigating an edge image of detected regions. The speed of a vehicle is estimated by tracking the middle point of the vehicle's front edge in the image sequence, where a constant speed assumption is made. However speed estimation results were not mentioned. The box model is applied directly to perspective images; no 3D measurements of vehicles were conducted.

Zielke et al [10] proposed a method for detecting and tracking cars based on symmetry. This method can be used in situations where a camera is mounted on another vehicle in the same lane. However it will not be the best viewing position for monitoring traffic. A fully symmetrical view of a vehicle within an image is possible only from some particular vantage points.

In order to extract landmarks for global localization of the mobile robot, Zheng and Tsuji [11] proposed a panoramic representation of roadside scenes. Careful study is needed on how to use this panoramic representation for traffic monitoring. In motion analysis, Baker et al [12] first introduced the concept of epipolar plane image (EPI) analysis. In the application of traffic monitoring, Nakanishi and Ishii[13] presented a method for extracting images of laterally moving vehicles from image sequences based on EPI analysis. Problems of occlusion and background updating under typical daylight variations were addressed. They detected the locus of a vehicle using Hough transform and classified the vehicle type based on silhouette analysis. Their experiments were carried out in a Sparc Station 1; however, real-time operations and nighttime operations were not mentioned.

While developing algorithms for visual traffic monitoring are needed, the evaluation of these algorithms is also an important aspect for real applications. Due to technology limitations, these algorithms have traditionally been evaluated at a macroscopic level by comparing counts obtained by loop detectors with an image-based detection system. Bullock and Mantri [14] presented a multimedia data model for investigating the microscopic performance of video detection algorithms. Because video data tends to consume huge quantities of storage, disk-space requirements are an obvious concern. A compact visual representation of traffic events for examination and evaluation needs to be developed.

1.2. Overview of our approach

In this paper we present a novel approach to automatic traffic monitoring using 2D spatio-temporal images. A TV camera is mounted above the highway to monitor the traffic through two slice windows for each traffic lane (Fig. 1). One slice window is a "detection line" perpendicular to the lane and the other is a "tracking line" along the lane. Two types of 2D spatio-temporal (ST) images are combined in our system: the panoramic view image (PVI) [11] and the epipolar plane image (EPI) [12]. The problems of vehicle counting, speed estimation and vehicle classification are solved through analyzing these two 2D ST images. An inexpensive real-time system, VISATRAM (VIsion System for Automatic TRAffic Monitoring), with a PC486 and a commercially available frame grabber has been tested with real road images.

Our approach has the following features and advantages:

(1) *Real-time and robust performance*: VISATRAM is a real-time visual system working robustly under various light conditions including shadows and vehicle lighting. It can automatically cope with both slow and sudden illumination changes. The system can automatically recover from false sensing or abrupt changes in environment.

(2) *Enhanced functions.* Our system can not only count vehicles and estimate their speed, but also classify the passing vehicles using 3D measurements (length, width and height). Moreover, robust speed and height estimation are obtained from the loci of a vehicle's front and rear edges instead of using only the vehicle's locations at two different instants as in [3,4,8]. The detection results are visually superimposed on the live video screen.

(3) *Low cost and efficient computation.* Traffic parameters are obtained by 2D spatio-temporal image techniques implemented using an inexpensive image processing system: a PC486 and a cheap frame grabber. Only a few scan lines are processed in each frame. ST images are more generic and simpler than frame-by-frame images in this special application. Narrow spatial viewing windows are compensated for by dense temporal sequences, and vehicles that are partially viewed in a single frame can be reconstructed using ST images.

(4) Easy installation and calibration. The camera system can be installed without disturbing the traffic flow. Once the hardware is installed, a detection line and tracking lines can be easily re-defined or re-positioned on a video screen to adapt for changing traffic control and/or data collection requirements. Camera parameters for the 2D ST image geometry can be easily decided without actually measuring any 3D coordinates of the road environment. Instead, camera calibration is realized using only the known size of a passing vehicle.

(5) Compact visual representation. PVIs are a compressed and panoramic representation of a traffic flow and they can be saved on hard disk for further examination and study. ST images are also suitable for performance analysis of a traffic monitoring system when the ground truth information is not available.

This paper is organized as follows: In the next section, the 2D ST geometry and an image rectification technique are described. Section 3 presents methods of acquiring the necessary vehicle metric measurements, namely, speed and 3D size, using 2D ST images. A calibration method and error analysis will also be given in Section 3. Section 4 discusses image processing techniques used for vehicle separation and locus tracking. A background updating method is included in this section. The experimental results with real-time performance and compact visual representation for traffic flow will be provided in Section 5. Section 6 is a brief conclusion.

2. Spatio-Temporal Geometry

2.1. Camera setting and ST image geometry

In the ideal system setup of VISATRAM, a camera is mounted over the center of a highway, although other camera settings are possible. The pan/zoom/tilt settings should be fixed to retain detection configurations. We assume that vehicles move away from the camera with constant

speeds along straight lanes in the camera's field of view (FOV) (Fig. 1). This setting is suitable for the detection and tracking of a vehicle, since the vehicle enters into the FOV in the high-resolution end of the image, and a tracking line can be determined according to the position of each vehicle in the lane. Moreover the locus of any point on the vehicle will be a straight line in the rectified epipolar plane image (Fig. 2, Fig.4). This setting also reduces the negative effect of a vehicle's headlight at night, since the light is not directly reflected to the camera. The system is designed to work in conditions that include heavy shadows, dim light, and nighttime conditions. Auto iris is permitted and somewhat advantageous for vehicle separation and background updating.

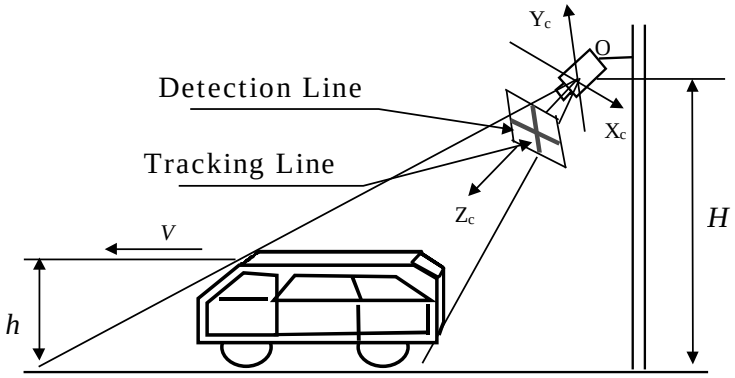


Fig. 1. The camera setting

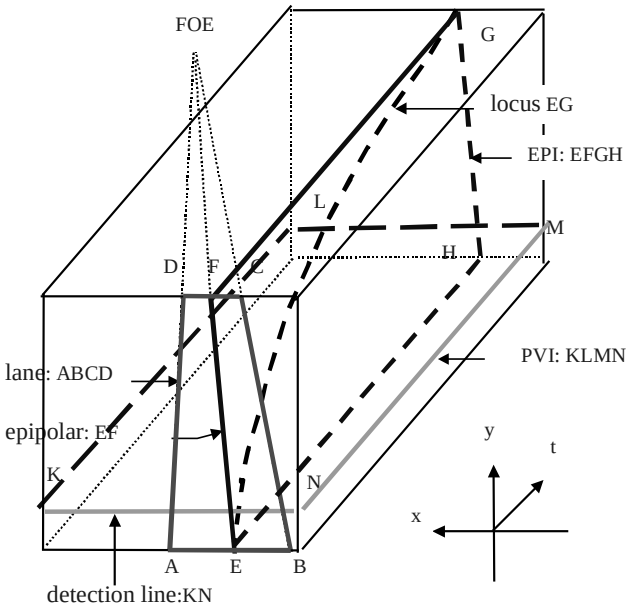


Fig. 2. ST geometry

The behaviors of vehicles moving on a road can be calculated by monitoring traffic through two slice windows (Fig. 1), a detection line and a tracking line, and by further generating a panoramic view image (PVI) and an epipolar plane image (EPI). Inside the 3D ST image cube xyt, PVI and EPI are two kinds of representative 2D intersecting planes that reveal most of the traffic flow information (Fig. 2)**.** The PVI is formed by piling up the horizontal detection lines from consecutive frames, conceptually one scanline from each frame. The EPI, on the other hand, is formed for each lane by piling up the tracking line of consecutive frames along the epipolar line. Fig. 4 shows one of the examples. The PVI shows the presence of a vehicle, and its width W and duration time T across the detection line, while the EPI tells us the speed *V*, length *L* and height h of the vehicle. The size and class of the vehicle can be easily obtained by integrating measurements from the above two sources. Based on these measurements, other traffic parameters such as volume, occupancy, headway, etc., can also be recovered easily.

2.2. ST image rectification

The camera is calibrated in order to find the relationship between the world coordinates and the image coordinates. It should be noted that our approach is quite different from the traditional methods of calibration in that no 3D measurements are needed, thus the algorithm is simple and straightforward. At first, the focus of expansion (FOE) is estimated by using the lane boundaries. For example, in Fig. 2, the FOE is the intersection of the two boundary lines AD and BC of a lane. Any image projection *p* of a 3D point on a passing vehicle will moves along the epipolar line passing through both point *p* and the FOE (Fig. 2). A "tracking line" in each frame is defined as a line segment (e.g., EF in Fig. 2) along the epipolar line, and an EPI is formed. However the locus of point *p* in the original EPI will not be a straight line if the optical axis is not perpendicular to the road surface, which is mostly the case in a practical setting (Fig. 4(2)). Therefore, we re-project the sensor image to a rectified image plane parallel to the road surface. The rectification is made only along the selected epipolar line inside a lane using the cross-ratio invariance (Fig. 3)

$$\frac{\overline{PP_1}/\overline{PP_2}}{\overline{pp_1}/\overline{pp_2}} = \frac{\overline{P_3P_1}/\overline{P_3P_2}}{\overline{p_3p_1}/\overline{p_3p_2}} = \lambda(\text{constant}) \qquad (1)$$

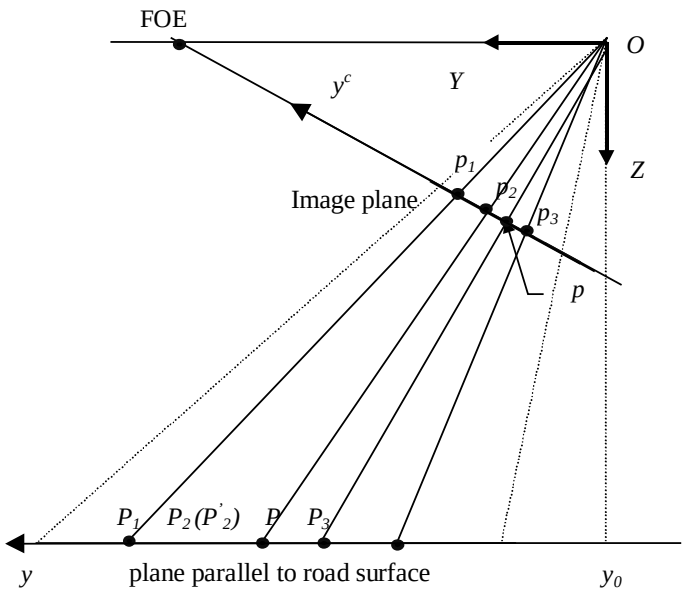


Fig. 3. Image rectification

In equation (1), P_1, P_2 and P_3 are known points on a reference plane parallel to the road surface, and p_1, p_2 and p_3 are their corresponding image points. Point p represents the image of any point P on the reference plane. $\overline{P_i P_j}$ is the signed distance between points P_i and P_j, etc.. Without measuring any absolute coordinates in the world, we use the fact that the real size of a moving vehicle is not changed. In equation (1) it means $\overline{P_3 P_2} = \overline{P_2 P_1}$, where P_3 and P_2 are the y coordinates, at the same instant, of two points on the top of the vehicle with the same height, while P_1 is the new y coordinate of point P_2 at the instant when P_3 moves to P_2 (P'_2). Their corresponding points in the EPI, p_1, p_2, p'_2 and p_3, are shown in Fig. 4(2)). In this way the original EPI with a curved locus is transformed to a rectified EPI with a straight locus, whose slope is proportional to the speed of the vehicle.

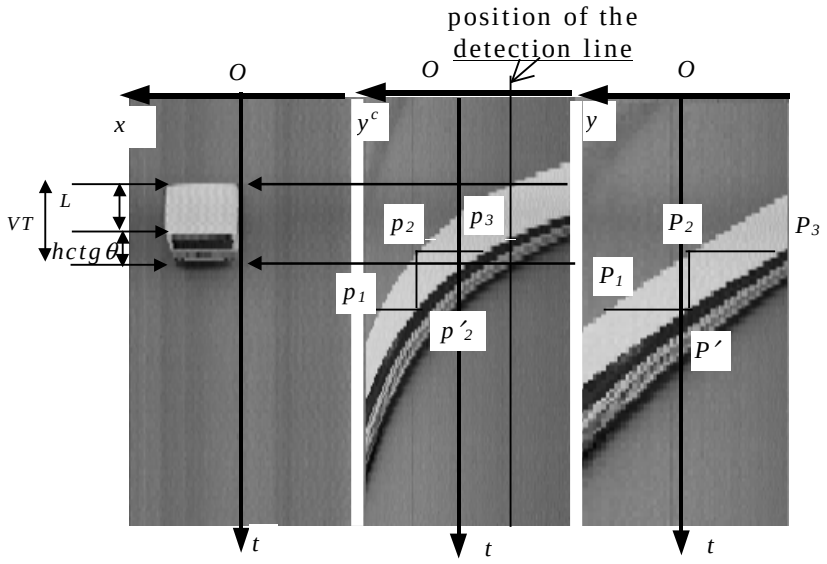
If the spatial coordinates in the original and rectified EPIs are denoted as y^c and y respectively (Fig. 3), we have

$$\lambda = \frac{(y_3 - y_1)/(y_3 - y_2)}{(y_3^c - y_1^c)/(y_3^c - y_2^c)} \qquad (2)$$

where y_1^c, y_2^c, y_3^c can be measured from the original EPI, and y_1, y_2 and y_3 can be determined by giving coordinate y_2 and the length $\overline{P_3P_2} = \overline{P_2P_1}$. For any point (y,t) in the rectified EPI , we can find its correspondence (y^c,t) by using equation

$$y^c = \frac{(y-y_1)y_2^c - \lambda(y-y_2)y_1^c}{(y-y_1)-\lambda(y-y_2)} \tag{3}$$

Fig. 4(3) shows the rectified EPI of Fig. 4(2). It should be pointed out that the point y = 0 is not necessarily the point y_0 where the optical axis of the rectified virtual camera pierces through (Fig. 3, Fig. 4).



(1)PVI (2)Original EPI (3) Rectified EPI

Fig. 4. 2D ST images

2.3. 2D ST image models

After the image rectification, the image coordinates, *(x,t)* in the PVI and *(y,t)* in the rectified EPI, of a 3D point *(X,Y,Z),* are measured in a rectified " virtual camera" and a rectified image plane under the pinhole camera model, as (Fig. 5, Fig. 6)

$$x(t) = f_x \frac{X(t)\sin\theta}{H-h} \tag{4}$$

$$y(t) = f_y \frac{Y(t)}{H-h} + y_0 \quad (5)$$

where H is the height of the camera above the road , h is the height of a point on the vehicle (i.e., $Z = H\text{-}h$), θ is the angle between the road surface and the plane passing through the optical center and the detection line, and f_x and f_y are the equivalent focal lengths for the PVI and rectified EPI respectively. The focal lengths f_x and f_y may not be the same due to the aspect ratio of the camera and the rectification of the EPI. The coordinates *x(t), y(t), X(t)* and *Y(t)* are all functions of time *t*. A calibration method is given in Section 3.2.

3. Vehicle Metric Measurements

3.1. Speed and size estimation

In order to simplify the 3D estimation, a cuboid vehicle model is assumed. When a vehicle moves away from the camera, Loci of the vehicle in the rectified EPI are bounded by the loci of the front and the rear (Fig. 4). The point on the rear that appears in the EPI is roughly considered as ground point (i.e., $Z = H$) and the point on the front is considered as a roof point of the vehicle. By differentiating equation (5) we have

$$\frac{\partial y(t)}{\partial t} = \frac{f_y}{H-h} \frac{\partial Y(t)}{\partial t}$$

Under constant speed assumption during the monitoring period, the speed of the vehicle on the road is $V = \frac{\partial Y(t)}{\partial t}$, and the slope of the locus in the image is $v = \frac{\partial y(t)}{\partial t}$. So we have

$$v = \frac{f_y}{H-h} V$$

The speed V and of the vehicle can be estimated by computing the locus slope, v_g , of a ground point (e.g., the bottom of the rear or its shadow, i.e., $h = 0$) as

$$V = \frac{H}{f_y} v_g \quad (6)$$

Then the height h of the vehicle can be estimated using the locus slope of a point in the front, v_h :

$$h = H(1 - \frac{v_g}{v_h}) \tag{7}$$

The EPI approach is superior to the two-frame approach in simplicity and robustness. There is no correspondence problem. Speed is estimated using more than two points in a locus. All we need to do is to extract the (straight) locus and calculate the slope.

The length and width of a vehicle can be calculated by combining the information from both PVI and EPI. Sometimes the front and the rear of a vehicle cannot be presented in a single frame if the vehicle is too large. But there is no problem in the ST image approach. The length L of a vehicle can be calculated as the production of speed V and the duration time T during which the vehicle is passing through the detection line (Actual calculation of time T will be given in subsection 3.3). Compensating for the projective distortion, the vehicle length can be further modified as

$$L = VT - h\,ctg\,\theta \tag{8}$$

The geometry is shown in Fig. 5 and an example is shown in the PVI in Fig. 4(1).

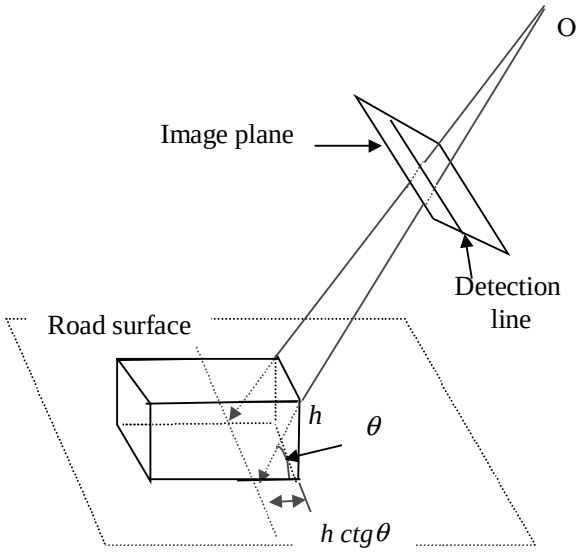


Fig. 5. Length computation

Let x_L and x_R be the x coordinates of the left and right boundaries of the vehicle in the PVI, and h_L and h_R be the heights of the corresponding boundaries respectively. These heights can be decided by the following equation (Fig. 6)

$$(h_L, h_R) = \begin{cases} (h,0), \text{ if } x_L > x_R > 0 \\ (h,h), \text{ if } x_L \geq 0 \text{ and } x_R \leq 0 \\ (0,h), \text{ if } 0 < x_L < x_R \end{cases} \quad (9)$$

Hence the vehicle width can be calculated as (Fig. 7)

$$W = \frac{1}{f_x \sin\theta}[x_R(H - h_R) - x_L(H - h_L)] \quad (10)$$

In the example in Fig. 4(1), we have $h_L=0$ and $h_R = h$.

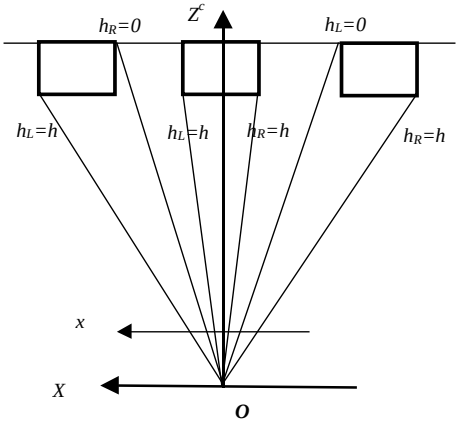


Fig. 6. Three cases for heights

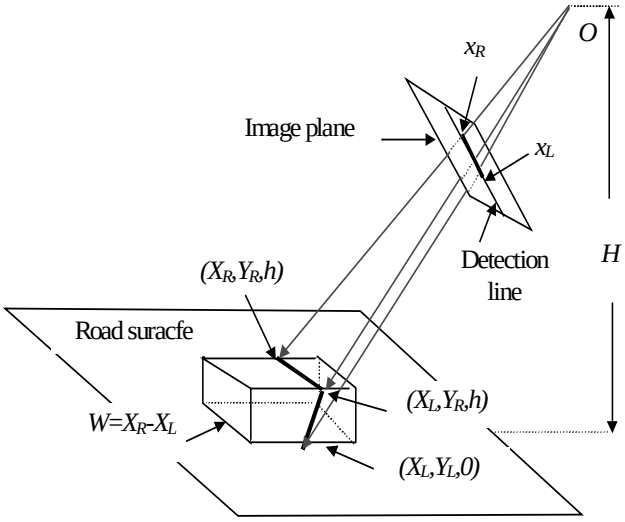


Fig. 7. Width computation

Based on these real-time measurements, other traffic parameters for each lane, such as class, volume, occupancy, headway, mean speed, etc., can be estimated without difficulty. The definitions of real-time parameters and statistical parameters are listed in Table 1 and Table 2 respectively.

Table 1. Real-time parameters

Parameter	Notation in equations	Definition and unit	Symbol in Fig. 13
length	L	3D size measured as L×W×h (m^3)	Length
width	W		Width
height	H		Height
speed	V	Measured in km / hr	Speed
class		Classified by size e.g., small, media, huge,...	Kind

Table 2. Statistical parameters

Parameters	Definitions	Symbol in Fig. 13	Unit
Volume	Number of vehicles detected during the time intervals	Volume	number
occupancy	lane occupancy measured in percent of time	Occupy	%
headway	Average time interval between vehicles	Headway	seconds
mean speed	average vehicle speed in the lane	M-Speed	km/hr

3.2. System calibration

In order to obtain the metric information for a vehicle, the camera system should be calibrated first. H, f_x, f_y and θ can be easily decided by a simple calibration procedure using the PVI and rectified EPI of a moving vehicle of known size (length, width and height).

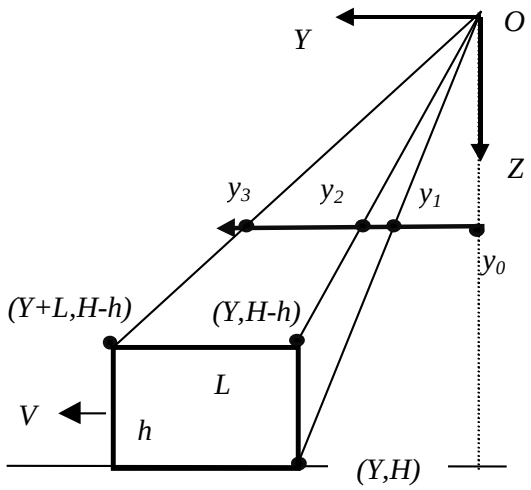


Fig. 8. Calibration of the EPI

First, H, y_0 and f_y can be decided by the loci of three image points in the EPI, giving the height (h) and length (L) of a cuboid-shaped vehicle (Fig. 8). The coordinates of the vehicle need not to be measured. For three particular points *(X,Y,H)*, *(X,Y,H-h)* and *(X,Y+L,H-h)* on the vehicle, and their y coordinates y_1, y_2 and y_3 in the EPI, we can obtain the following results from equations (5) and (7):

$$\begin{cases} H = h/(1-\dfrac{v_1}{v_2}) \\ y_0 = y_2 - \dfrac{H(y_2 - y_1)}{h} \\ f_y = \dfrac{h(y_1 - y_0)(y_3 - y_2)}{L(y_2 - y_1)} \end{cases} \tag{11}$$

where $v_1 = \dfrac{\partial y_1}{\partial t}, v_2 = \dfrac{\partial y_2}{\partial t}$, and f_y is in pixels. It should be noted that only the height H and length L are used in the above equations, but the coordinates X and Y are not included.

Similarly, giving the width (W) and length (L) of a moving vehicle and its PVI and EPI, f_x and θ can be estimated from equation (5) to (10)

$$\begin{cases} \theta = \arctan(\dfrac{h}{VT - L}) \\ f_x = \dfrac{1}{W \sin\theta}[x_R(H - h_R) - x_L(H - h_L)] \end{cases} \quad (12)$$

where T, V, h, (x_R,h_R) and (x_L,h_L) can be obtained from the PVI and the EPI, and f_x is in pixels.

3.3. Real calculations and error analysis

In this subsection, we give a theoretical analysis of the metric estimation of our ST approach, given the image resolution, vehicle speed, vehicle size, temporal sampling rate, and the localization errors. It is helpful to give an idea of error bound of the ST approach, and it also reveals some aspects of what are the real computations for the metric measures. The real error statistics need far more engineering work of on-site tests, and it will be briefly discussed in Section 5.

(1) Width estimation

The width of a vehicle, W, is estimated in a PVI. From equations (4) and (10), the error of width estimation can be roughly calculated as

$$\delta W = \frac{H}{f_x} \cdot \delta w \quad (13)$$

where δw is the localization error of width in the PVI.

(2). Length estimation

From equation (8), the length can be roughly estimated as $L = VT$, where speed V is estimated in an EPI and the duration T is calculated in the PVI. Hence the length error can be computed as

$$\delta L = V \cdot \delta T + T \cdot \delta V \quad (14)$$

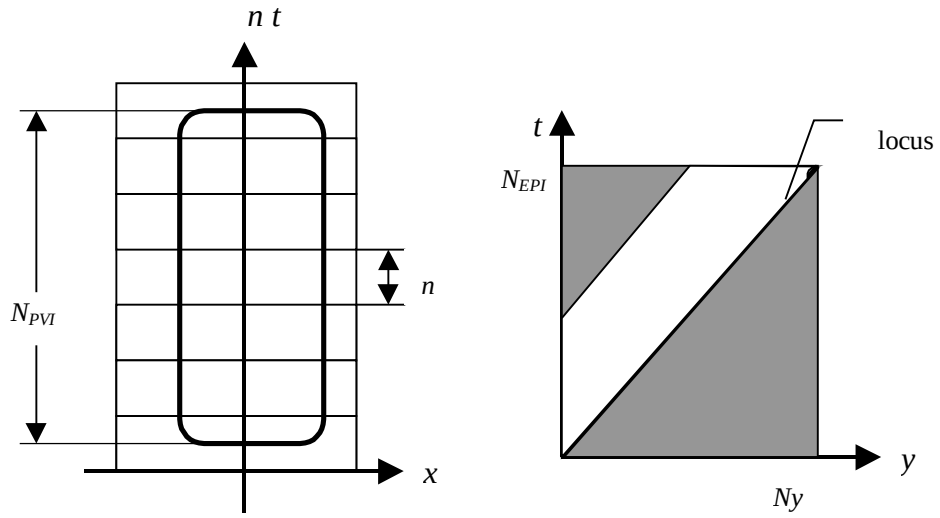
The estimation of speed error δV will be given later. Here we discuss how to estimate duration error δT. To avoid missing a fast-moving vehicle in a single "detection line" of the conceptual PVI, we use a "detection slice" of n scanlines per frame. Hence an extended PVI is constructed by extract n scanlines from each successive frame. The translation of the vehicle in the image should be less than n, otherwise parts of the front and/or rear of the vehicle would be missed in

the extended PVI (Fig. 9 (1)). Therefore n varies according to the mean speed ($\bar{V}$) of a lane during a certain time interval

$$n \geq f_x \frac{\bar{V}\tau}{H} \tag{15}$$

where τ is the temporal sample rate. Ideally, our system can monitor the road at a speed of 50 fields per second for the PAL system, so we have $\tau = 1/50$ second / frame (s/f). Note that we use f_x instead of f_y in equation (15) since f_x is used for the PVI geometry. Assume that the vehicle occupies N_{PVI} lines in the extended PVI (Fig. 9(1)). Then the duration T can be calculated as

$$T = \frac{N_{PVI}}{n}\tau \tag{16}$$



(1) Extended PVI (2) EPI

Fig. 9. Metric Measures in PVI and EPI

We therefore have

$$\delta T = \frac{\delta N_{PVI}}{n}\tau \tag{17}$$

where δN_{PVI} is the localization error of the length in the extended PVI. From equation (16) and (8) with h=0, given the speed of the vehicle, the number of lines of a vehicle can be predicted as

$$N_{PVI} = \frac{nL}{V\tau} \geq f_x \frac{L}{H}\frac{\overline{V}}{V} \tag{18}$$

Equation (18) give us an idea of how many lines a vehicle occupies in the extended PVI. The number of lines is inversely proportional to the speed; however it will be a function of the vehicle's length L only if the scanlines per frame are dynamically changed with the speed itself, i.e., $\overline{V} = V$.

(3) Speed estimation

We can also estimate the number of frames in which a vehicle covers the tracking line in the effective field of view in y direction, L_{FOV} (Fig. 9(2))

$$N_{EPI} = L_{FOV} / V\tau \tag{19}$$

From equation (6) we have

$$V = \frac{H}{f_y} \cdot \frac{s}{\tau} \tag{20}$$

where s is the slope of the image locus of a ground point measured in pixels. So the absolute error and the relative error of speed can be computed as

$$\delta V = \frac{H}{f_y \tau} \cdot \delta s, \quad \frac{\delta V}{V} = \frac{\delta s}{s} \tag{21}$$

If only two points on the locus are used to calculate the slope, then we have (Fig. 9(2))

$$s_2 = \frac{N_y}{N_{EPI}}, \quad \delta s_2 = \frac{N_{EPI} \cdot \delta N_y + N_y \cdot \delta N_{EPI}}{N_{EPI}^2} \tag{22}$$

where N_y is the effective image resolution in y direction, δN_y and δN_{EPI} are the localization errors for N_y and N_{EPI} respectively. The accuracy is improved in our approach in that the slope is estimated by fitting the locus using least square mean method. The procedure can be considered as approximately calculating the average of $\frac{N_{EPI}}{2}$ slope values s_2 that come from

$\frac{N_{EPI}}{2}$ pairs of points. If the errors δs_2 are independent Gaussian noises, then the error of s can be reduced to

$$\delta s = \frac{\frac{N_{EPI}}{2}\delta N_y + \frac{N_y}{2}\delta N_{EPI}}{\left(\frac{N_{EPI}}{2}\right)^2} \Big/ \left(\frac{N_{EPI}}{2}\right) = \frac{4}{N_{EPI}}\delta s_2 \tag{23}$$

(4). Height estimation

From equation (7) the absolute error of height estimation can be calculated as

$$\delta h = H\frac{s\cdot\delta s_v + s_v\cdot\delta s}{s_v^2} \tag{24}$$

where s_v is the slope of the locus of the point on the vehicle's roof measured in pixels/frame.

Table 3. Error analysis*

V (km/hr)	0	10	50	80	120	160
n	1	1 (≈1.08)	6 (>5.39)	9 (>8.62)	13 (>12.9)	18 (>17.2)
N_{PVI}	/	72	86	81	78	81
T (s)	/	1.44	0.287	0.180	0.120	0.09
N_{EPI}	/	238	48	30	20	15
δV (km/hr)	0	2.71e-3	0.204	0.787	2.56	5.96
δV/V	/	0.03%	0.41%	0.98%	2.13%	3.72%
$V\delta T$ (m)	0	0.111	0.093	0.099	0.103	0.099
δL (m)	/	0.112	0.109	0.138	0.188	0.248
δW (m)	0.103	0.103	0.103	0.103	0.103	0.103
δh (m)	/	0.004	0.059	0.142	0.312	0.556

* N_{PVI} is estimated when L = 4 m and δh is estimated when h = 2 m. Note that V and δV are given in km/hr, and should be converted into m/s for other computations.

Suppose H=10 m, f = 5*256 /6.6 (pixels) (i.e., a 5 mm focal length camera with target size of 6.6 mm and an image size of 256 pixels), N_y = 256, L_{FOV} =13.2m. Assume that the localization errors are 2 pixels for all the measurements in images, i.e., δw = 2, δN_{PVI} =2, δN_y =2, δN_{EPI} = 2. Table 3 gives theoretical results of speed and size estimation under different vehicle speeds. The length error is estimated when L = 4 m. The absolute error of height is computed for a 2-meter high vehicle at different speeds. Note that the width error and duration error ($V\delta T$) are irrelevant to the vehicle's speed; however, the length error is a function of speed. The number (n) of scanlines in the detection window is estimated using the given speed V. For references, N_{PVI} and N_{EPI} are also given in Table 3.

4. Vehicle Separation and Locus Tracking

4.1. Vehicle extraction

Vehicles are separated from the road surface, shadows and vehicle lights by fusing multiple cues including intensity, spatio-temporal changes, and models of vehicles and the environment. The basic principles are summarized as follows.

1) Background subtracting. Intensity differences always exist between vehicles and a road surface. In the daytime those portions whose intensities are higher than that of the road are directly classified as belonging to a vehicle. Those portions with lower intensities need further analysis. Intensities of shadows are always lower than the intensity of the road. At night the intensities of areas onto which a vehicle's lights project are higher than those of the road surface, therefore further investigation is needed.

2) ST differentiating. There are rich intensity changes inside the vehicle, especially in the longitudinal direction of a vehicle and around its boundary; while the intensities of shadow areas are nearly constant and areas where the vehicle's lights project have no distinctive edges.

3) Modeling of vehicles, shadows and lights. The symmetry and the length of a vehicle as well as the minimum headway between two vehicles, can be used as models to group the

different portions into a vehicle. The direction and size of a shadow area can be estimated by using the knowledge of the sun's position at different times of day. The fact that the headlights of a vehicle always project in front of the vehicle can also be used for vehicle segmentation.

4) Merging multiple lanes. To handle the situation where a vehicle is crossing over lanes or shadows of vehicles project onto other lanes, the detection line covers all the related lanes and all the lanes are considered together.

Based on the above principles, we have designed an algorithm for vehicle detection and separation. Theoretically, a single detection line *g(x,t)* is processed at each time t. The background is initialized as *b(x)* and is updated according to different situations.

Step 1. *Image pre-processing and background subtracting*

In order to reduce the image noise and the effect of inevitable vibration of the camera, the original PVI, the ST image *g(x,t)*, is first smoothed in time and space

$$f(x,t) = \sum_{(i,j)\in S} g(x+i,t+j) \tag{25}$$

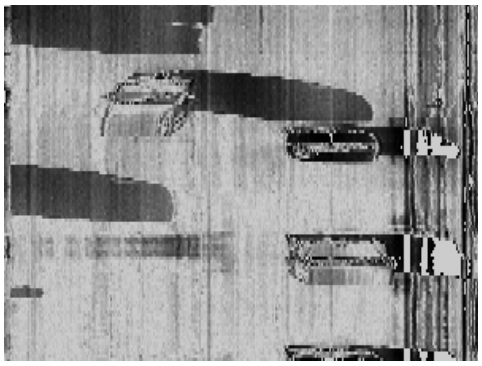
where *S* is the smoothing neighborhood. After smoothing, the background is subtracted from the intensity image *f(x,t)*

$$I(x,t) = f(x,t) - b(x) \tag{26}$$

Step 2. *ST differentiating*. The differentiation-of-subtraction (DoS) image *d(x,t)* is computed as

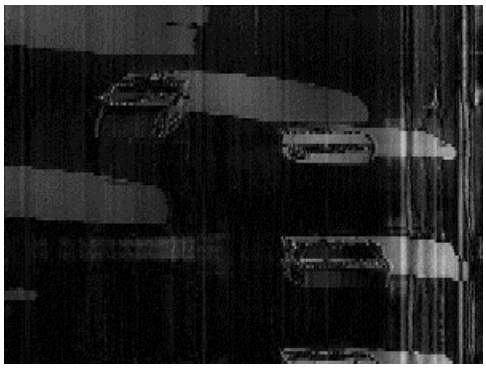
$$d(x,t) = a\left|\frac{\partial I(x,t)}{\partial x}\right| + b\left|\frac{\partial I(x,t)}{\partial t}\right| \tag{27}$$

where *a* and *b* are weights for magnitudes of gradient components in *x* and *t* directions respectively. In practice we set $a > b$ in order to repress the effect of shadows, since edges of a vehicle in the longitudinal direction are more fertile (Fig. 10).

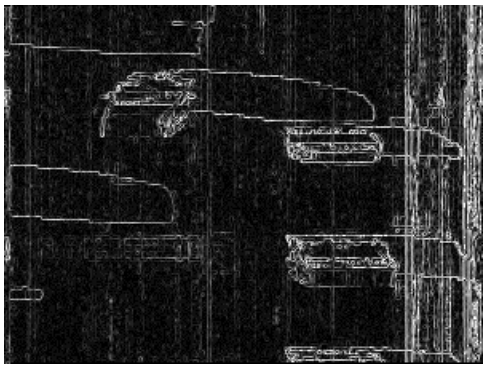


|<- lane1 ->|<- lane2 ->|

(1) PVI *f(x,t)*

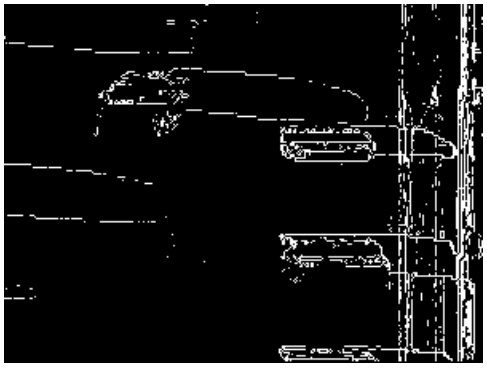


(2) Absolute Subtraction |*I(x,t)* |

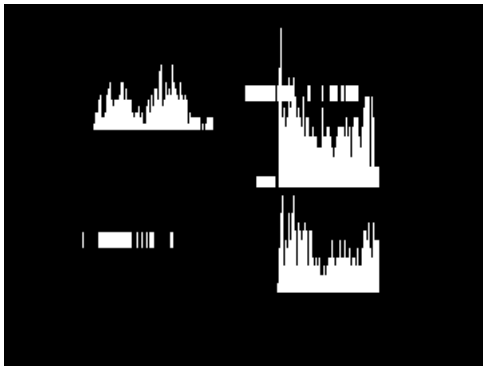


(3) DoS image *d(x,t)*

(4) Thresholding |*I(x,t)*|



(5) Thresholding the DoS



(6) Temporal projection

Fig. 10. Vehicle separation from heavy shadows in daytime

Step 3. *Vehicle detection*. For a given time t, a image point x, whose subtraction *I(x,t)* is greater than a given threshold T_I in daytime or whose DoS *d(x,t)* is greater than a given threshold T_d is regarded as a (possible) point on the vehicle, and a binary map is formed as

$$e(x,t) = \begin{cases} 1, & d(x,t) > T_d \text{ or } (I(x,t) > T_I \text{ in daytime}) \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

In equation (28), intensities contribute to the detection for the brighter (parts of) vehicles, while the DoSs play a vital role in general cases in separating the vehicle from the road, shadows and headlights.

Spatio-projection is calculated along the detection line for each line as

$$q(t) = \sum_{x \in W_W} e(x,t) \quad (29)$$

where W_W is the width of the detection window for a lane. If *q(t)* is greater than a certain threshold N_{Wid} (the minimum number of pixels in vehicle width), a possible vehicle line segment is labeled.

Step 4. *Grouping and separation*. In a 2D PVI *e(x,t)*, those labeled lines that satisfy the criterion of minimum headway between vehicles, minimum duration time and minimum width of a vehicle, are grouped to form a possible vehicle region. The length of the region along *t* axis is the duration time (*T*) of this possible vehicle across the detection line.

During the time period *T*, we project the *e(x,t)* in the time axis as

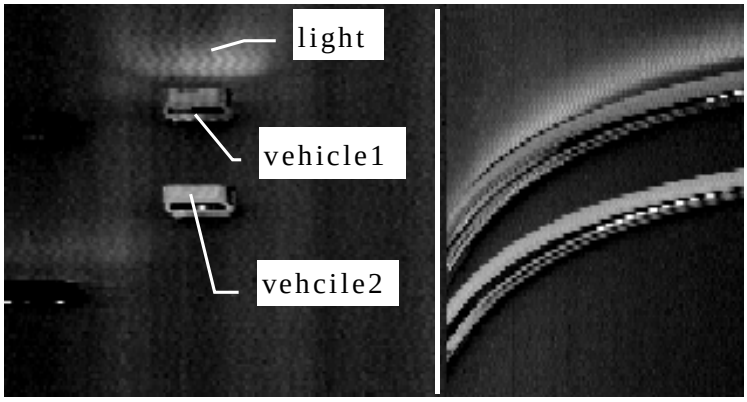
$$p(x) = \sum_{t \in T} e(x,t) \quad (30)$$

The left and right boundaries of the vehicle in the PVI are searched from both sides, and are estimated as x_L and x_R if $p(x_i)$ $(i=L,R)$ are greater than N_{Len}, the minimum number of pixels in vehicle length.

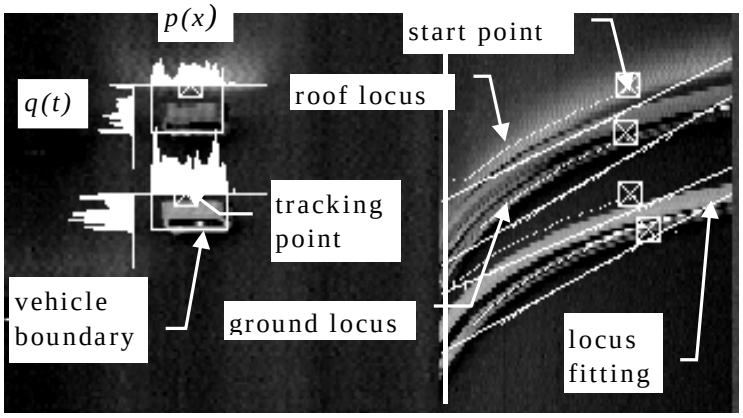
The image region of size $(x_L - x_R) \times T$ in the PVI is further analyzed to eliminate the shadows or to separate two side-by-side vehicles, using the knowledge of the normal length/width ratio, the approximate symmetry of the vehicle's image, the model of daytime sunlight and shadow,

and the headlight model at night. In this way, the final bounding box, $(x_L - x_R)\times T$, of the vehicle in the PVI is determined and the width can be calculated by using equations (9) and (10).

Fig. 10 shows an example in daytime operation. It can be seen in Fig. 10(1) that heavy shadows of cars are cast on the road. Simple thresholding of the subtracted image cannot separate the vehicles from the shadows (Fig. 10(2), Fig.9(4)). However, the shadow regions are nearly intensity-smooth and edges of vehicles are distinct and fertile. Hence the differentiation-of-subtraction (DoS) method followed by temporal- and spatio-projections are quite effective. Fig. 10(5) shows the thresholding result of the DoS image in Fig. 10(3). In Fig. 10(6) temporal projections are calculated for five possible vehicle regions within the two lanes annotated in Fig.9(1). In this example, the fact that shadows are cast on the right of the vehicles is used to verify the separation of shadows from vehicles.



(1). Original PVI (2) Original EPI



(3) Vehicle separation (4) Locus tracking

Fig. 11. Vehicle separation and loci extraction at night

Fig. 11 shows an example of vehicle detection and separation during nighttime operations. The PVI in Fig. 11 (1) shows strong beams of headlights from the first vehicle. Fortunately most of the headlight's edges are not intuitive except the edge of the low beam of headlights near the vehicle. The spatio- and temporal-projections are superimposed in the PVI of Fig. 11(3). The two rectangles represent width×duration for the two vehicles. The estimated duration of the first vehicle from the basic procedure will be larger than the actual duration. It can be further refined by verifying that the region in front of the detected vehicle is really of headlights, and then using the headlight model to guide the separation. Notice that there is a valley in spatio-projection *q(t)* between the front of the first vehicle and the edge of the headlights.

4.2. Background updating

To make the system adaptive to varying light conditions, the background should be updated from time to time. In our approach, only a small fraction of the image (several scan lines for the PVI) needs to be updated. The background intensity of the PVI is initialized and then updated whenever the detection lines are not covered by vehicles, shadows and vehicles' light. The background is updated in the following three cases:

Case (1). The background changes gradually. Since the illumination change of the background (i.e., the road surface) is slow, we use sufficient time slices (e.g., F = 100 frames) before time t to update the background $b(x)$. First, a weight function $w(t)$ is estimated for each time slice, as

$$w(t) = \sum_{x \in W_W} | b(x,t) | \tag{31}$$

For the current time t, the background is modified as

$$b_{new}(x) = [\frac{\sum_{t_i=t}^{t-F} G(w(t_i))\, f(x,t)}{\sum_{t_i=t}^{t-F} G(w(t_i))} + b_{old}(x)] / 2 \tag{32}$$

where $G(\cdot)$ is the Gaussian function with $G(0)$=1.0 and $G(\pm T_I * W_W)$=0.01, and is very small if the road has an average intensity change greater than T_I due to a passing vehicle or shadows. In real

implementation, $G(\cdot)$ is calculated off-line to generate a look-up table, and the summation in equation (32) can be computed iteratively from time *t-1* to time *t*.

Case (2). The background changes abruptly, for example, when a piece of cloud passes over the road in daytime, or when the streetlights are turned on in the evening. If duration time *T* of an assumed "vehicle" is greater than the maximum duration for the largest vehicle, then an abrupt background change is assumed. In this case an alarm is given to indicate that the system enters a recovery period. We make the assumption that the road image before and after an abrupt illumination change satisfies the following linear relation

$$b_{new}(x) = \alpha b_{old}(x) + \beta \tag{33}$$

Differentiating both sides with *x* we have

$$b'_{new}(x) = \alpha b'_{old}(x) \tag{34}$$

Hence α can be estimated as

$$\alpha = \sum_{x \in W_W} b'_{new}(x) \Big/ \sum_{x \in W_W} b'_{old}(x) \tag{35}$$

During the recovery period, the average difference between the spatial gradient $f'(x,t)$ of the current image *f(x,t)* and that of the old background image *b(x)* is calculated as

$$d_g(t) = \frac{1}{W_W} \sum_{x \in W_W} | f'(x,t) - \alpha_t b'_{old}(x) | \tag{36}$$

where $\alpha_t = \sum_{x \in W_W} f'_{new}(x,t) \Big/ \sum_{x \in W_W} b'_{old}(x)$ (37)

It can be seen that if the current image is of road surface without vehicles, $d_g(t)$ should be very small under the linear illumination change model of equation (33). Therefore the road discrimination criterion is defined as

$$\sigma(\mathrm{t}) = \begin{cases} 0, & d_g(t) > \mathrm{MaxDg} \\ 1, & d_g(t) \le \mathrm{MaxDg} \end{cases} \tag{38}$$

where MaxDg is the predefined maximum gradient difference for road image, and the background is refreshed as

$$b_{new}(x) = \sum_{x \in T_r} f(x,t)\sigma(t) \Big/ \sum_{x \in T_r} \sigma(t) \qquad (39)$$

where T_r is the recovery time interval. During the recovery period T_r and the previous "false vehicle" period T, the passing vehicles may be missed by the system. So the PVI section during these periods is saved and then be reprocessed afterward using the refreshed background information.

Case (3). The background changes frequently. This situation may occur when energy-saving street lamps are used at night or the camera vibrates violently due to the passing of heavy vehicles. The intensities of the background change periodically while the light flashes (or the camera vibrates) at a certain frequency, so the short headway and vehicle duration are detected frequently. In this case the PVI is smoothed using a 5×5 or larger Gaussian operator according to the frequency.

The intensity gradient T_I and the gradient threshold T_d is also changed according to the changes of the background, and the average difference between vehicles and the road surface at different times of day.

4.3. Loci extraction

For real-time implementation, we use re-projection look-up tables (RLUTs) to map the original epipolar lines to rectified ones. When vehicles move bottom-up in the image and the detection line is set at the bottom of the image, the lateral position of the epipolar line for a vehicle is selected adaptive to the position of the vehicle inside the lane. So we have several (e.g., 2) LUTs for each lane.

While the EPI is being formed, the locus of the front and the rear of a vehicle are tracked at the same time. Theoretically, it seems advantageous to rectify the EPI before we track the locus, since the locus is straight after the rectification. However, in practice, the rectification procedure will degrade the resolution of the image. We therefore re-project only the locus points after we have tracked them in the original EPI; straight-line constraints can also be applied to guide the tracking. The tracking of the loci of a vehicle's front and rear is relatively easy since they are the

border edges of the loci's pattern of the vehicle. The cost function for the front or the rear loci tracking of a gradient image *G(y,t)* of the EPI is

$$E = \alpha\left|y - y^*\right| + \beta\left|G - G^*\right| + \gamma G + \kappa d \qquad (40)$$

where α, β, γ and κ are the normalized weight coefficients for the measurements of locus's straightness, gradient similarity, gradient magnitude and "border-ness" for the point *(y,t)*. In equation (40), y^* is the estimated value for *y* using the straight locus constraint in the rectified EPI; G^* is the average gradient value of the tracked points; and d is the distance between the point *(y, t)* and the outmost edge-like point. The start point of a locus is determined using the detecting result in the PVI (labeled as "tacking point" in Fig. 11). The locus is tracked by using a heuristic search method with the cost function expressed in equation (40). The values of α, β, γ and κ are changed during tracking and in different light conditions. For example, at the beginning of tracking, we set α=0 and β equals a small value, since a straight locus constraint based on a few points is not reliable. The weights of locus straightness and gradient similarity increase when a certain number of tracking points has been obtained. In nighttime operation the weight κ is set to a relatively small value to reduce the negative effect of the vehicle's headlights.

When enough points have been obtained for both loci, Two straight lines are fitted to the rectified loci. The speed and height can be obtained using equations (6) and (7), and the length can be estimated using equation (8). Fig. 11 also shows the loci tracking and fitting results of front and rear edges. The tracked points (white curves) are moved up 2 pixels for clarity of the superimposed display.

5. Real-time Implementation

5.1. System overview

A Visual System for Automatic Traffic Monitoring, VISATRAM, has been implemented on a PC486 with a frame grabber OFG-100 by Imaging Technology Inc. The resolution of each frame is 576x768 pixels. VISATRAM can monitor two or three lanes simultaneously in the current implementation. More lanes can be processed if a faster computer is used. A single

detection slice window (conceptually a detection line) covers all lanes, therefore a single PVI is formed. On the other hand one EPI is formed for each lane. In practice the PVI and EPIs are processed while they are being formed, so the system is operated at frame rate (i.e., 25 frames per second) while the captured images are active. Fig. 12 gives the system diagram. An example of real-time operation is shown in Fig. 13. Two lanes were processed so one detection line and two tracking lines are superimposed on the image. While a car was passing through the detection line in the right lane, a rectangle is superimposed on the detection slice window. The display-box of real-time parameters (Table 1) just above the car were those of the last vehicle. The statistical parameters (Table 2) of each lane are on the top of the display-box of the real-time parameters.

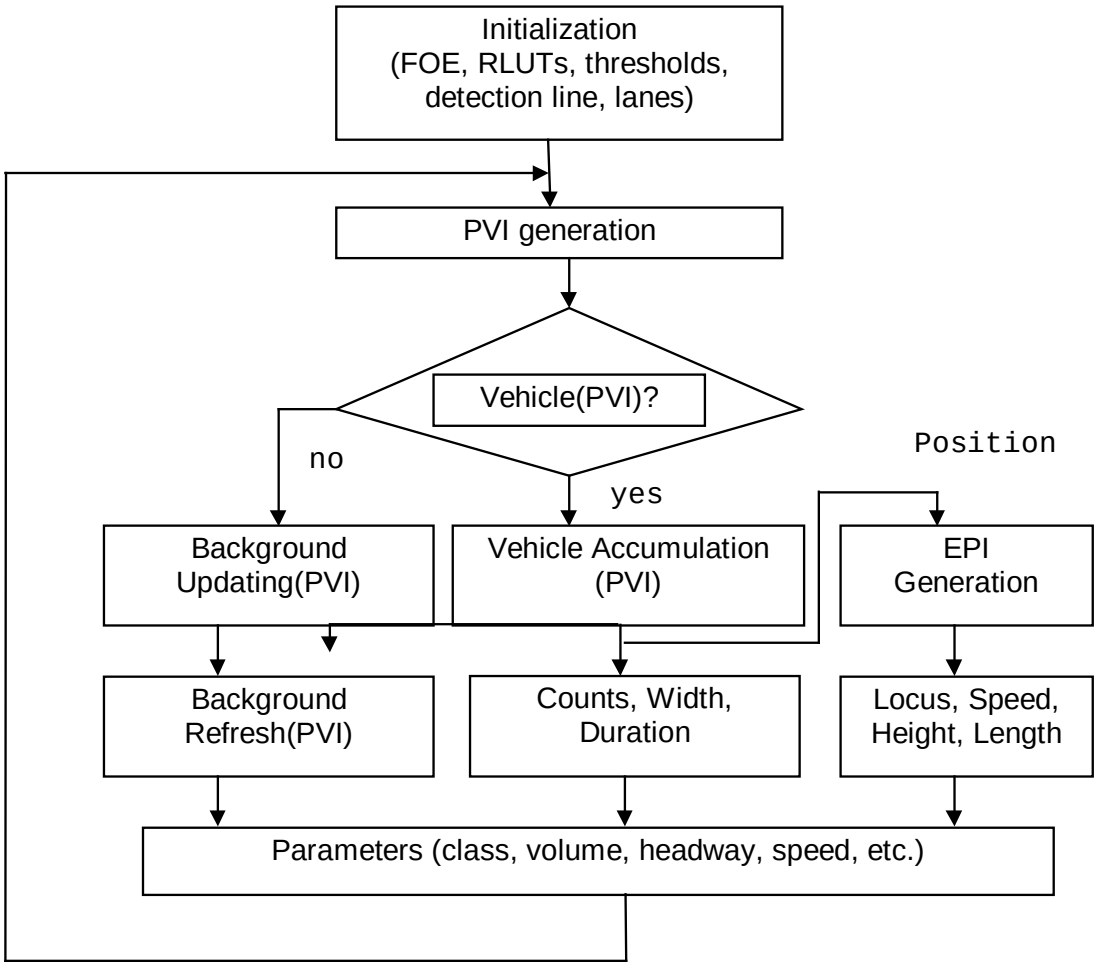


Fig. 12. System diagram

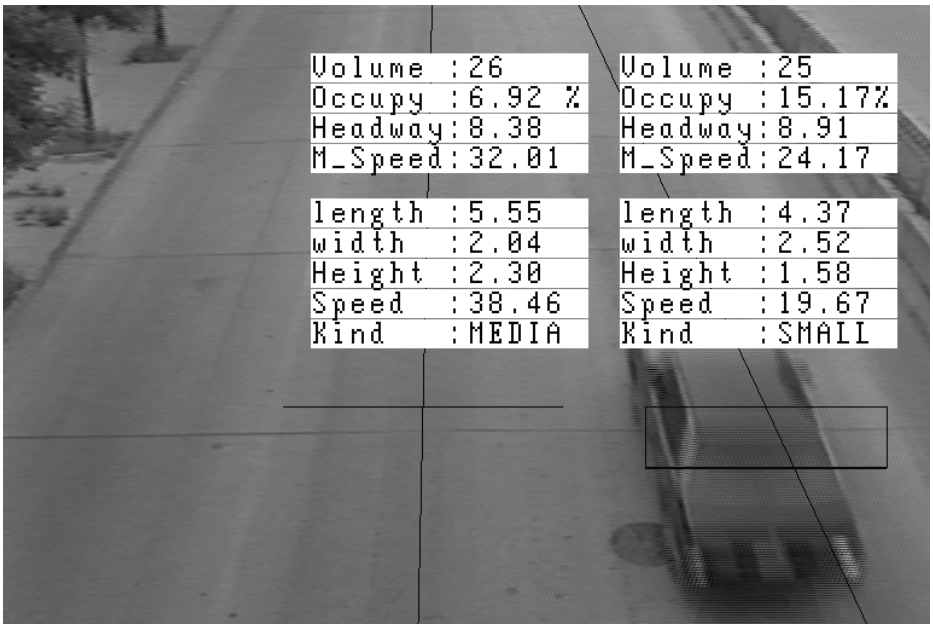


Fig. 13. An example of real-time operation

5.2 Real-time performance

The aim of VISATRAM is to obtain traffic parameters in real-time using inexpensive hardware. This is realized by processing only the most relevant data along a detection line and a few epipolar lines. The execution of the system can be divided into three steps.

Step 1. *Image acquisition and transfer.* For each frame of the video sequence, several scanlines around the detection line are transferred into the memory to form the panoramic view image (PVI). The pixels along each epipolar line of the corresponding lane are read and transformed to an epipolar plane image (EPI). The transformations are carried out by LUT operations.

Step 2. Image processing. The operations include ST image smoothing, background subtracting and ST differentiation, and background updating.

Step 3. Vehicle detection and parameter estimation. This part consists of analyzing the PVI and EPIs to obtain the 3D size and speed parameters.

We have carried out experiments on hours of traffic video sequences at different times of day – in the morning with long-casting shadows, at the noon with heavy shadows, in the late

afternoon when the sunlight is dim, and in the evening when street lamps are turned on. In these experiments two or three lanes are processed by a PC 486/66. The size of the detection window is n (1~5 scanlines)×768 (pixels) and the convolution kernel for smoothing is 3×3. The detection rate is almost 100% (i.e., VISATRAM seldom misses a vehicle). Roughly speaking, the system works well for vehicle classification and speed estimations in the daytime, and the measurements are acceptable at night. However the judgement is made only by human observation since ground truth is not available. Typical experimental results are shown in Table 4 and Table 5. Table 4 shows a statistical result for a 12-minute video of two-lane traffic monitoring. The real-time performance shown in Table 5 indicates that the experimental system can work at frame rate. It should be noted that more than half of the processing time is spent in the image acquisition step because the I/O mapping mode has to be used for data transfer from the frame buffer of the OFG-100 to the main memory. This problem can be easily solved using a frame grabber with direct-mapped frame memory. Field rate performance (50 fields per second) is not a problem using the currently available Pentium PC.

Table 4. Statistical results in one of the experiments

Lane	Volume	Headway (s)	Occupancy (%)	Mean speed (km/hr)
1	105	6.86	11.02 %	42.63
2	89	8.08	11.04 %	32.87

Table 5. Real-time performance

Total time(s)	Total frames(f)	Frame rate(f/s)	Step 1 (s)	Step 2(s)	Step 3(s)
726.39 (100%)	17949	24.7 (real-time)	455.74 (62%)	212.42 (30%)	58.23 (8%)

lane1 lane2

(1) PVI (2) EPI of lane 1 (3) EPI of lane2

Fig. 14 Compact visual representation

5.3 Compact visual representation

The 2D ST images, the PVI and the EPIs, are highly compressed visual representations for most of the traffic information. Fig. 14 shows part of the ST images for a real highway traffic scene. The representations can be used for traffic verification and off-line experiments and research. Along with other image compression techniques, such as JPEG, huge traffic image sequences can be highly compressed on the hard disk for future use. Further work is needed to evaluate the system performance by comparing the results to the ground truth; the representation of PVIs and EPIs provides an effective way to find the rough "ground truth" by manual annotation of these images.

6. Conclusions

We have developed a visual traffic monitoring system, VISATRAM, based on 2D spatio-temporal image analysis. Experimental results with real traffic images are very encouraging. The common difficult situations, such as daylight variation, shadows, and nighttime operations, have been tested. The system is also very cost-effective and real-time performance has been achieved on a simple 486/66 PC.

Acknowledgements

The first author is grateful to Ms. Arden Phillips at University of Massachusetts at Amherst for her help in improving the presentation of this paper. This work was supported by China Advanced Research Project during 1993-1997. An earlier version of this paper is presented in the 1996 IEEE Workshop on Application of Computer Vision [15].

References

[1] Hockaday, S, Chatziioanou A, Nodder R and Kuhtenschmidth S, Evaluation and comparison of video image processing systems for traffic detection, *Transp. Res. Board Preprint #920744*. Transp. Res. Board, Washington D C (1992).

[2] Michalopoulos, P, Vehicle detection video through image processing: the AUTOSCOPE system, *IEEE Trans. on Vehicular Techno.*, 40(1), (1991) 21-.

[3] Sal D'Agostino, Commercial machine vision system for traffic monitoring and control, *SPIE* vol. 1615 (1991) 180-186

[4] Takatoo, M., et al, Traffic flow measuring system using image processing, *SPIE* vol. 1197 (1989) 172-180

[5] Sullivan, G.D, Visual interpretation of known objects in constrained scenes, *Phil. Trans. R. Soc. Lond. B.*, B(337) (1992) 109-118.

[6] Yuan, X, Lu Y-J and Sarraf, S, Computer Vision System for Automatic Vehicle Classification, *Journal of Transportation Engineering*, Vol. 120, No.6 (Nov/Dec 1994) 1861-876.

[7] Jolly, M-P D, Lakshmanan S and Jain A K, Vehicle Segmentation and Classification Using Deformable Templates, *IEEE Trans. PAMI*, Vol. 18, No. 3 (March 1996) 293-308.

[8] Ferrier, N J, Roew S M and Blake A, Realtime traffic monitoring, *Proc. IEEE Workshop on Application of Computer Vision* (1994) 81-88.

[9] Kilger, M, A shadow handler in a video-based real-time traffic monitoring system, in: *Proc. IEEE Workshop on Application of Computer Vision* (1992) 11-18.

[10] Zielke, T. et al, Intensity and edge-based symmetry detection with an application to car following, *CVGIP: Image Understanding*, Vol. 58, No 2 (1993) 177-190.

[11] Zheng, J Y and Tsuji S., Panoramic representation of scenes for route understanding, in: *Proc 10th-ICPR, IAPR* (June 1990) 161-167.

[12] Bolles, R C, Baker H H and Marimont D H, Epipolar plane image analysis: An approach to determine surface from motion, *Int. J. Computer Vision*, vol 1, no 7 (1987) 7-15.

[13] Nakanishi, T and Ishii, K, Automatic vehicle image extraction based on spatio-temporal image analysis, in: *Proc 11th ICPR, IAPR* (1992) 500-504.

[14] Bullock, D and Mantri S, Multimedia data model for video detection research, *J. of Transportation Engineering*, vol 121, no 5 (1995) 385-390.

[15] Zhu, Z, Yang B, Xu G and Shi D, A real-time vision system for automatic traffic monitoring based on 2D spatio-temporal images, in: Proceedings of the *3rd Int. Workshop on Applications of Computer Vision*, IEEE, Sarasota, Florida (Dec. 1996) 162-167.