MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

Video-Rate Image Processing System for an Autonomous Personal Vehicle System

Mayumi Ohzora, Tohru Ozaki, Shigeru Sasaki, Masumi Yoshida, and Yoshitaka Hiratsuka[†]

Fujitsu Laboratories Ltd., 1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, JAPAN
[†]Fujitsu Limited, 1015 Kamikodanaka, Nakahara-ku, Kawasaki 211, JAPAN

ABSTRACT

This paper describes a video-rate image processing system for an autonomous Personal Vehicle System (PVS). It detects white lines on a road including straight sections, curves, and intersections, and detects obstacles in the road by using binocular 3D vision. We developed white-line and obstacle detection algorithms that can be processed by the Fujitsu Integrated Visual Information System / Video-Rate Image Processing (FIVIS/VIP) [1][2]. The PVS operated autonomously at 60 km/h on straight sections, 15 km/h on curves, and 5 km/h through intersections on the testing ground. This paper details the image processing system, the algorithms, implementation, and results.

1. INTRODUCTION

To make a robot that operates autonomously requires the help of computer vision, artificial intelligence, and machine control. In the mid-1980s, the US Defense Advanced Research Project Agency (DARPA) started the Autonomous Land Vehicle (ALV) project, along the lines of the ALV project at Martin Marietta and Maryland University [3], the Navigation Laboratory (NAVLAB) project at Carnegie Mellon University [4], and the Vamors project at BW Munich University [5]. These systems use local processing in which an image is mapped on multiple small windows to separate components such as roads. This does not provide vehicles with sufficient information to adapt to all road conditions, however, including intersections and junctions. The ALV and NAVLAB use a laser range finder to get one in-depth image each half second in detecting objects. The image is accurate but the scanning speed is too slow for the vehicle to operate at high speed.

We developed a video-rate image processing system for a personal vehicle system PVS detects white lines by processing the entire input image and extracting depth information from binocular 3D views obtained by two TV cameras. This paper details the algorithms we developed to detect white lines and obstacles and the results of our experiments and evaluation.

2. SYSTEM CONFIGURATION

In the visual information processing system we developed for the PVS (Figure 1), the white-line detector consists of three monochrome TV cameras, a camera switcher, and FIVIS/VIP for image processing. FIVIS/VIP detects the white lines on roads based on the algorithm explained in Section 3. The controller calculates the locations of white lines and operates the camera switcher to handle curves and intersections. The obstacle detector consists of two monochrome TV cameras, FIVIS/VIP for video-rate binocular 3D vision, and a second controller. FIVIS/VIP calculates parallactic images from those obtained through the cameras. The controller calculates the depth to the object using these parallactic images and the cameras' location and angles. Depth and road-area data obtained by the white-line detector are compared to determine the location of three-dimensional objects that may be obstacles. Only information on the location of white lines and obstacles is sent to the vehicle controller.

3. DETECTION ALGORITHM

3.1 Definition of Coordinate System

We defined the coordinate system as shown in Figure 2. The XZ plane (Y = 0) is the road surface and the XY plane (Z = 0) the area in front of the vehicle. The image coordinate system has an xy plane on the image plane with the scanning line direction on the x axis perpendicular to the y axis. The

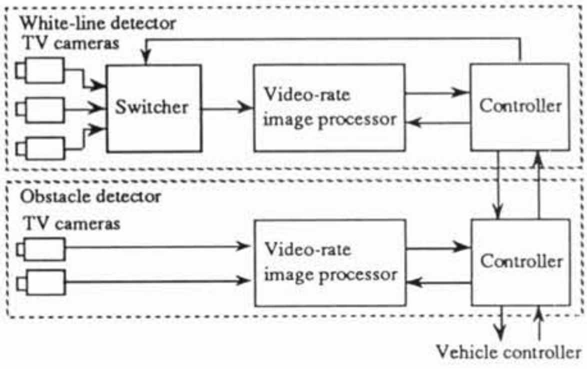


Fig. 1 System configuration

center camera for white-line detection is parallel with the x and X axes, with the lens focusing on point A on the Y axis. If a line is drawn from central point p on the image plane perpendicular to the Z axis to cross the Z axis at point P, the straight line p-P is called the camera's visual axis.

3.2 While-Line Detection Algorithm

White-line detection (Figure 3) using FIVIS/VIP involves the following steps:

1. Area identification

An image is binarized based on a threshold value determined from the difference in contrast between the white line and the road, and the area is labeled (numbered). Only whitish areas are extracted.

2. Candidate extraction

The histogram of a labeled image is calculated to determine the size of each area.

3. White-line determination

White-line candidates are searched for starting from origin o on the xy plane and moving toward ①, then ② on the x coordinate. If a whitish area of the preset threshold size or larger is detected, the system tracks the area as shown at ③ and ④. If no such areas are found in ① and ②, the search start point is shifted one meter in the direction of the y axis and the operation is repeated. For a broken white line, the area is a bit smaller but can be detected by searching an area one to four meters away from the edge point. At an intersection, the edge point is located at the side of the image, so an area about 4 meters wide (the road width) away from the point in the direction of the y axis is searched.

4. White-line location determination

Once the coordinate values on the projected xy plane are determined, the white-line location on the XYZ space is calculated by using camera location and angle and lens parameters. Here, the digital image resolution is R_x in the x direction and R_y in the y direction, the TV camera height H, and the depression angle θ. Lens parameters are horizontal view angle φ_H and vertical view angle φ_V. The white-line location coordinate on the plane is derived using equations 1 and 2:

$$Z = H \tan[\frac{2\pi\theta}{360} - \tan^{-1}\{(1 - \frac{2}{R_y}y)\tan(\frac{\pi\varphi_V}{360})\}] \qquad (1)$$

$$X = \frac{2}{R_y}\sqrt{y^2 + H^2}(\frac{R_x}{2} - x)\tan(\frac{\pi\varphi_H}{360}) \qquad (2)$$

3.3 Obstacle Detection Algorithm

The algorithm behind the binocular 3D vision system basically identifies the corresponding horizontal edges. We installed two TV cameras at different heights on the vehicle. The edge image in Figure 5, for example, has a pair of vertical positive edges (from dark to bright) and a pair of vertical negative edges (from bright to dark). If distances D1 and D2 between the positive and negative edges are not equal, the

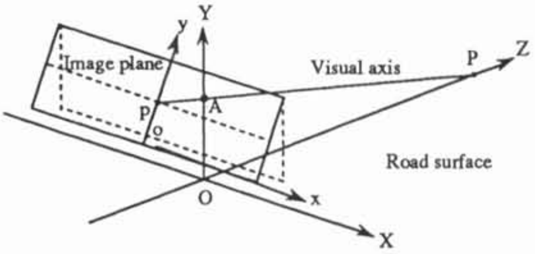


Fig. 2 Coordinate definition for TV camera

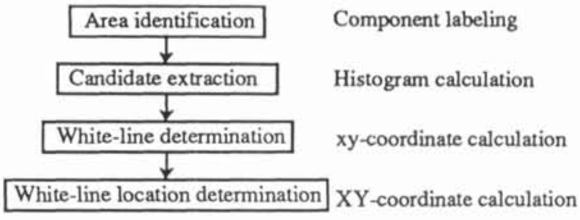


Fig. 3 White-line detection

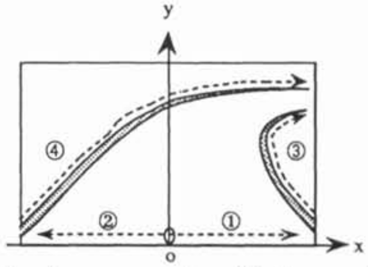
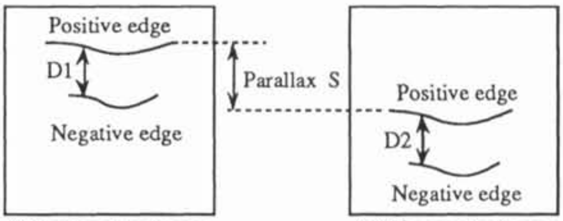


Fig. 4 Search sequences for white-line candidates



(a) Lower edge image (b) Upper edge image

Fig. 5 Principle of double image correspondence

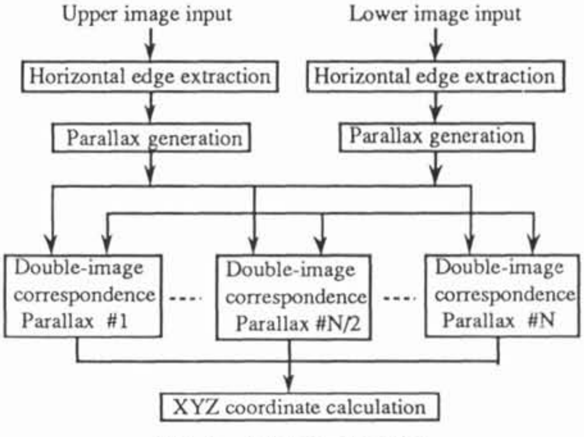

Upper image input
Lower image input
Horizontal edge extraction
Horizontal edge extraction
Parallax generation
Parallax generation
Double-image correspondence Parallax #1
Double-image correspondence Parallax #N/2
Double-image correspondence Parallax #N
XYZ coordinate calculation

Fig. 6 Obstacle detection

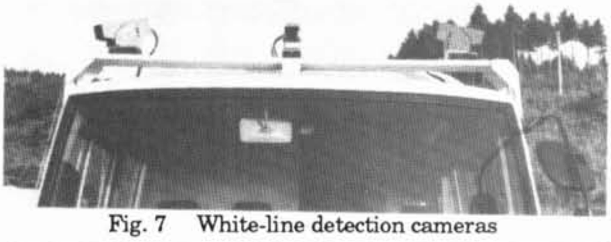
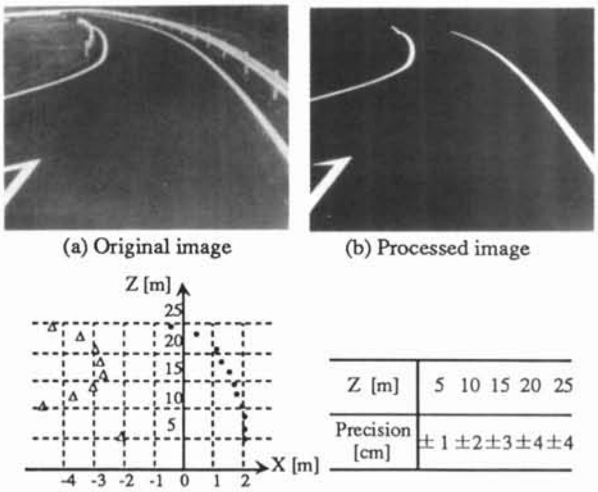


Fig. 7 White-line detection cameras



(a) Original image (b) Processed image

Z [m]	5	10	15	20	25
Precision [cm]	± 1	±2	±3	±4	±4

(c) Detection result (d) Detection precision

Fig. 8 Example of white-line detection

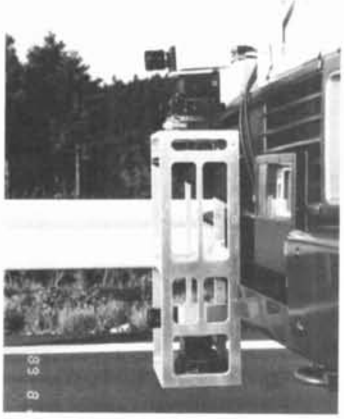


Fig. 9 Obstacle detection cameras

paired images in the upper and lower images do not correspond. If they are equal, the paired edges in the upper and lower images can be corresponded by shifting the lower image for the amount of parallax S. If the parallax can be extracted, it is possible to calculate depth information in the XYZ space from the camera location and angle and lens parameters. Figure 6 shows the obstacle detection flow which involves the following steps:

1. Horizontal edge extraction

Horizontal edge images are extracted using a spatial filter with a window of 3 x 3 pixels.

2. Parallax generation

An image on the xy plane is shifted in the direction of y one pixel at a time to generate N types of parallactic images.

3. Double-image correspondence

N types of parallactic images are compared and areas with corresponding edge pairs are replaced with parallax S. This means an area (obstacle line or plane) enclosed by positive and negative edges has the same parallax.

4. XYZ coordinate calculation

Once parallax S is determined, the depth to the obstacle in the XYZ space can be calculated from the distance between the two cameras and from lens parameters. Here, the resolution of digital images is R_x in the x direction and R_y in the y direction, the depth between the two TV cameras A, and the depth to the crosspoint of the visual axes of the two cameras B. Lens parameters are horizontal view angle φ_H and vertical view angle φ_V. The value of the X and Y coordinates of an obstacle on the XY plane is derived using equations 3 and 4:

$$Z = B \Big/ \{\frac{2S}{R_y} \tan(\frac{\varphi_H}{2}) + \frac{A}{B}\} \tag{3}$$

$$X = \frac{2}{R_y} Z(y - \frac{R_y}{2}) \tan(\frac{\varphi_V}{2}) \tag{4}$$

Last, depth and road-area data obtained by the white line detector are compared to determine the location of three-dimensional objects that may be obstacles, and three points (middle, leftmost, and rightmost) on the object closest to the vehicle are transferred to the vehicle controller.

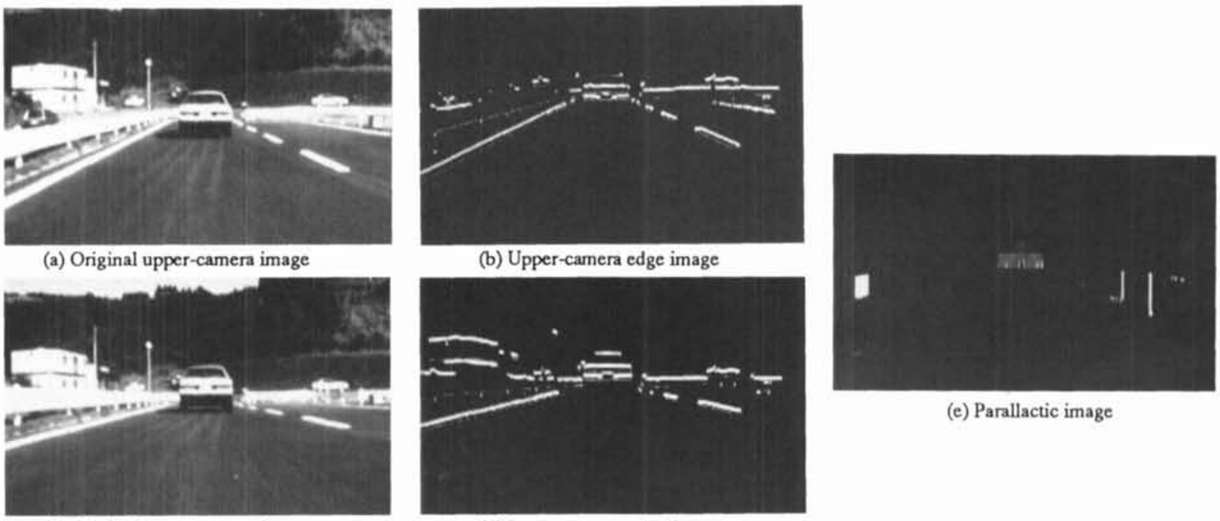
4. EXPERIMENTAL RESULTS

4.1 White-Line Detection

The three cameras used for white-line detection (Figure 7) are 2.43 meters above the ground with a depression angle of 71.7°, a vertical view angle of 42.8°, and a horizontal view angle of 55.5°. The visual axes of the three cameras cross on the ground about 5 meters in front of the vehicle. The switcher switches the three cameras to make the horizontal view about 120° wide. Figure 8 shows results: (a) Image input from the center camera, (b) white-line extraction results, (c) plotting of white-line location calculation results after conversion to XZ coordinate data at one-meter intervals, and (d) detection precision when image processing detection results and measured values are compared without the vehicle moving. Results showed that white lines on the road are detected very precisely.

4.2 Obstacle Detection

The lower of the two cameras used for obstacle detection (Figure 9) was 45 cm from the ground with a depression angle of 90° and the upper one was 92 cm from the ground with a depression angle of 87.3°. The distance between the two cameras was set to 47 cm because the average height of obstacles on roads is about 100 cm. Lens parameters are 60.8° for the vertical view angle and 44.5° for the horizontal view angle. Figure 10 shows an example: (a) Image input from the upper camera, (b) example of edge detection, (c)


(a) Original upper-camera image

(b) Upper-camera edge image

(e) Parallactic image

(c) Original lower-camera image

(d) Lower-camera edge image

Fig. 10 Example of obstacle detection

Table 1 Obstacle detection precision

Obstacle distance Z [m]		5	10	15	20	25
Z-axis precision		- 0.3 to 0.1	- 0.9 to 0.3	- 1.7 to 0.6	- 3.5 to 1.2	- 4.2 to 0.7
X-axis precision	X = 0 m	± 2	± 4	± 6	± 8	± 10
	X = ±2 m	±14	± 22	± 28	±43	±43
	X = ±4 m	-----	±40	± 50	±78	±77

image input from the lower camera, (d) example of edge detection, and (e) extracted parallactic image. Locations of the object were (X, Z) = (83 cm, 2013 cm) for the mid point, (-77 cm, 2152 cm) for the leftmost point, and (87 cm, 2152 cm) for the rightmost point. Table 1 compares obstacle detection results by image processing and for actually measured values. Results showed that detection precision drops if the obstacle moves away from or to the side of the vehicle. Because the precision of obstacle size and depth improves as the vehicle approaches the obstacle, no problems occur in autonomous operation.

4.3 Run Test

We operated the PVS (Figure 11) over a 110 x 70 m^2 test area having white lines 15 cm wide. The vehicle operated at 30 km/h on the straight stretches, 15 km/h on curves with a radius of 15 meters, and 5 km/h at intersections. The vehicle also operated at 60 km/h on a different long straight course. In the obstacle detection experiment, the vehicle automatically stopped or drove round parked cars on the course. These results confirm that the white-line and obstacle detection systems are effective.

5. CONCLUSION

We installed FIVIS/VIP on a vehicle and developed a visual information processing algorithm enabling it to operate flexibly even on curves or at intersections. We also developed a binocular 3D vision system so that the vehicle recognized an object 20 cm or more around as an obstacle at 1/30 second/frame and automatically avoided or stopped before hitting the obstacle. Experiments confirmed that the vehicle operated smoothly on curves and at intersections and achieved a maximum 60 km/h on the straightaway.

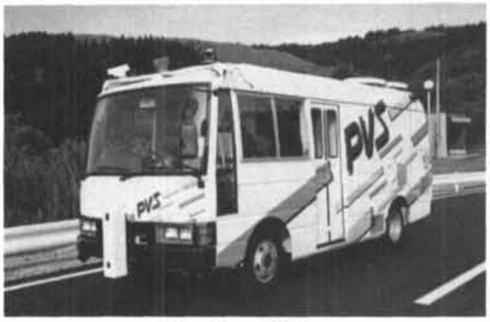


Fig. 11 PVS

REFERENCES

[1] S. Sasaki et al.: "IDATEN: A real-time image processing system with modifiable network (in Japanese),"*Information Processing Society of Japan*, CV 37-1 (1985).

[2] M. Komeichi et al.: "A color time-varying image processing system: 'color-IDATEN' (in Japanese)," *Information Processing Society of Japan*, CV 49-2 (1987).

[3] A. Waxman et al.: "A visual navigation system for autonomous land vehicles," *IEEE Journal of Robotics and Automation* , vol. RA-3, no. 2, pp. 124-142 (1987).

[4] C. Thorpe et al.: "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. on PAMI* , vol. 10, no. 3, pp. 362-373 (1988).

[5] E. D. Dickmanns: "Subject-object discrimination in 4D-dynamic scene interpretation for machine vision," *IEEE Proc. In Workshop on Visual Motion*, pp. 298-304 (1989).