A METHOD OF AUTONOMOUS VEHICLE GUIDANCE BY USING WHITE DASH-LINE RECOGNITION

Thawatchai Yiengwattananon and Kosin Chamnongthai
Faculty of Engineering, King Mongkut's Institute of Technology Thonburi (KMITT)
Bangkok, Thailand
E-mail : ikosthai@cc.kmitt.ac.th

ABSTRACT

This paper proposes a method of white dash-line recognition for autonomous vehicle guidance. In this method, vehicle is navigated according to white dash-line with finite length on flat floor. These dash-lines are sensed by a CCD camera mounted in front of the vehicle. The dash-line images are processed and performed pattern-matching process with a set of white dash-line patterns (64 patterns) in database. If the matching result shows that it is one of determined patterns, the system will control proper angle of wheel by radius of curve computation. Otherwise, the vehicle stops, because it is assumed that an obstacle exists on the path. In addition, the system also measures running distance. Since the dash-line has finite length, the running distance can be known by prediction. In this method, predicted dash-line position is used to compare with dash-line position on image. If the result of comparison is different, the system will calculate actual running distance. And the experiment with constructed vehicle demonstrates the possibility and efficiency of this method.

Index Terms- Autonomous vehicle, pattern matching, dash line, pattern recognition, mobile robot, vehicle guidance

1. INTRODUCTION

Nowadays, robot systems are used in industrial plant for production, assembly, inspection process and part transporting system. Especially in dangerous sites such as nuclear power plant, the replacement of human process is required.

For automatic transporting system, the autonomous vehicles need to move into determined direction with correct distance and skillfully as human. Among various approaches, the system using visual information for navigation is considered to be interesting, since three dimensional shape of obstacles can be recognized.

Autonomous vehicle using visual information can be classified into two groups according to running environment : outdoor and indoor. First group, outdoor (including both on-road and off-road terrain), since the vehicle has big size and needs high speed, for real time processing, most of them need to use high performance computers and various excellent sensors e.g. RGB video camera, laser range scanner, sonic imaging sensor, 2D radar, infrared, etc. The examples of those reports are as following. Moravec[1] uses a single camera to take nine pictures, spaced along a 50 cm. track, and pick out distinctive features in the images. These features were correlated between images and their three dimensional positions by using a stereo algorithm. Kuan[2] uses sonic imaging sensor for obstacle avoidance, infrared sensor for target detection and color camera for road following, by using color transformation, geometric reasoning, world knowledge and terrain database. Turk[3] uses RGB video camera with pan/tilt control and a laser range scanner for obstacle detection. Thorpe[4] presents two types of vision algorithms : color vision for road following, and 3D vision for obstacle detection and avoidance. Kenshiro[5] uses 2D radar and camera to develop the real time Hough transform module and improves the quadratic prediction model to the two-state prediction model for the trajectory control. Manigel[6] uses two cameras, one with a telephoto lens is for obstacle, and another one with a wide-angle lens is for lane detection. De Saint Vincent[7] uses stereo cameras and a laser range finder for robot position correction and tracking world features.

For the second group, indoor, since the path in factory and office is narrow, the vehicles are required to run not so fast and be constructed by compact size. And the practical vehicle needs simple hardware system, flexible software and reasonable cost-performance. The examples of studied reports in this group are as following. Inigo[8] uses edge detection, perspective inversion, and line fitting (via Hough transform) to find the path, and stereo vertical cameras, called "motion driven scene correlation", to detect obstacles. Ishikawa [9] uses white line for vehicle guidance. The vehicle system can select path and detect obstacles by white line patterns, but no distance measurement function.

The authors tried to develope an indoor-autonomous vehicle which consists of flexible system with simple sensor and running distance measurement function, and we propose a method of autonomous vehicle guidance by using white dash-line recognition in this paper. In this method, determined white dash-line patterns are used for pattern-matching process with input image. The obstacles are assumed to exist on the path when pattern-matching result reveals that there is no matched pattern. And because of finite length dash-line, the image of dash-line can be predicted. Then the running distance of vehicle can be known by the result of comparison between dash-line on input image and the predicted one. And the experiment with constructed vehicle demonstrates the possibility and efficiency of this method.

2. AUTONOMOUS VEHICLE SYSTEM

The system consists of hardware system and image processing system. The details are in following sections.

2.1 Hardware system

The system consists of devices as shown in fig.1.

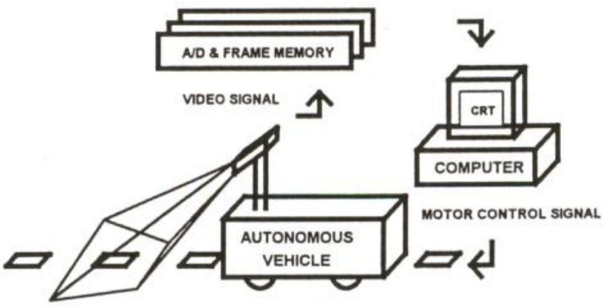


Fig.1 Hardware system

- CCD Camera, set in front of the vehicle to take dash-line image on the floor.

- A/D and Frame Memory Card, convert video (analog) signal to digital signal and store in image frame memory.

- Computer, used for image processing and calculation, and then commands to control vehicle.

- Vehicle, consists of three wheels that is driven by electric motor.

2.2 Image processing sequences

The image processing sequences are shown in fig.2. First, the system takes white dash-line and determines appropriate threshold value by histogram that shows number of pixels in each brightness level. This threshold value is used to distinguish white dash-line and floor. When the vehicle starts to run, the system takes image, then pattern-matching process is performs with a set of white dash-line patterns in database. If the matching result shows that it is one of determined pattern, the system will control wheel angle along dash-line in determined direction firstly and compute running distance. Otherwise, the vehicle stops, because it is assumed that an obstacle exists on the path. For confirming that it is not misunderstood by light condition, the system will take image to find threshold value and recognize white dash-line again. Then the system monitors running distance of the vehicle to get the goal. The details of white dash-line recognition, determination of wheel angle and running distance calculation are in section 3, 4, 5 respectively.

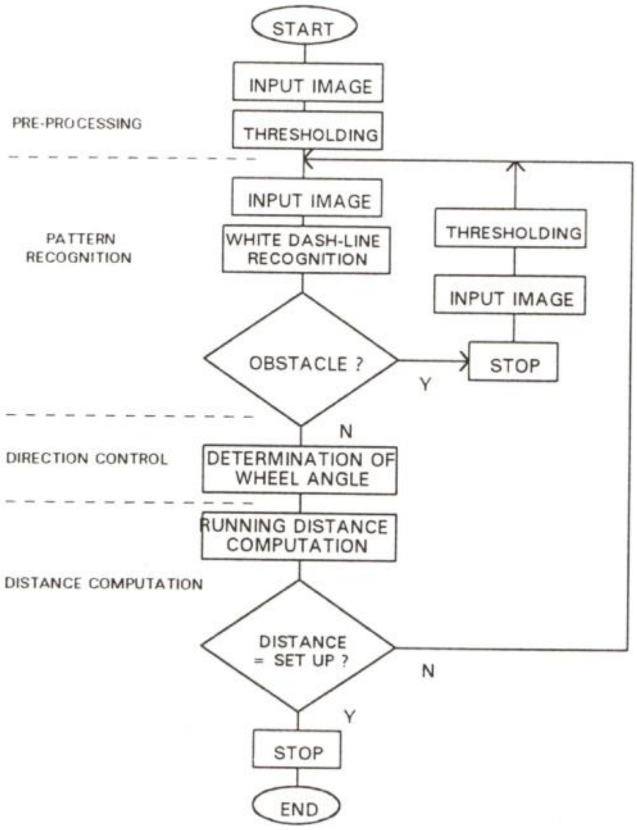
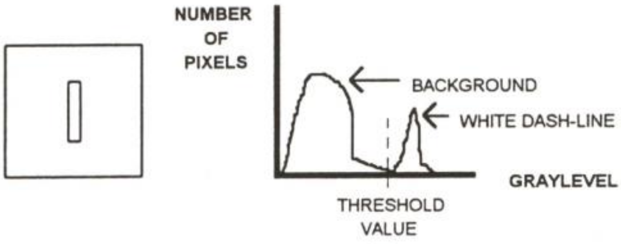


Fig.2 Image processing sequences

2.2.1 Threshold selection

The colors of floor and white dash-line are assumed to be different, so the histogram of the image can be shown as fig.3.



(a) (b)

Fig.3 (a) White dash-line image (b) Histogram

Since it is bimodal histogram, the threshold can be determined by finding graylevel between two peak that has minimum number of pixels. The threshold is used to separate white dash-line and floor into two levels : black (0) and white (1).

$$g(x,y) = 0 \text{ if } f(x,y) \leq \text{threshold (floor)}$$
$$= 1 \text{ if } f(x,y) < \text{threshold (white dash-line)}$$

where $g(x,y)$: binary image

$f(x,y)$: gray level of point (x,y)

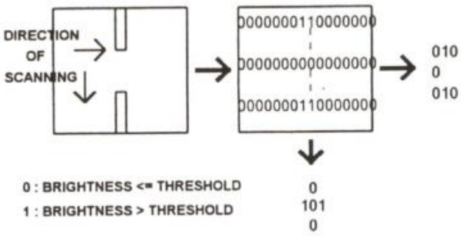


Fig.4 White dash-line recognition

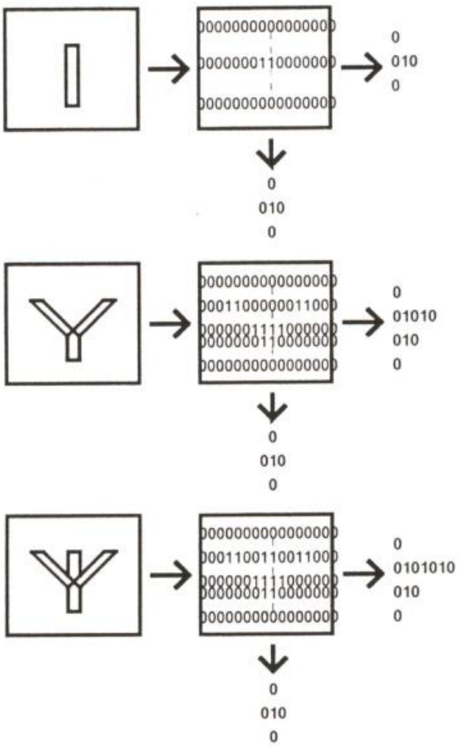


Fig.5 Examples of white dash-line pattern

2.2.2 White dash-line recognition

Pattern recognition of white dash-line is performed by scanning in horizontal and vertical direction to encode image pattern as shown in fig.4.

By encoding image patterns, the patterns of straight way, 2 ways branch and 3 ways branch can be determined as shown in fig.5. In reality, many patterns of dash-line image occur while the vehicle is running. The seen image depends on overlap ratio of successive image, height and tilt angle of camera as shown in fig.7(a).

3. DETERMINATION OF WHEEL ANGLE

After the system determined direction (e.g turn left at 2 way branch and turn right at 3 way branch), the system calculates proper angle of wheel by radius of curve computation as following steps. (see fig.6)

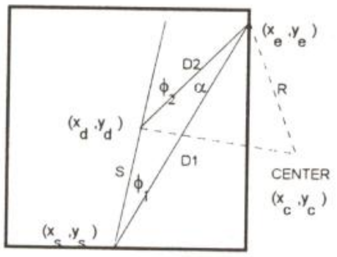


Fig.6 Determination of wheel angle

- selection three appropriate points along determined direction : (x_s,y_s), (x_d,y_d),(x_e,y_e)

- inverse projection transform all points from 2D image plane (x,y) into 3D world plane (X,Y,Z) to find actual distance by : (see appendix)

$$Z = Y\frac{[F\cos\theta + y\sin\theta]}{y\cos\theta - F\sin\theta} \qquad (3.1)$$

$$X = x\frac{[Z\cos\theta + Y\sin\theta]}{F} \qquad (3.2)$$

- finding center point of curve (x_c,y_c) by below equations.

$$-\frac{x_d - x_c}{y_d - y_c} = \frac{y_d - y_s}{x_c - x_s} \qquad (3.3)$$

$$(x_d - x_c)^2 + (y_d - y_c)^2 = (x_c - x_e)^2 + (y_c - y_e)^2$$

- finding radius of curve (R) by :

$$R = \sqrt{(x_c - x_e)^2 + (y_c - y_e)^2} \qquad (3.5)$$

or $$R = \sqrt{(x_c - x_d)^2 + (y_c - y_d)^2} \qquad (3.6)$$

(x_s,y_s) : start point before delay

(x_d,y_d) : delay point before turning

(x_e,y_e) : end point

(x_c,y_c) : center point of curve of radius R

S : delay distance before turning $[S = V.t_d]$

where V : vehicle velocity

t_d : delay time before turning

after delay, distance D_1 will be D_2 and angle ϕ_1 will be ϕ_2 as following.

$$D_2 = \sqrt{D_1^2 + S^2 - 2D_1 S\cos\phi_1} \qquad (3.7)$$

$$\phi_1 = \phi_1 + \alpha \qquad (3.8)$$

$$\text{where } \alpha = \cos^{-1}\left[\frac{S^2 + D_1^2 - D_2^2}{-2D_1D_2}\right]$$

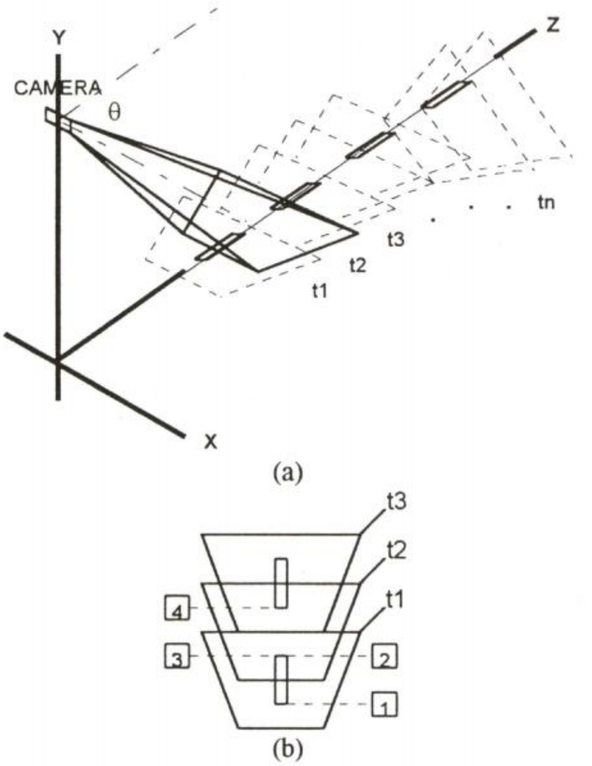
4. RUNNING DISTANCE COMPUTATION

Because of appropriate setting of tilt angle of camera and the length of dash-line, the number of dash-lines in one image is assumed not excess two dash-lines. The successive images will be overlaped while the vehicle is running. (fig.7(b)) The equations for determination tilt angle of camera and length of dash-line are as following.

$$\theta = TAN^{-1}\sqrt{\frac{y_{ub}^{2}VT - 2y_{ub}FY[1-K]}{VTF^{2} + 2y_{ub}FY[1-K]}} \quad (4.1)$$

$$L = \frac{VT}{2[1-K]} \quad (4.2)$$

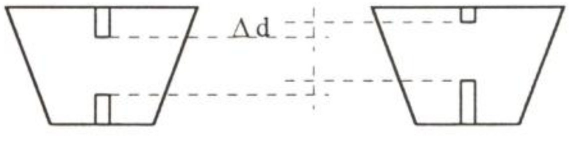
where y_{ub} : upper bound of image plane
F : focal length
Y : camera height
K : overlap ratio of successive image
V : vehicle velocity
T : process time per image



(a)

(b)

Fig.7 (a) Seen image while vehicle is running
(b) Overlap of successive images



(a) Pedicted image (b) Seen image

Fig. 8 Difference of dash-line position between predicted image and seen image

In fig.7(b), shows overlap of successive image with K = 0.5. It is found that position 2 on t_1 image and position 3 on t_2 image are the same position in world plane. Since the dash-line has finite length, the running distance of vehicle can be known by the result of comparison between predicted image by

$$y_{tn+1} = y_{tn} -(1-K)(y_{ub}-y_{lb}) \quad (4.3)$$

and dash-line position of seen image. (fig.8) If the result of comparison is different, dash-line position on last seen image and present seen image will be transformed from 2D image plane into 3D world plane, and subtract to find actual running distance.

5. EXPERIMENTAL RESULTS

Table 1 Condition of experiment.

VEHICLE	size wheel velocity	60x40 cm. 3 0.3 m./sec.
DASH-LINE	color width length interval	white 2 cm. 60 cm. 60 cm.
CCD CAMERA	height focal length tilt angle	40 cm. 1200 pixels 25°
IMAGE PROCESSING	image size graylevel CPU process time	512X512 pixels 256 levels Intel 486DX2-66 ≈ 0.6 sec./frame

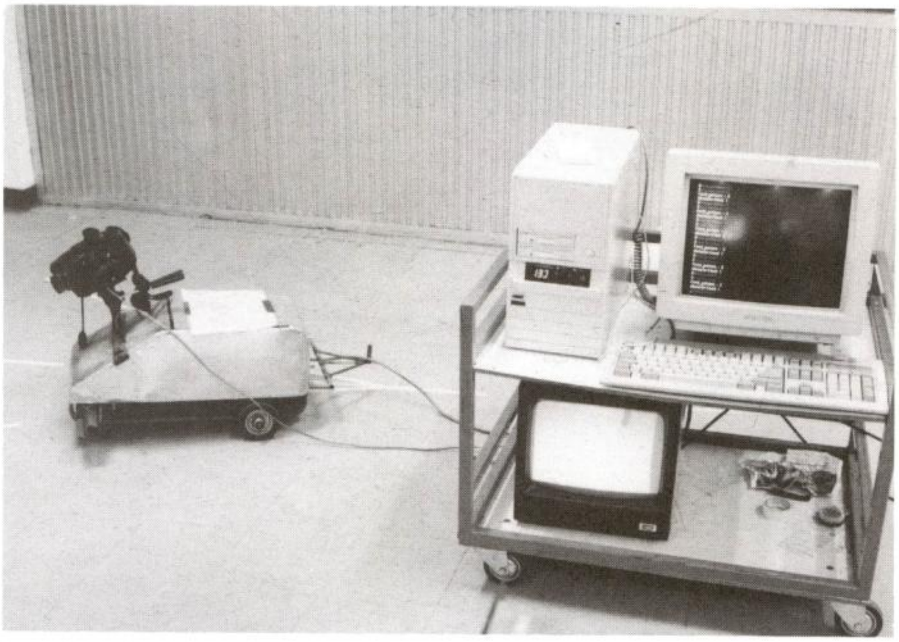


Fig. 9 Constructed vehicle for experiment

The experiment was done under condition as shown in table 1, to demonstrate possibility and efficiency of this method with constructed vehicle. (fig.9) For the experiment by dash-line patterns (e.g. straight, two-way (radius 1 m.) and three way branch (radius 2 m.)), the system can recognize various patterns, and the vehicle can run along determined direction and stop when obstacle exists on the path. Because of shadow problem, the system misunderstanding dash-line pattern as obstacle, so the vehicle is stopped. In fig.10, shows running test with constructed vehicle.

Graph in fig.11, shows error of longest (Zmax) and nearest (Zmin) distance that camera sees at various height and tilt angle of camera with 11 mm. focal length. In both cases, errors are below 7.1% and 14.4% respectively when tilt angle is 25-65 deg.. Trend of errors will be least at tilt angle 45 deg.. The condition of experiment, height 40 cm. and tilt angle of camera 25 deg., errors are 3.717% and 10.836% respectively. Graph in fig.12, shows error of running distance. The error is between 2.917%-15% within 5 m. distance. It is considered that the error occurs by projection transformation, 3D world plane (X,Y,Z) into 2D image plane (x,y) and slip of vehicle when it stops. But it decreases up to the running distance, because of no accumulated error. From experiment, some problems as following are needed to be solved in the future.

1. Shadow problem.
2. Turning problem.
3. Higher speed processing.
4. Obstacle avoidance without stopping vehicle. etc.

6. CONCLUSION

This paper proposes a method of guidance and distance measurement by white dash-line recognition with a CCD camera. The dash-line images are processed and performed pattern-matching process with a set of white dash-line patterns (64 patterns) in database. If the matching result shows that it is one of determined patterns, the vehicle will be navigated according to white dash-line.

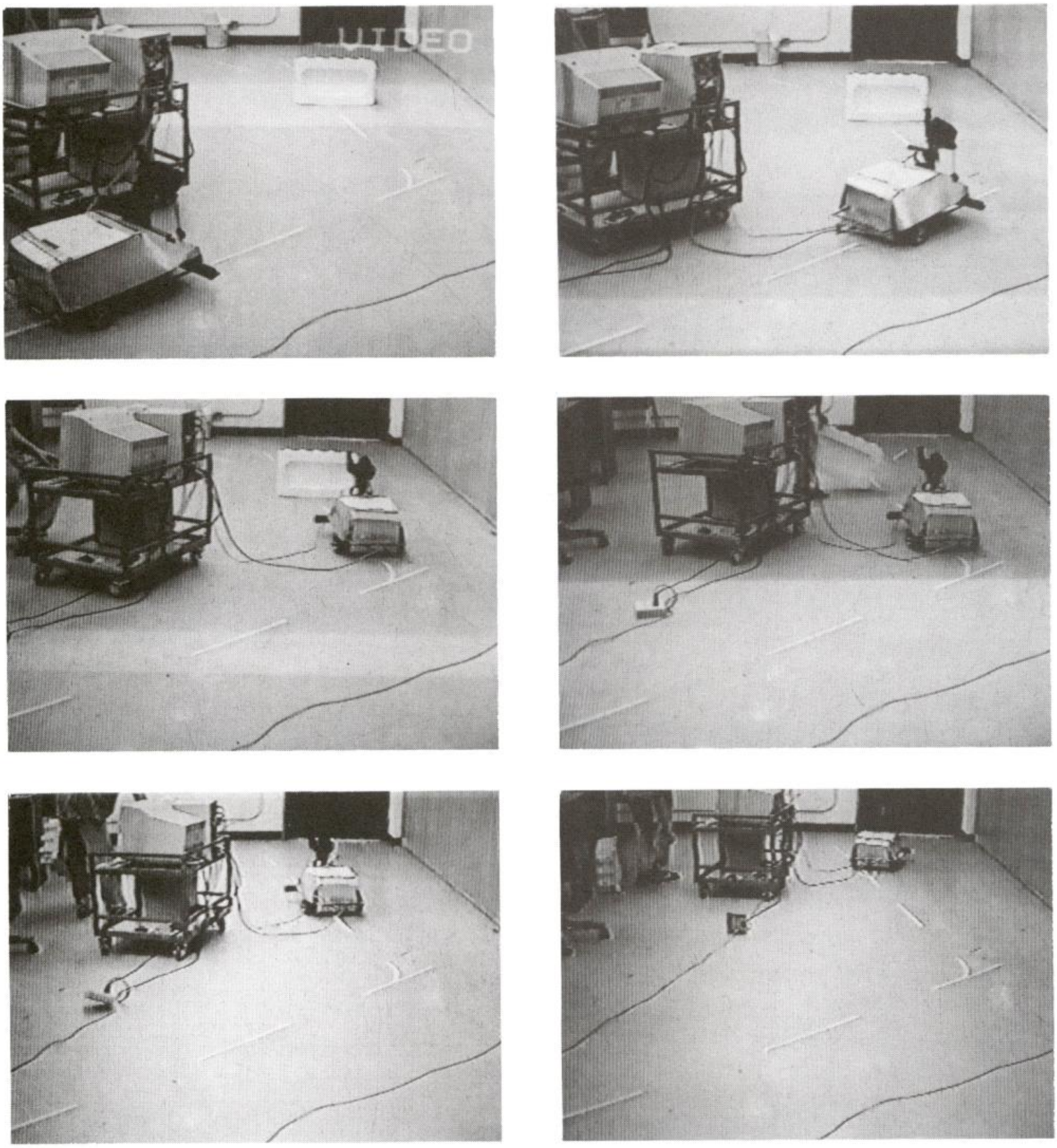


Fig.10 Running test with constructed vehicle

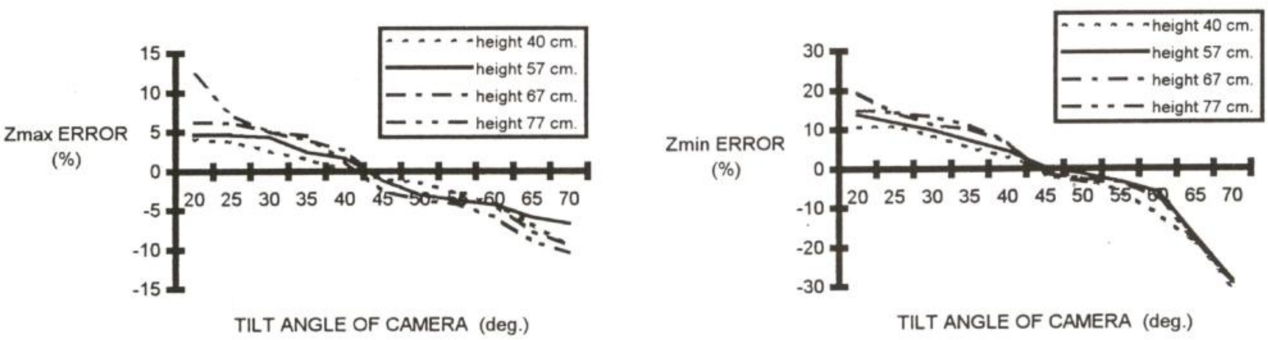


Fig.11 Error of longest (Zmax) and nearest (Zmin) distance that camera sees

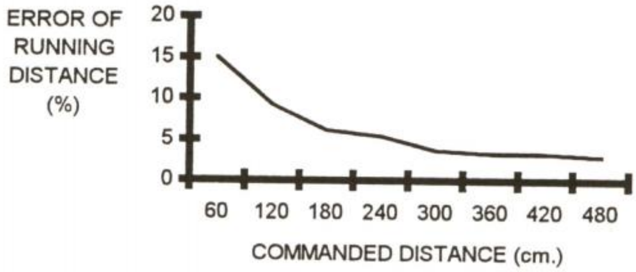


Fig.12 Error of running distance

Otherwise, the vehicle stops, because it is assumed that an obstacle exists on the path. Since the dash-line has finite length, the running distance can be known by prediction. In this method, predicted dash-line position is used to compare with dash-line position on image. If the result of comparison is different, the system will calculate actual running distance. The experiments by constructed vehicle reveal the possibility to utilize in factory and office.

APPENDIX

(a) Tilt angle of camera

Projection transformation, 3D world plane (X,Y,Z) into 2D image plane (x,y) is defined by [10] :

$$x = \frac{FX}{ZCOS\theta - YSIN\theta} \qquad (1)$$

$$y = \frac{F[YCOS\theta + ZSIN\theta]}{ZCOS\theta - YSIN\theta} \qquad (2)$$

from (2)

$$y_{ub} = \frac{F[YCOS\theta + Z_{max}SIN\theta]}{Z_{max}COS\theta - YSIN\theta}$$

$$\therefore Z_{max} = \frac{Y[FCOS\theta + y_{ub}SIN\theta]}{y_{ub}COS\theta - SIN\theta} \qquad (3)$$

similarlity

$$\therefore Z_{min} = \frac{Y[FCOS\theta + y_{lb}SIN\theta]}{y_{ub}COS\theta - SIN\theta} \qquad (4)$$

Image will overlap if $Zmin_{tn+1} < Zmax_{tn}$ so

$$Zmin_{tn+1} = Zmax_{tn} - K[Zmax_{tn} - Zmin_{tn}] \qquad (5)$$

where K : overlap ratio of successive image (not excess 1)

$$\text{but } Zmin_{tn+1} = Zmin_{tn} + VT \qquad (6)$$

where V : vehicle velocity

T : process time per frame

substitute (6) into (5)

$$\therefore Zmin_{tn} + VT = Zmax_{tn} - K[Zmax_{tn} - Zmin_{tn}]$$

$$VT = [Zmax - Zmin][1-K] \qquad (7)$$

replacing Zmax and Zmin in (7)

$$\therefore \frac{VT}{Y[1-K]} = \frac{[F + y_{ub}TAN\theta]}{[y_{ub} - FTAN\theta]} - \frac{[F + y_{lb}TAN\theta]}{[y_{lb} - FTAN\theta]}$$

it $y_{lb} = -y_{ub}$

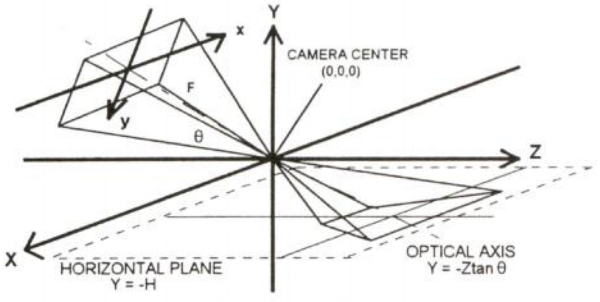
$$\therefore \frac{VT}{Y[1-K]}[(FTAN\theta)^2 - y_{ub}^2] = -2y_{ub}F[1 + TAN^2\theta]$$

$$[\frac{VTF^2}{Y[1-K]} + 2y_{ub}F]TAN^2\theta = \frac{y_{ub}^2VT}{Y[1-K]} - 2y_{ub}F$$

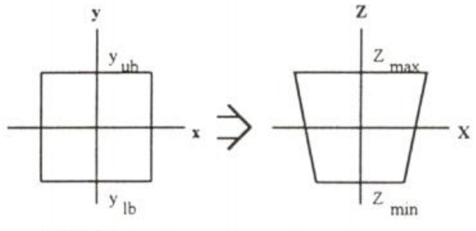
$$\therefore TAN^2\theta = \frac{[\frac{y_{ub}^2VT}{Y[1-K]} - 2y_{ub}F]}{[\frac{VTF^2}{Y[1-K]} + 2y_{ub}F]}$$

$$= \frac{y_{ub}^2VT - 2y_{ub}FY[1-K]}{VTF^2 + 2y_{ub}FY[1-K]}$$

$$\therefore \theta = TAN^{-1}\sqrt{\frac{y_{ub}^2VT - 2y_{ub}FY[1-K]}{VTF^2 + 2y_{ub}FY[1-K]}} \qquad (8)$$



3D COORDINATE SYSTEM FOR REAL WORLD



(b) Dash-line length

If we define dash-line number in image not excess 2, dash-line length(L) and interval(D) have equal length so

$$Zmax - Zmin = L + D = 2L$$

from (7) $$Zmax - Zmin = \frac{VT}{[1-K]}$$

$$\therefore L = \frac{VT}{2[1-K]} \quad (9)$$

REFERENCES

[1] H.P. Moravec, "The Standford Cart and the CMU Rover," *Proc. IEEE,* Vol.71, No. 7, pp.872-884, July 1983.

[2] Darwin Kuan, Gary Phipps and A. Chuan Hsueh, "Autonomous Robotic Vehicle Road Following," *IEEE Trans. on Patt. Anal. and Mach. Intell.,* Vol.10, No.5, Sept. 1988.

[3] Matthew A. Turk, David G. Morgenthaler, Keith D. Gremban and Martin Marra, "VITS-A Vision System for Autonomous Land Vehicle Navigation," *IEEE Trans. on Patt. Anal. and Mach. Intell.,* Vol.10, No.3, May 1988.

[4] Charles Thorpe, Martial H. Hebert, Takeo Kanade and Steven A. Shafer, "Vision and Navigation for the Carnegie-Mellon Navlab," *IEEE Trans. on Patt. Anal. and Mach. Intell.,* Vol.10, No.3, May 1988.

[5] H. Kenshiro, N. Shigeto, T. Saito, S. Ishida, Y. Okada and N. Oono, "An Image Processing Architecture and a Motion Control Method for an Autonomous Vehicle," *Proc. of The Intelligent Vehicle'92 Symposium,* pp.213-218.

[6] J. Manigel and W. Leonhard, "Vehicle Control by Computer Vision," *IEEE Trans. on Industrial Electronics,* Vol.39, No.3, June 1992.

[7] A. R. de Saint Vincent, "a 3D Perception System for The Mobile Robot Hilare," *in Proc. IEEE Int. Conf. Robotics and Automation,* pp.1105-1111, April 1986.

[8] R. M. Inigo, E. S. McVey, B. J. Berger and M. J. Mirtz, "Machine Vision Applied to Vehicle Guidance," *IEEE Trans. on Patt. Anal. and Mach. Intell.,* Vol. PAMI-6, No.6, Nov. 1984.

[9] S. Ishikawa et.al., "A Method of Image Guided Vehicle Using White Line Recognition," *in Proc. IEEE Computer Vision and Pattern Recognition,* pp.47-53, June 1986.

[10] W. Sohn and N. D. Kehtarnavaz, "Analysis of Camera Movement Errors in Vision-based Vehicle Tracking," *IEEE Trans. on Patt. Anal. and Mach. Intell.,*Vol.17, No.1, Jan. 1995.