IJCTA, 6(1) June 2013, pp. 47-58

Computer Vision based Autopilot System for an Intelligent Robot

Yuan-Wei Tseng*, Chang-Meng Li* and Guo-Cin Wang*

ABSTRACT

The purpose of this paper is to design an integrated program for an intelligent wheel robot, which can recognize and follow a predefined forward sign while automatically bypassing any encountered obstacle. By distributing those forward signs, the path of the robot is determined. With this concept, an image based autopilot system with immunity against electromagnetic interference is constructed. Harr-like feature training algorithm in OpenCV has been applied so that the robot can quick detect the forward sign. The rotation of the robot for automatic target detecting is achieved by fuzzy control. The experimental results showed that the robot could successfully detect forward sign and response properly. Consequently, the proposed design approach is proven to be workable. Simply redistributing the recognizable signs by the robot, a new path for robot is constructed. Therefore, it has great flexibility for applications.

INTRODUCTION

Self-propelled platforms that travel on predefined paths play many important roles in automation such as automatic manufacture lines, warehouse storages and public transportation. Those platforms usually travel on rails or are guided by underground magnetic devices [1]. However, they have several major drawbacks. For example, rails usually become obstacles for other vehicles and pedestrians, consequently, cause safety problems. For the platforms that guided by underground magnetic devices, they are not suitable to carry items that are sensitive to magnetic fields. Furthermore, changing the layouts of both rails and underground magnetic devices are always time consuming and costly. To overcome those drawbacks, computer vision ability is added to self-propelled platforms so that they can follow the lines that are painted on the surface of paths to become line tracking robots [2]. However, the lines on the paths unavoidably suffer from erosions of weather and heavy traffics. Therefore, better guiding method for intelligent robot with computer vision ability is desired.

In this paper, equipped with camera, an intelligent wheel robot which can recognize and follow a particular forward sign while automatically bypassing any encountered obstacle is designed. By distributing the forward signs, the path of the robot is determined. With this concept, an image based autopilot system with extremely low maintenance and immunity against electromagnetic interference is constructed. Open Source Computer Vision Library (OpenCV) [3] which is a library of programming functions for real time computer vision provided by Intel is adopted in developing the integrated programs for the intelligent robot. Haar Classifiers in OpenCV, which apply AdaBoost classifier cascades [4-5] based on Haar-like features [4-5] rather than pixels can rapidly detect any object have been applied to enable the robot to quick detect the forward sign. The function of the forward sign is actually motion order to robot such as "rotating to aim the forward sign and approaching to it". When the forward sign is detected by the robot, the robot will response properly. In the end of approaching to the forward sign, the robot will

* Department of Electrical Engineering, I-Shou University, Dashu District, Kaohsiung City 84001, Taiwan, R. O. C., *E-mail: yuanwei@isu.edu.tw*

bypass the forward sign when their distance is close enough and will start to search for another forward sign.

In control design, the performance of explicit model based control designs usually suffers from inaccurate modeling, nonlinearity, and system uncertainty. To put all of those factors into the consideration in designing model based control system usually results in lots of mathematics overhead that only can be handled by experts. On the contrary, fuzzy control [6-7] is a specific type of knowledge-based control. It requires neither explicit system model nor advanced mathematics. The control actions corresponding to particular conditions of the system are described in terms of fuzzy if-then rules. Fuzzy sets [8] are used to define the meaning of qualitative values of the controller inputs and outputs such as small error, large control action. Fuzzy logic can capture the continuous nature of human decision processes and as such is a definite improvement over methods based on binary logic. Therefore, fuzzy control provides better feasibility, robustness and fault tolerance for designing the motion controller. When the forward sign is detected, the robot will automatically rotate to keep the forward sign in the center of the captured image. Fuzzy control is used to control such rotation because its accurate system model is not available.

In this research, for the intelligent robot with computer vision and fuzzy motion controller, by properly distribute those forward signs, the robot paths can be easily established and modified. Therefore, people have lots of flexibility while only have very limited need of maintenance to apply the intelligent robot to automation applications.

SYSTEM OPERATION OVERVIEW

The control system of the robot is integrated with programs of image process and fuzzy motion control process. The image process program compares the webcam image inputs with the forward sign' features from training program to detect the forward sign. Once a forward sign is detected by the image processing program, fuzzy motion control program will rotate the robot to aim the forward sign and then move toward it. When an obstacle is detected by the ultrasonic sensors, fuzzy motion control program will launch a bypass process.

The details of system operation are shown in Fig. 1. When the robot is power on, it starts to search the forward sign by rotating itself and moving forward at preprogrammed ranges. If an obstacle is detected by robot's ultrasonic sensors, the robot will bypass the obstacle and resume the search of the forward sign. When a forward sign is detected, the robot will rotate itself to aim the forward sign and the rotation is controlled by the fuzzy motion control program. Then the robot will approach to the sign. When the robot and the forward sign are close enough so that the ultrasonic sensors detect the sign as an obstacle, the robot will also launch the bypass maneuver. Every time after the robot finishing bypass maneuver, it starts to search for the forward sign again.

IMAGE PROCESS BASED ON HAAR-LIKE FEATURES

In this paper, Haar-like features based tracking is applied so that the intelligent robot can quickly identify and track its object, the forward sign. In 2001, Paul Viola and Michael Jones proposed integral image [4]. Based on integral images, a fast face detection system that utilized AdaBoost and Cascade algorithms [4] was proven to be very effective. Rainer Lienhart *et al.* [5] further proposed an extended set of Haar-like features of four edge features, eight line features, and two center-surround features as shown in Fig. 2. The rectangle features are relative to the enclosing detection window. The sum of the pixels which lie within the white rectangles is subtracted from the sum of pixels in the black rectangles.

Basically, the integral image at location (x, y) contains the sum of the pixels above and to the left of (x, y). The non-rotated integral image is defined as follows.

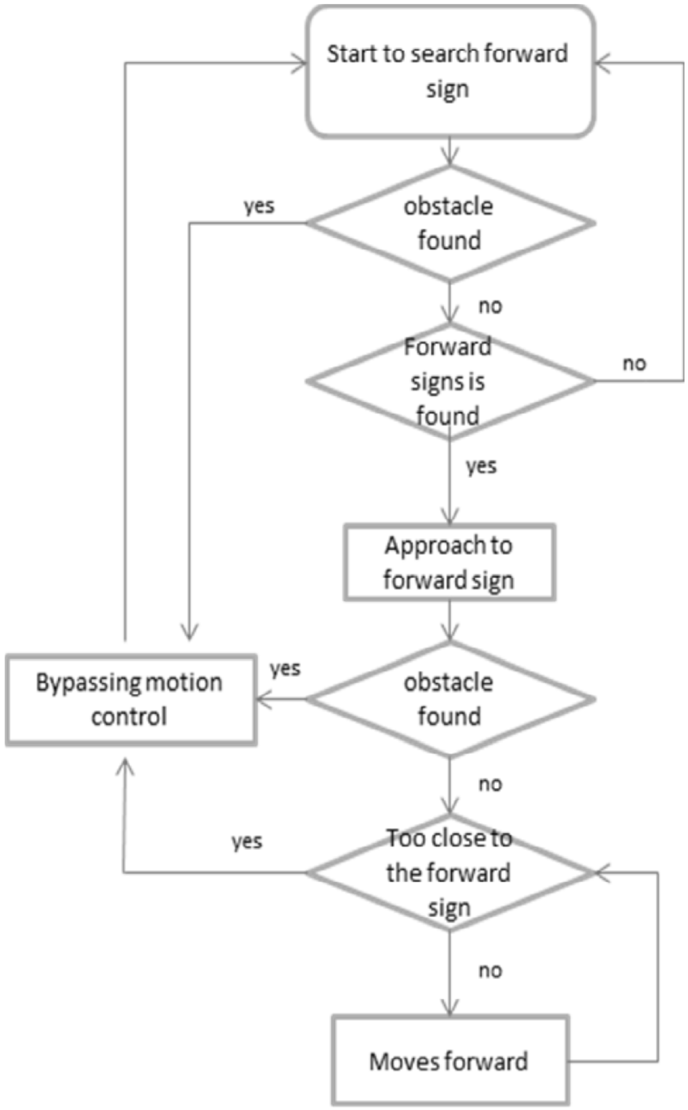

Figure 1: System Operation Flow

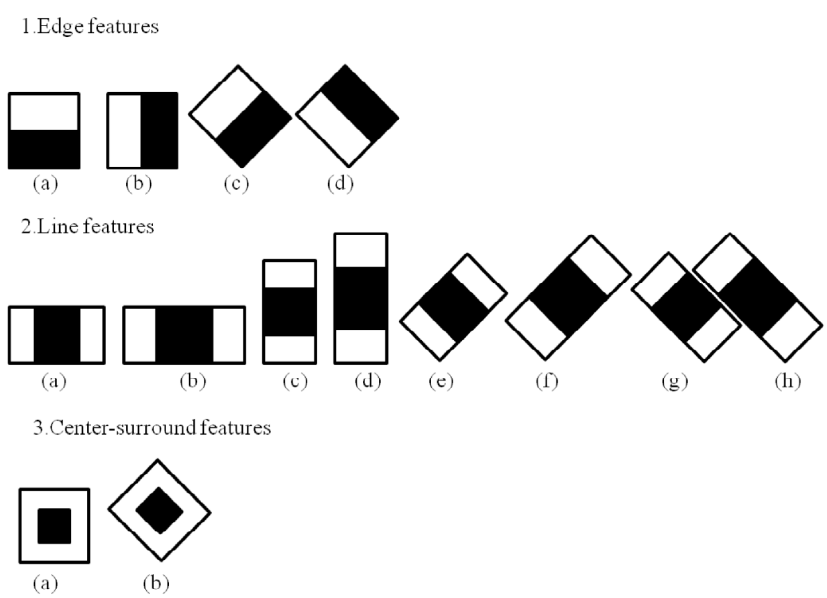

Figure 2. Extended Set of Haar-like Features [5]

$$I_i(x,y)=\sum_{x<W,y<H} I(x,y)$$
$$I_i(x,y)=I_i(x,y-1)+I_i(x-1,y)-I_i(x-1,y-1)+I(x,y) \qquad (1)$$
$$I_i(-1,y)=I_i(x,-1)=0$$

where $I(x, y)$ is the grey image of pixel (x, y), $I_i(x, y)$ is integral image of (x, y), w is the width of image and h is the width of image. Once the integral images are available, it shows that any rectangular sum of pixels can be computed in four array references as follows.

$$\mathrm{Re}\,ctSum(x,y,w,h)=I_i(x-1,y-1)+I_i(x+w-1,y+h-1) -I_i(x-1,y+h-1)-I_i(x+w-1,y-1) \qquad (2)$$

For 45° rotated rectangles in Fig. 2, the auxiliary image is the *rotated summed area table RSAT (x, y)* in [5]. Haar views the integral image of every pixel as an independent variable. Haar features are found by performing addition and subtraction between integral images. The larger the integral image value after addition is, the more important the integral image value is. Integral images of large value are more easily visualized. Therefore, the integral images after additions are belong to low frequency parts while the integral images after subtractions are belong to high frequency parts which emphasize the edges in image. Addition and subtraction between integral images can be performed horizontally and vertically to extract the image features. Second order examples of addition and subtraction between integral images in horizontal and vertical directions are as shown in Fig. 3 and Fig. 4, respectively. Here, L stands for low frequency while H stands for high frequency.

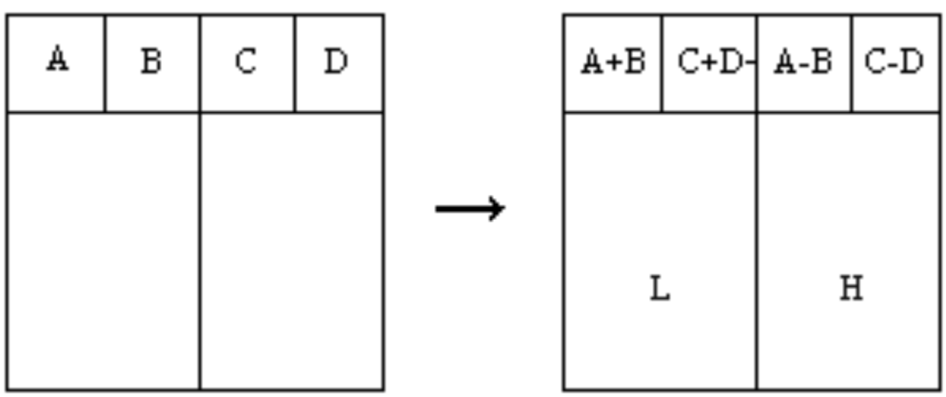

Figure 3: Horizontal Addition and Subtraction between Integral Images

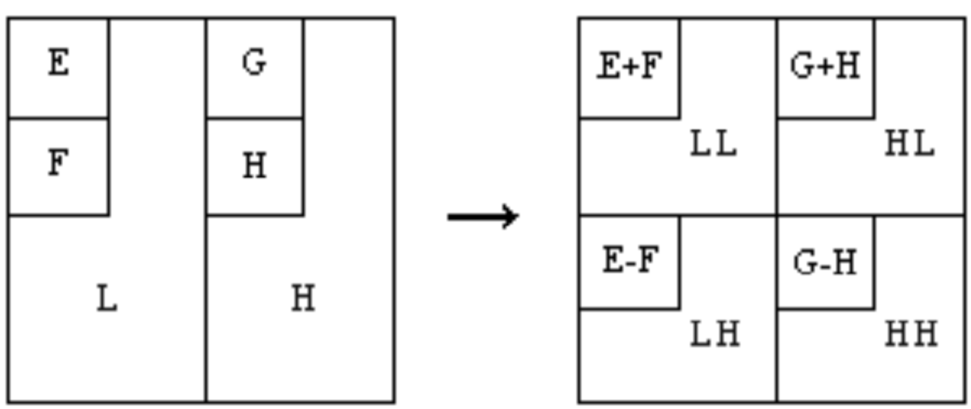

Figure 4: Vertical Addition and Subtraction between Integral Images

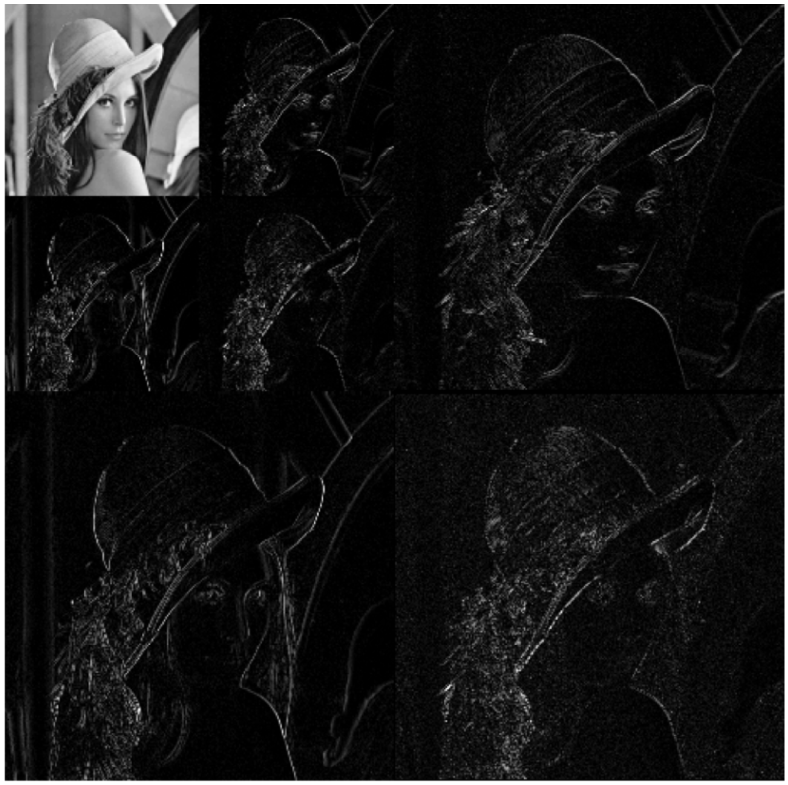

Figure 5: A Second order Example of Haar Feature Extractions [9]

A second order example of Haar feature extractions [9] is given in Fig. 5 for visualizing the effectiveness. Comparing Fig. 4 with Fig. 5, one can find that LL part is a quarter scale of the original image because it preserves the important extracted features while HL and LH parts are edges of image and the HH part collects corners of image. When higher order of Haar feature extractions is performed, the low frequency image on the top left corner will be further scaled down and preserves fewer but more important features of image. Once the image features are extracted, they can be applied to build up the classifiers for detecting the objects of interest.

Viola and Jones use AdaBoosting [4] as their basic classifier, which combines a collection of cascade weak classification functions to form a stronger classifier that often outperforms most ‘monolithic’ strong classifiers such as support vector machines [10] and Neural Networks. A weak classifier is only required to be better than chance, and thus can be very simple and computationally inexpensive. The image blocks are fed through a series of cascade weak classifiers. If it is not matched with any weak classifier, it is immediately categorized as a non-feature block. Only the image blocks that pass all weak classifiers are the blocks with features.

OpenCV provides *createsmaple* and *haartraining* for training classifier. The program *createsmaple.exe* is first applied to form training vector from different folders that contain positive samples of the forward sign specified in Fig. 7 and negative samples of non-objects. The created training vector in .dat file is then imported into program *haartraining.exe* for training a classifier cascade on a training set. After running *haartraining.exe,* a classifier description in an xml that can be deployed or tested is created. The complete flow is given in Fig. 6.

We must select the number of training stages when running *haartraining.exe*. In experiments, when the training stages were too few, we found that the robot caught not only the object, the forward signs but also other non-objects. On the other hand, when the training stages were too many, we found that the robot somehow easily lost the object when the object was at slightly different angles.

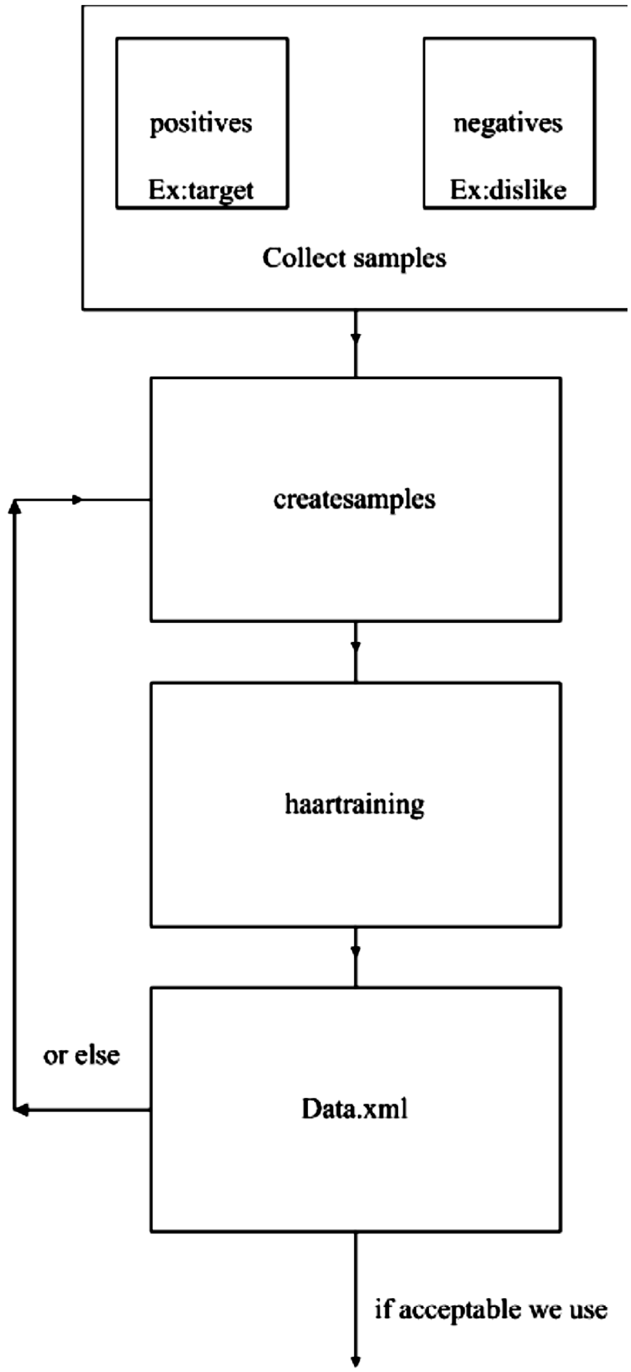

Figure 6: Flow of Training Classifier

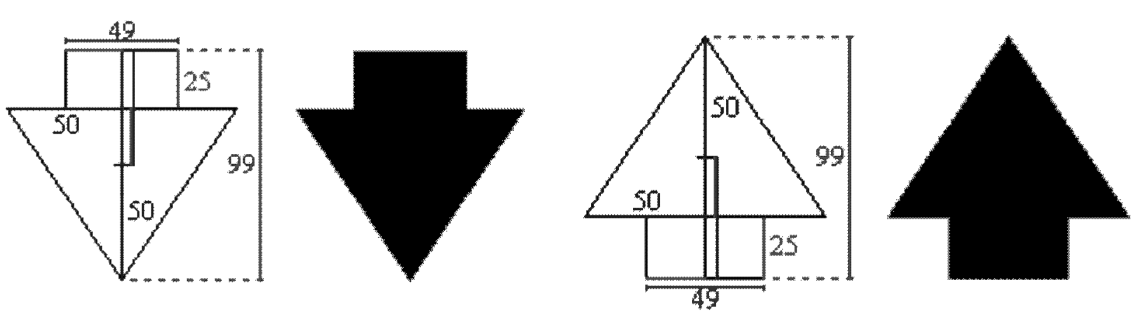

Figure 7: Positive Samples of Forward Sign

After trial and error, we found that for our application, the parameter settings of running *createsmaple* and *haartraining* to get best results is given in Fig. 8 and Fig. 9, respectively. The setting of training stages is 17 for running *haartraining*.

```
C:\OpenCV2.1\bin\img1>opencv_createsamples.exe -info positives/positives.txt -ve
c data/positives.vec -num 2112 -w 25 -h 25
Info file name: positives/positives.txt
Img file name: (NULL)
Vec file name: data/positives.vec
BG  file name: (NULL)
Num: 2112
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 25
Height: 25
Create training samples from images collection...
Done. Created 2112 samples
```

Figure 8: Parameter Settings of Running *createsmaple* to Get Best Results

```
C:\OpenCV2.1\bin\img1>opencv_haartraining.exe -data data/data -vec data/positive
s.vec -bg negatives/negatives.txt -npos 2112 -nneg 100 -w 25 -h 25 -nstages 17 -
nsplits 2 -mode ALL
Data dir name: data/data
Vec file name: data/positives.vec
BG  file name: negatives/negatives.txt, is a vecfile: no
Num pos: 2112
Num neg: 100
Num stages: 17
Num splits: 2 (tree as weak classifier)
Mem: 200 MB
Symmetric: TRUE
Min hit rate: 0.995000
Max false alarm rate: 0.500000
Weight trimming: 0.950000
Equal weights: FALSE
Mode: ALL
Width: 25
Height: 25
Applied boosting algorithm: GAB
Error (valid only for Discrete and Real AdaBoost): misclass
Max number of splits in tree cascade: 0
Min number of positive samples per cluster: 500
Required leaf false alarm rate: 7.62939e-006
```

Figure 9: Parameter Settings of Running *harrtraining* to Get Best Results

Applying the trained classifier in image process, the robot could always detect the forward sign in different environments as shown in Fig. 10.

FUZZY MOTION CONTROL DESIGN OF INTELLIGENT ROBOT

Once a forward sign is detected by image processing program, fuzzy motion control program will rotate the robot to aim the forward sign and then move toward it. When an obstacle is detected by the ultrasonic sensors, fuzzy motion control program will launch a bypass process.

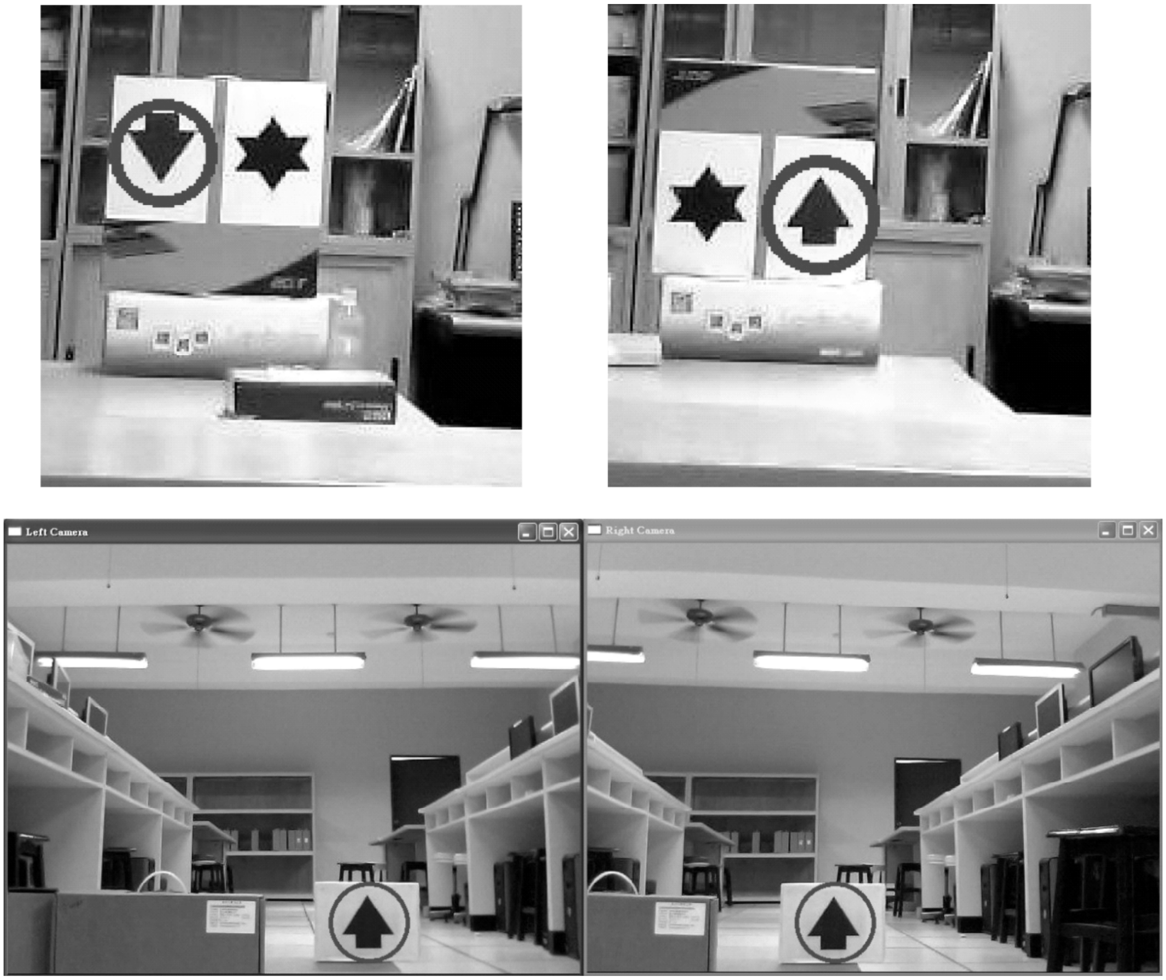

Figure 10: Forward Signs Detected in Different Environments

As shown in Fig.11, the structure of fuzzy controller consists of Fuzzifier that converts the system sensors' crisp measurement input to a linguistic variable using the membership functions stored in the Fuzzy rule base, Fuzzy inference engine that using If-Then type fuzzy rules converts the fuzzy input to the fuzzy output, Defuzzifier that converts the fuzzy output of the inference engine to crisp signal to drive system actuator using membership functions analogous to the ones used by the Fuzzifier.

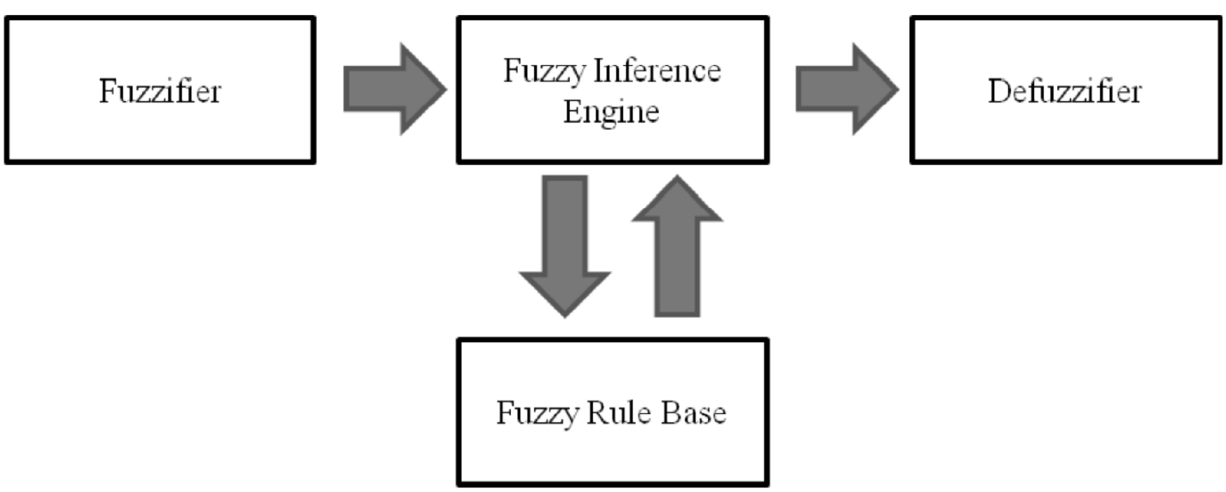

Figure 11: Structure of a Fuzzy Controller

In this research, image process extracts the features of images obtained from webcams for the forward sign matching. Once the forward sign is located, the forward sign's the center of mass in horizontal coordinate is calculated. Based on the displacement of center of mass of the forward sign in consecutive images, the robot is rotated to aim the object and approach to the forward sign by motion commands generated from fuzzy motion control process. For this application, webcam has *320×240* resolution. Therefore, the horizontal axis of pixel coordinates is from left to right ranging from 0 to 320. In this research, Triangular Membership Function shown in Fig. 12 is selected for the fuzzy motion control because it results in faster calculation speed and good enough performance that is comparable of using Gaussian Membership Function [7].

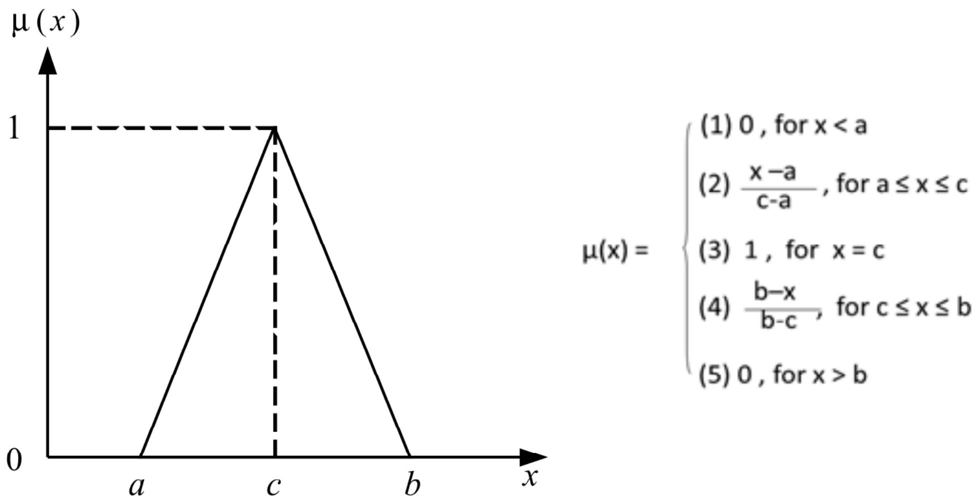

Figure 12: Triangular Membership Function

Horizontal axis is divided into five fuzzified locations and the corresponding membership functions based on the center of mass of the forward sign are as shown in Fig. 13. Please note that the triangular membership function in the middle is narrower than the others for better fine-tune results of final aiming stage. The current position of the center of the mass is categorized into five fuzzified locations as shown in Fig. 13. On the other hand, the displacement of center of the mass of the forward sign in consecutive images is categorized into three fuzzified displacements as shown in Fig. 14.

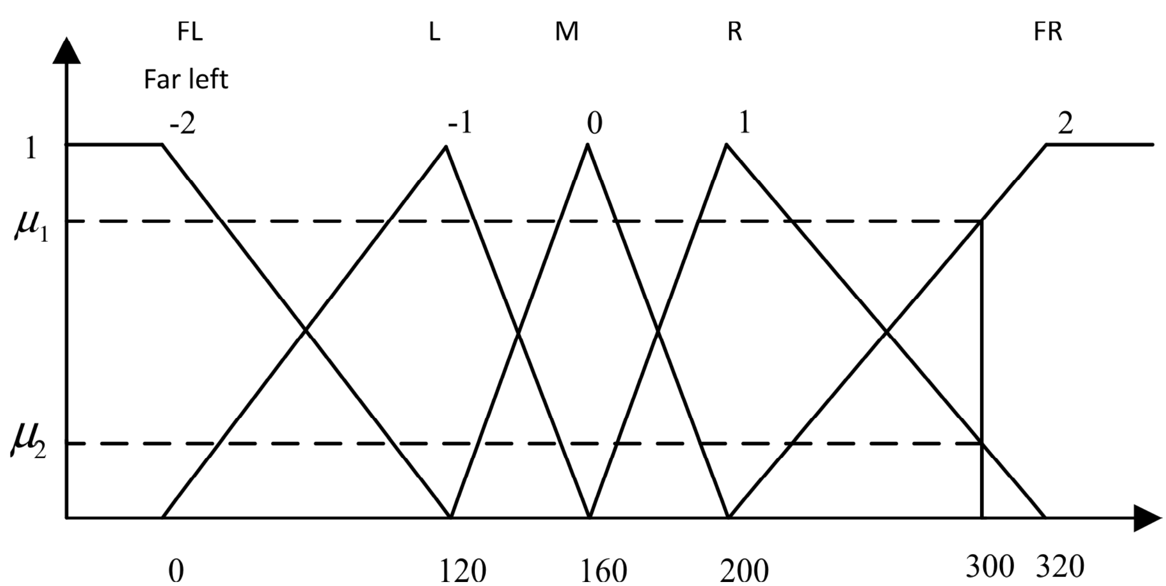

Figure 13: Fuzzified Locations of Center of Mass

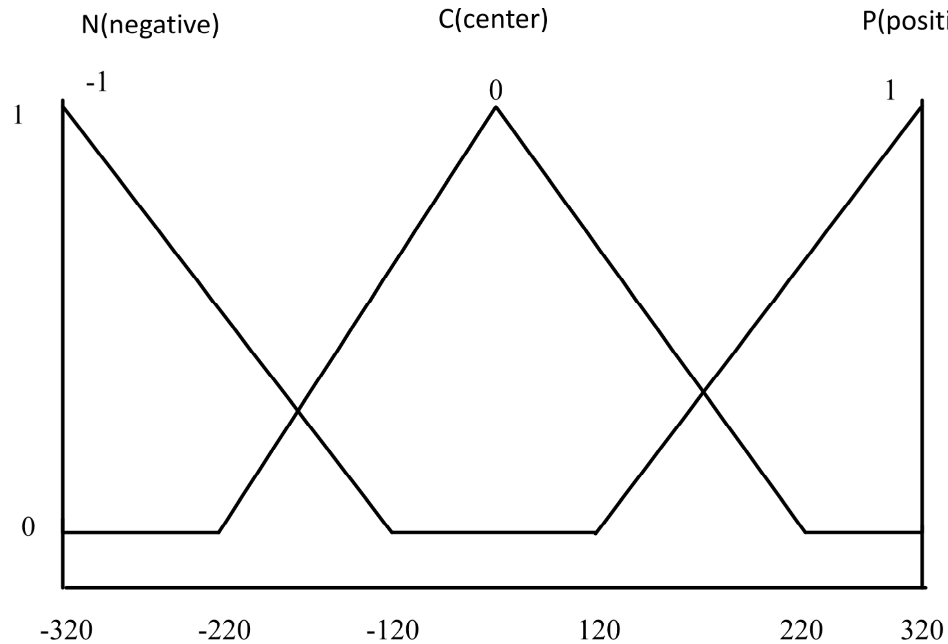

Figure 14: Fuzzified displacement of Center of Mass

According to fuzzified locations and displacements of the center of the mass, fuzzy motion control process generates rotation command for the robot based on rotation rule given in table 1. Based on the table, for example, when the location of the center of the mass is on the right hand side and the displacement of the center of the mass is positive, the right rotation speed cannot catch the moving right speed of the forward sign. Therefore, the right rotation speed should be increased. When the location of the center of the mass is on the right hand side but the displacement of the center of the mass is negative, the right rotation speed is too fast and should be decreased to avoid overshoot.

Table 1
Robot Rotation Rules

Robot rotation command		*Fuzzified displacements of the center of mass*		
		N	*C*	*P*
	FL	-3	-2	-1
	L	-2	-1	0
Fuzzified locations of the center of mass	M	-1	0	1
	R	0	1	2
	FR	1	2	3

EXPERIMENTAL RESULTS

When the strong classifier of the xml file obtained from running *haartraining.exe* was integrated into the image process program, the robot could always detect the forward signs of arrow in tests. Therefore, the performance of the obtained strong classifier is guaranteed.

In the experiment, ↓ forward sign indicates the robot has to turn right to bypass the forward sign while ↑ forward sign indicates the robot has to turn left to bypass the forward sign. The experiment was proven to be successful as shown in the consecutive images in Fig. 15-19. The robot did catch the differences of ↓ forward sign and ↑ forward sign and bypassed them by turning right and left accordingly. The full length video of experiment is available on Youtube [11]. Properly distributing the forward signs, the robot can turn a corner [12].

Figure 15: Start to Search for the Forward Sign

Figure 16: The ↓ Forward Sign is Located

Figure 17: Turns Right to Bypass the ↓ Forward Sign

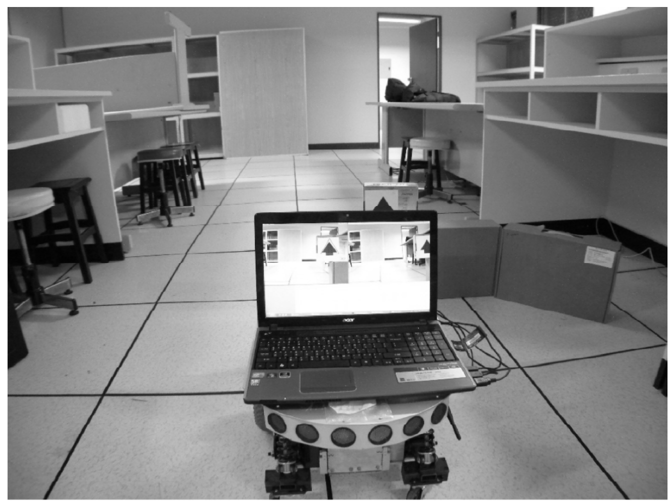

Figure 18: The ↑ Forward sign is located

Figure 19: Turns Left to Bypass the ↑ Forward Sign

CONCLUSIONS

In this paper, the idea of distributing forward signs to arrange the traveling paths for an intelligent robot with computer vision ability is proposed. The structure of integrated system of computer vision and fuzzy motion control for the intelligent robot has been introduced. Image process applying OpenCV library for detecting the object based on Haar-like features has been introduced and successfully verified by experiments. The algorithm of fuzzy motion control process of the intelligent robot has been presented and also verified by the experiment. Therefore, simply redistributing the recognizable forward signs by the robot, a new path for robot can be constructed. The contribution of this paper is to provide a simple and low cost solution of image based robot autopilot system which has extremely low maintenance, lots of implementation flexibility and immunity against electromagnetic interference.

ACKNOWLEDGEMENT

This research is supported by The National Science Council, Taiwan, Republic of China under grant NSC101-2221-E-214-042-.

REFERENCES

[1] Ho-Duck Kim, “SLAM of Mobile Robot in the Indoor Environment with Digital Magnetic Compass and Ultrasonic Sensors, ” ICCAS ’07. International Conference on Control, Automation and Systems, 2007, pp. 87-90.

[2] Jianjun Zhou, “Fuzzy Control for Automatic Steering and Line Tracking of Agricultural Robot,” 2008 International Conference on Computer Science and Software Engineering, Dec. 2008, pp. 1094–1097.

[3] Open Computer Vision Library, *http://sourceforge.net/projects/opencvlibrary/*

[4] Paul Viola, and J. Michael Jones. “Rapid Object Detection Using a Boosted Ca-scade of Simple Features, W IEEE CVPR, **1**(2): 511-518, 2001.

[5] Rainer Lienhart, JochenMaydt, “An Extended Set of Haar-like Features for Rapid Object Detection,” MRL Technical Report, May 2002, revised December 2002.

[6] Kevin M. Passino and Stephen Yurkovich, *Fuzzy Control*, Addison Wesley Longman, Menlo Park, CA, 1998.

[7] Jan Jantzen, *Foundations of Fuzzy Control*. Wiley, 2007.

[8] L. A. Zadeh “Fuzzy Sets,” *Information and Control*, **8**(3): 338–353, 1965.

[9] Mathematics @ University of Iowa, *http://www.math.uiowa.edu/~jorgen/Haar.html B*

[10] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, ACM Transactions on Intelligent Systems and Technology, 2: 27: 1–27: 27, 2011.

[11] *http://youtu.be/BKKwU2IgF4I*

[12] *http://youtu.be/pQVk99rhxQI*