International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 7, Issue 6, June 2017)

An Empirical Research in Autonomous Vehicles Control

Ngo Tung Son[1], Tran Binh Duong[2], Bui Ngoc Anh[3], Luong Duy Hieu[4]
[1, 4]*Department of Computing Fundament, FPT University, Vietnam*
[2]*Department of Software Engineering, FPT University, Vietnam*
[3]*Department of Information Technology Specialities, FPT University, Vietnam*

***Abstract*— Recent years have witnessed a growing attention to automatic-driving vehicles as this is one of the key technologies for the future industry. Even though being successful at many aspects, there has been a long interest in designing an efficient control system for automatic driving vehicles. This paper empirically demonstrates the efficiency of our system which only employs low cost camera for visual sensing. Our approach puts the focus on 2 main objectives in autonomous vehicle control: (1) lane detection and (2) speed and direction decisions for the sake of fast processing. This is to help the vehicle always moves in the right lane while keeping a suitable speed. For decision making fuzzy logic is used for effective reasoning. We test our system in mini automatic-vehicles to show that it is not only efficient but also reliable.**

***Keywords*— Image Processing, Lane Detection, Support Vector Machine, Automatic-Car Control, Fuzzy Control.**

I. INTRODUCTION

In recent years, the researches in autonomous cars have been one of the promising trends. There are many efford spending with the aim of improving road traffic efficiency and safety [1]. Many researches have been done. Most of them use sensing the surroundings of the vehicle as the key technology to build the systems [2, 3]. There are a number of studies that apply advances in machine learning technology in autonomous driving [4]. The objective of this study is to find an efficiency approach to construct the software system. It should be accuracy and high performance. In general, our method includes 4 major steps (see [figure 1]). Firstly, the "pre-processing" transforms input images from the camera into well-format data structure for further processing. "Lane detection" then will be applied to detect the road's lane. The road type (curve or straight) recognized by "Road Recognizer" which is a SVM classifier. Finally, the "Fuzzy controller" receives the result of other steps as inputs to determine the direction and speed of the car. These components use some of formal techniques that described in the next section as the background.

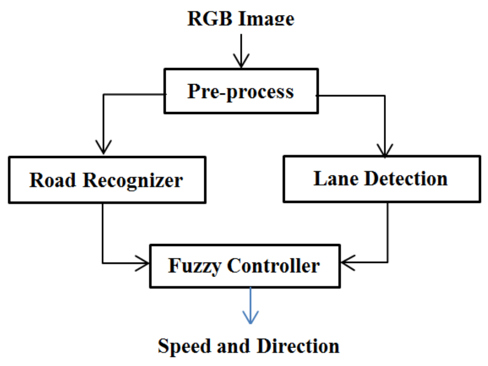

Figure 1: System Overview

II. BACKGROUND

A. Hough Line Transform

The standard Hough line Transform is a famous technique to detect the lines in the input image, was introduced in [5]. This is the common use technique for road's lanes detection. For example in [6] Shanti Bhushan et al used the Hough line to construct lane detection model. The idea of this method can be summarized as following: on the coordinate system, draw a straight line that looks like [Figure 1]. The line can be expressed as:

$$y = -\frac{\cos(\theta)}{\sin(\theta)} * x + \frac{\rho}{\sin(\theta)} \Leftrightarrow \rho = x * \cos(\theta) + y * \sin(\theta)$$

Where: p is the length of line r that is perpendicular of the line and passing the center of coordinate system. θ is the angle between r and x-axis, $p \in [-D, +D]$ with D is the length of the diagonal of the input image. $\theta \in [0,2\pi]$.

There are some other versions of Hough line transform reviewed in [7]. The Probabilistic Hough Line Transform which gives as output the extremes of the detected lines.

B. Support Vector Machine (SVM)

In this study, classification model will be used to recognize the types of roads to support the vehicle in determining the corresponding incremented or decremented velocity as well as the direction of movement.

S. Arunadevi provided a survey on image classification algorithm based on per-pixel in [8]. SVM seems to be an efficient supervised binary classification technique. it was first introduced by Vapnik in [9].

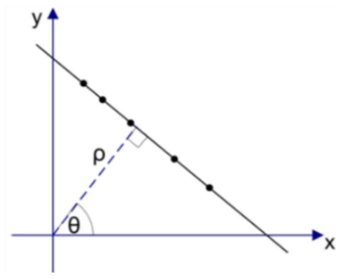

Figure 2: Rho and theta representation of a straight line. Each line has a unique parameter set (ρ, θ).

Considering the binary classification problem – classify data into 2 classes (+1 and -1). The training set $D = \{(x_i, y_i) \mid i = 1,2,\dots,n, x_i \in R^p, y_i \in \{+1,-1\}\}$ where x_i is an input vector in p dimension and y_i is the pre-classified class of x_i. The need is to find the hyperplane with maximum Euclidean distance to the closest training examples. The equation of the hyperplane is: $w^T \cdot x + b = 0$. The separator property can be formalized as:

$$w^T \cdot x_i + b \geq 1 \text{ iff } y_i = +1$$
$$w^T \cdot x_i + b \leq 1 \text{ iff } y_i = -1$$

The optimization problem of SVM is following:

$$\text{Minimize } \left(\frac{1}{2}(w^T \cdot w)\right)$$

$$\text{Constraints } \quad y_i(w^T \cdot x_i + b) \geq 1 \ \forall\, i \ \in 1..n$$

The optimal hyperplane has the minimum expected error of classification on an unseen and randomly selected example. SVM can be generalized to handle non-linear separable in 2 ways. First, adding slack variables: Where C is a constant to tradeoff between margin and training error.

$$\text{Minimize } \left(\frac{1}{2}(w^T \cdot w) + C \cdot \sum_{i=1}^{n} \xi_i\right)$$
$$\text{Constraints } y_i(w^T \cdot x_i + b) \geq 1 - \xi_i \ \forall\, i \ \in 1..n$$

Second, using kernel function $K(x_i, x_j)$ the simplest case is $K(x_i, x_j) = x_i^T \cdot x_j$. However the kernel function can be more complicated case makes SVM suitable for non-linearly separable problems.

The interesting of SVM is that the normal of the decision surface is a linear combination of examples. The decision function can be written in the following form:

$$f(x) = sign\left(\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot K(x_i, x_j) + b\right)$$

There are many researches applied SVM in image classification such as: [10] Jitendra Kumar used SVM with RBF kernel, [11] S. Agrawal et al classify color image on features extracted from histograms of color components …etc. We applied SVM classifier to support the decision of reducing or increasing the velocity.

C. Fuzzy Logic and Fuzzy Control

In the traditional logic, any expressions only take the value of 0 or 1. Unlike so, a fuzzy logic expression can take a float in [0, 1]. In fuzzy logic, an output can be high to some degree, low to some degree, both at the same time. Fuzzy logic is very suitable to be applied in engineering applications (read more in [12]). In [13],J. E. Naranjo et al described the way they use fuzzy logic in automated vehicle control.

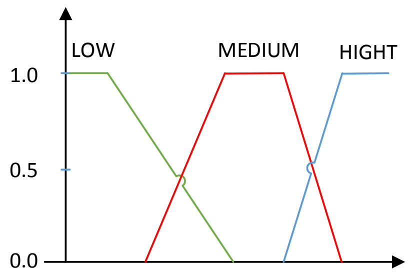

Figure 3: Membership Function for a fuzzy variable

The mappings of input variables into membership functions and truth values then makes decisions for what action to take based on a set of rules, each of the form: IF […] then [….]. This combination of fuzzy operations and rule-based inferences describes a fuzzy system. [Figure 2] illustrate an example of membership function for a fuzzy variable.

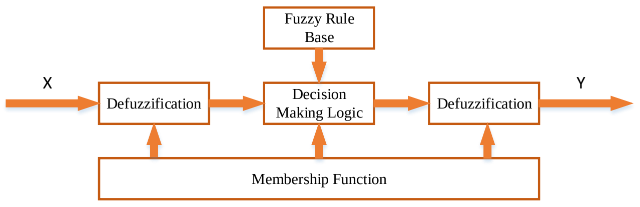

Figure 4: Basic configuration of Fuzzy Control System

The fuzzy control consists of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches and so on, to the appropriate membership functions and truth values (fuzzification). The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value (defuzzification). [Figure 3] shows the phases of the fuzzy control system.

III. Implementation

A. System Configuration

The automatic-car used in our experiment was designed compact. The major part is a computer that aims to perform most of processing (for example: image processing...). It connects to the rest of the car like a camera ([Figure 5]), Arduino. Arduino will control the engines and sonar sensor.

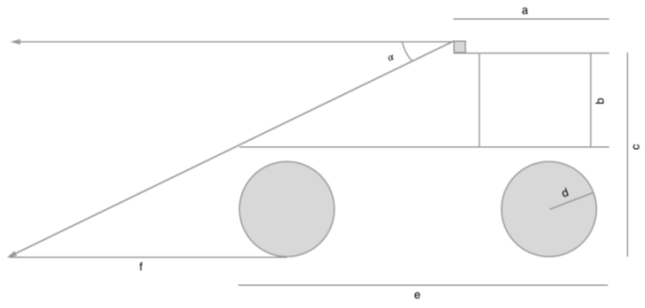

Figure 5: Configuration of used car in the experiments

Specifications provided by the manufacturer are listed in [Table 2]. However, in our empirical conditions, there are some errors in the obtained parameters. Therefore the measurements have been reworked based on our car design. [Figure 6] describes the parameters where their values can be referenced in [Table 1].

Table I
Actual Parameters Values of Experimented Car

Parameter	value	description
A	150 mm	The location of the camera
B	90 mm	
C	220 mm	Height of the car
D	50 mm	Wheel radius
E	490 mm	Length of the car
F	350 mm	The distance from the front wheel to the end of the blind spot
tan(α)	0.433333	View angle of the camera

B. Pre-Processing

Before putting the captured frames from the camera as RGB images [figure 6] into processing algorithms, these images need to be transformed accordingly.

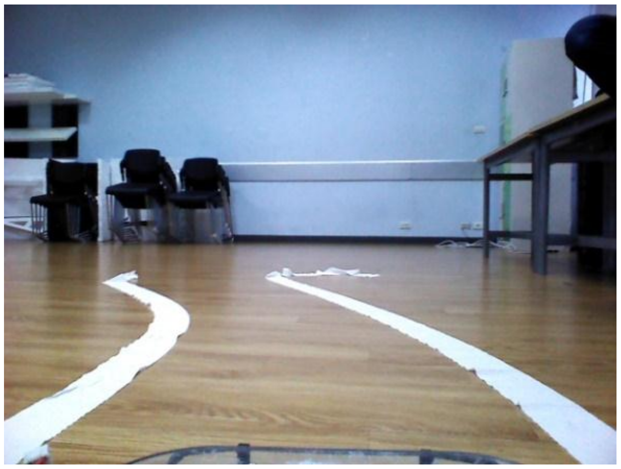

Figure 6: Example of Input RGB image from the camera

A region in the photograph carries different information. From the bottom up, this information is organized according to events that may occur over time. Thus, the bottom part of the photo is that the information must be processed immediately. This idea is similar to the idea of [14] when using regions of interest. Input image then will be cropped, noise removed, gray transformed, histogram equalized, binary image converted. In the experiments, we observed that 1/4 of bottom part is suitable for lane detection. [Figure 7] illustrates an example of preprocessing. The step 4 provides output binary image.

C. Lane Detection

The module receives binary output of the pre-processing as its inputs. Firstly, the lane detection will be started by finding suitable contours. Because the road lanes are usually of a certain width, so we remove the contours of an inappropriate size. After that, the method divides contours into three equal areas: Left, Middle, and Right. For each area, we find the contours base on captured rule:

- *IF contours found in the left and the right THEN ignore contours in the middle.*
- *ELSE IF contours found in the left and the middle THEN ignore contours in the right.*
- *ELSE IF contours found in the right and the middle THEN ignore contours in the left.*
- *ELSE IF contours found only in the middle THEN only considering these contours.*

- *OTHERWISE no lane detected.*

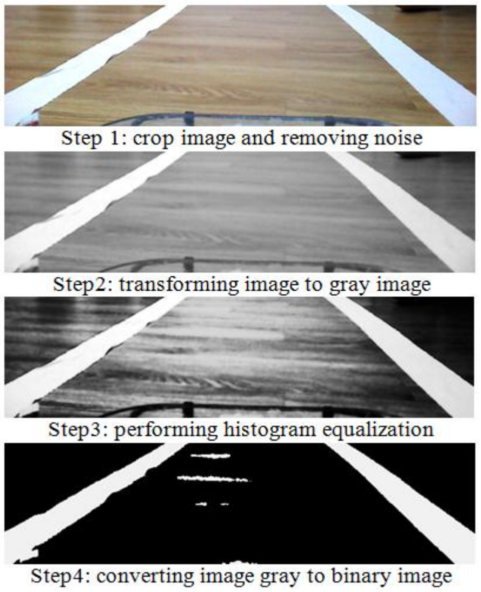
Step 1: crop image and removing noise
Step2: transforming image to gray image
Step3: performing histogram equalization
Step4: converting image gray to binary image

Figure 7: Example of pre-processing steps

Next step, the Probabilistic Hough Line Transform applied to find the straight lines which are candidates for lanes. Every line is presented by a pair of 2 points, pointBottom and pointTop. The angle between the line and OY axis can be estimated by using getTheta function (see [Figure 8]). If there are some similar lines (almost parallel and very close to each other), so only line with smallest theta will be selected. Straight lines that are too close to the center point will also be eliminated.

```
getTheta(pointBottom, pointTop)
Begin
     define: dx = pointTop.x-pointBottom.x;
     define: dy = pointBottom.y-pointTop.y;
     IF (pointBottom.x= =pointTop.x)
          return 0
     ELSE IF (pointBottom.y= =pointTop.x)
          return  (pointTop.x < pointBottom.x) ? -90 : 90
     ELSE
          return (dx < 0) ?  -arctan(-dx / dy) * 180 / pi   :
arctan(dx / dy) * 180 / pi);
End
```

Figure 8: Algorithm to calculate theta

In empirical, we removed all of horizontal (theta>85) and vertical (theta<=5) lines. In order to decide the lane, 2 lines considered to close each other, if the difference between top and bottom point of each line is less than $\frac{1}{5}$ of frame's width.

[Figure 9] describes an example of lane detection in case of straight road.

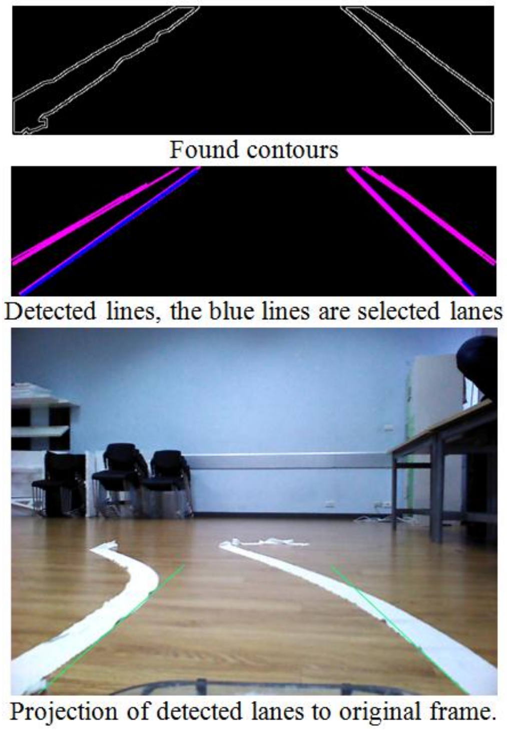
Found contours
Detected lines, the blue lines are selected lanes
Projection of detected lanes to original frame.

Figure 9: Example of lane detection process

There are some others example of lane detection in other cases listed in [Figure 10].

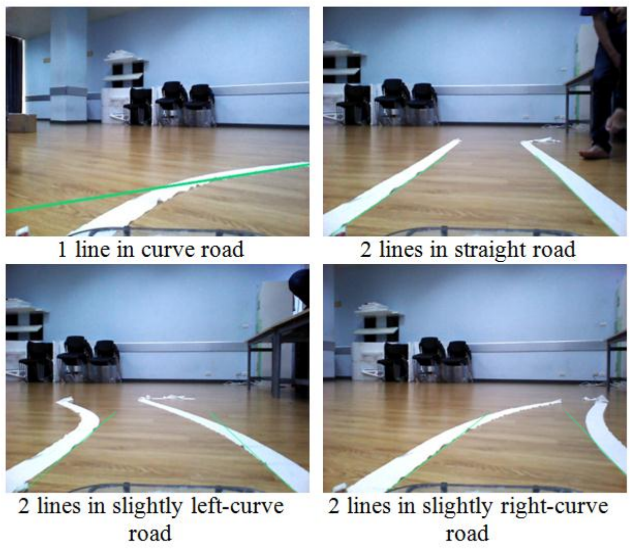
1 line in curve road
2 lines in straight road
2 lines in slightly left-curve road
2 lines in slightly right-curve road

Figure 10: Example of lane detection results

D. Road Recognizer

This module is a SVM classifier using linear-kernel. We use a set of road images, which include both straight lines and curves as training dataset. As described in the pre-processing section, an input image contains information of the 'near future' and 'far future' sequentially from bottom to top. We crop the input image such that the remaining part contains information of 200cm front. These images then will be pre- processed before training. The purpose of this is to support Fuzzy controller in estimating the direction and speed. In the distance of about 200cm, the system can recognize the curve road or straight road to adjust the speed.

The accuracy of the model changes with each simulator road. However the average accuracy is 72% with 500 samples for each of straight lane and curve lane. The accuracy can be improved if we take more samples or use different configurations for SVM model.

E. Fuzzy Controller

In our case, we use fuzzy logic to specify two outputs: out speed {0… 50} and out theta {0… 90} (the direction to turn the car), based on three inputs: the current speed {0… 50}, the angle of the lane line {0… 90} (called angle for short in this section), and the distance from our car to the curve {0… 200}. Fuzzy processing includes 3 sequential steps: fuzzification, processing and defuzzification.

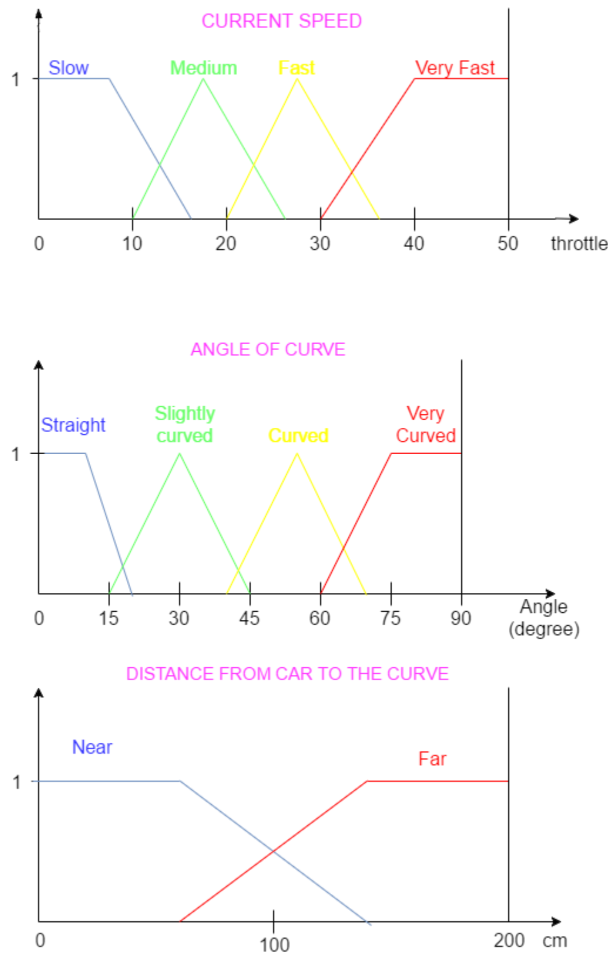

Figure 11: Membership functions of input variables: current speed, angle of curve lane, and distance to the curve

In the first step, three inputs will be divided into many states. Defining the bounds of these states is a bit tricky. There are several experiments have been done. An arbitrary threshold might be set to "slow" from "medium", but this would result in a discontinuous change when the input value passed over that threshold. Membership functions are used to do this.

Next, in the processing step, we create a rules table (see [Table 3]) to process the inputs. An example of applying the rules described in [Figure 12]. IF (Fast (0.25), slightly curved (0.67), near (0.25)) THEN Use min operator, we have: slightly curve (0.25), medium 0.25).

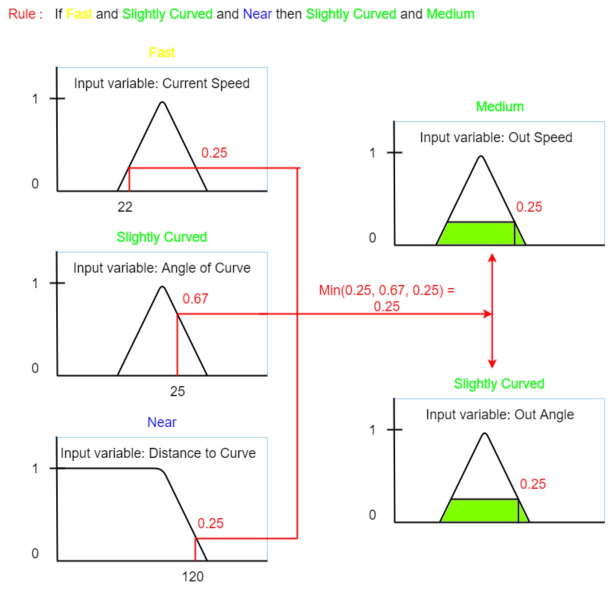
Rule : If Fast and Slightly Curved and Near then Slightly Curved and Medium
Fast
Input variable: Current Speed
1
0
0.25
22
Slightly Curved
Input variable: Angle of Curve
0.67
25
Near
Input variable: Distance to Curve
0.25
120
Medium
Input variable: Out Speed
0.25
Min(0.25, 0.67, 0.25) = 0.25
Slightly Curved
Input variable: Out Angle
0.25

Figure 12: Fuzzy processing on the Rule: IF current speed = 22, Angle of Curve = 25, Distance to Curve = 120

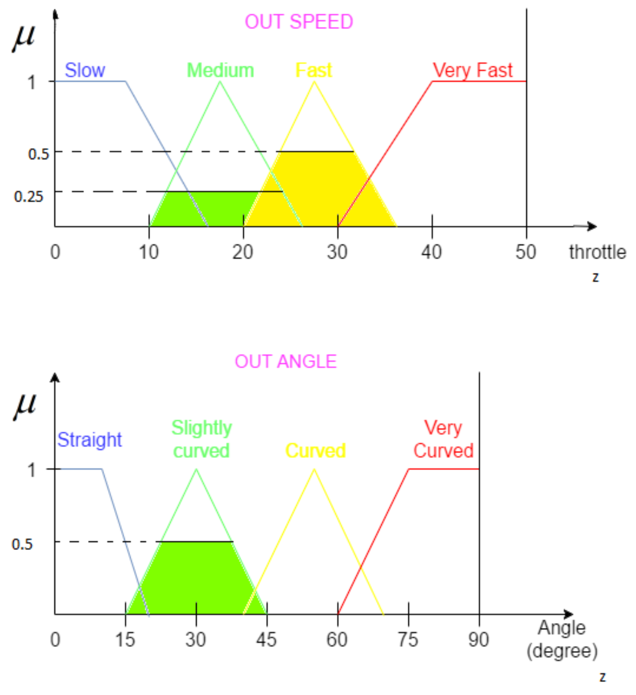

Figure 13: An example of Symmetric Membership Functions.

Lastly, the defuzzification performed. We applied the “weighted average method”.

$$z^* = \frac{\sum \mu_C(\bar{z}).\bar{z}}{\sum \mu_C(\bar{z})}$$

Let denotes the algebraic sum and where *z* is the centroid of each symmetric membership function. An example of the symmetric membership functions are presented in [Figure 13].

IV. EXPERIMENT

In order to test the effectiveness of the method, we have done several experiments on simulator roads. We selected the test results on a road that included curves and lines from the simulator roads (see [Figure 14]). The road is divided into several areas and marked accordingly in alphabet.

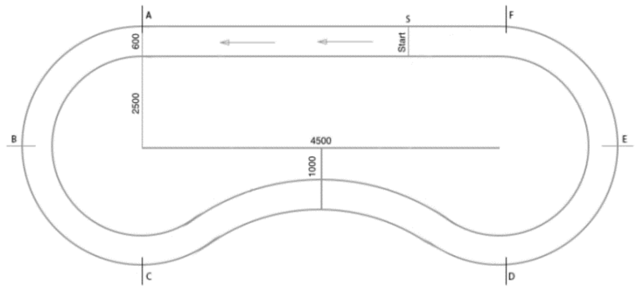

Figure 14: Simulator-Road

There are many different viewpoints to check if the car is running smoothly. Here, we believe that the car is always follow its lane with reasonable speed is good thing. We logged all of the entire differences of the center of the car and the center of the road. [Figure 15] illustrates the differences when the car finishes a round. Although the car started from the S, we conducted the measurements from A as the starting point. The positive value in the graph represents the difference when the car is deflecting to the right-lane of the road. Meanwhile, the negative value represents the difference when it is deflecting to the left-lane of the road. The result shows that the car is gripping the road quite well. Most of the time, the car is skewed to the lane, because we adjust the car to the lane instead of center of the road. This helps us adjust the grip on the road when traveling at strong corners. [Figure 16] illustrates logged speeds of the car when it completes a round in the last experiment (the unit of measurement uses the value of the velocity parameter that transmitted to the motor (not the real speed)). The car starts and ends at S. We can see that the speed is increased when detecting the straight line, and gradually decreased when the curve recognized.

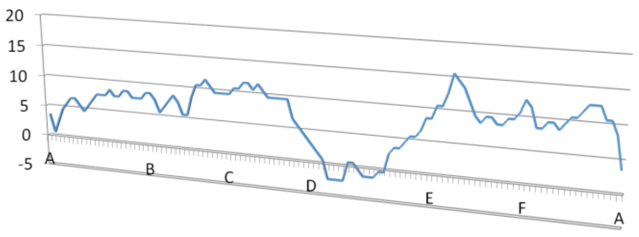

Figure 15: Deviation of car center to the road center

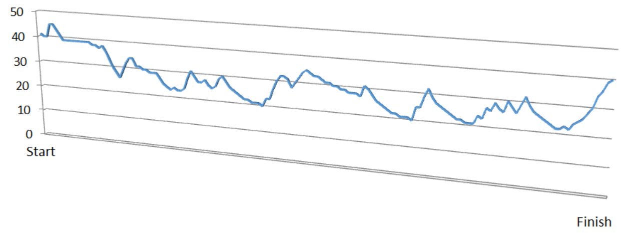

Figure 16: Actual speed in the last experiment

Recorded fps in [Figure 17] also shows that system performance is mostly stable. This includes the task of recording data. Recorded fps in [Figure 17] also shows that system performance is mostly stable. This includes the task of recording data. The performance will be dramatically higher if without writing logs.

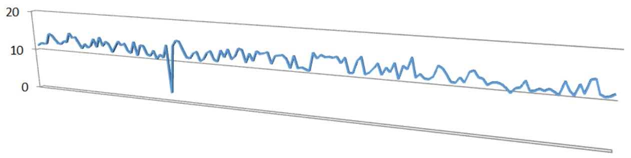

Figure 17: Fps logs in the last experiment

We have done many experiments and the final test presented basically demonstrates the effectiveness of the system. However, in others experiments, on many lightly-curves, the car still accelerate evenly. This sometime causes the car to deviate slightly from the road's centre.

V. CONCLUSION

This study found an empirical method in controlling automatic-vehicles, which is relatively effective. Experiments have been performed on a several simulator-roads. It show that the car run smoothly (always in its lane) and determine a reasonable speed (decreasing when approaching the curve and increasing speed when recognizing straight lines).

The way, we divide the information regions obtained from the camera is a noteworthy point. Working in each of region not only brings about computational performance by reducing size of input matrix for image processing techniques, but also contributes significantly to the accuracy of the system. Our lane detection method is very simple, with low computational complexity. However, it still guarantees the accuracy. This is suitable for systems that require high computing performance.

In contrast, few of our experiments encountered some errors when making light acceleration on the road. The cause of this is because the SVM model is not very accurate.

This can be improved by using more training samples and using different parameters of SVM. The Fuzzy controller actually shows its effectiveness when it comes to support for automated control systems. In further research, we will focus on detecting and avoiding objects in the road. Studies on the identification and compliance of the traffic signals will also be made.

Appendix

Table II
Technical Information of the Devices

Car:	HG P402 1/10 Scale Jeep, Lebtoys, made in China. Battery: Lipo 2S 7.4V 1500 mah, 60 gram
Size	50cm x 24cm x 26cm
Motor:	Brushed RC 540 20T
Camera:	Orbbec Astra, 0.6- to 8-meter range, Resolution - Frames / seconds: • 1280x960 – 10 frames / seconds • 640x480 – 30 frames / seconds Software: OpenNI2 / Astra SDK
Computer:	Board Jetson TK1 Tegra K1 SOC : • GPU: NVIDIA Kepler "GK20a" GPU with 192 SM3.2 CUDA cores • CPU: NVIDIA "4-Plus-1" 2.32GHz ARM quad-core Cortex-A15 CPU with Cortex-A15 battery-saving shadow-core Battery: Lion 11.1V 2200mah 25C. Dimensions: 127mm x 127mm DRAM: 2GB DDR3L 933MHz EMC x16 using 64-bit data
Arduino	Arduino Uno

Table III
Rules Table for Fuzzy Processing

INPUT			OUTPUT	
CURRENT SPEED	ANGLE	DISTANCE	OUT THETA	OUT SPEED
SLOW	STRAIGHT	NEAR	STRAIGHT	MEDIUM
SLOW	STRAIGHT	FAR	STRAIGHT	VERY FAST
SLOW	SLIGHTLY CURVED	NEAR	SLIGHTLY CURVED	SLOW
SLOW	SLIGHTLY CURVED	FAR	SLIGHTLY CURVED	MEDIUM
SLOW	CURVED	NEAR	CURVED	MEDIUM
SLOW	CURVED	FAR	CURVED	MEDIUM
SLOW	VERY CURVED	NEAR	VERY CURVED	SLOW
SLOW	VERY CURVED	FAR	VERY CURVED	SLOW
MEDIUM	STRAIGHT	NEAR	SLIGHTLY CURVED	MEDIUM
MEDIUM	STRAIGHT	FAR	STRAIGHT	VERY FAST
MEDIUM	SLIGHTLY CURVED	NEAR	SLIGHTLY CURVED	MEDIUM
MEDIUM	SLIGHTLY CURVED	FAR	SLIGHTLY CURVED	FAST
MEDIUM	CURVED	NEAR	VERY CURVED	SLOW
MEDIUM	CURVED	FAR	VERY CURVED	SLOW
MEDIUM	VERY CURVED	NEAR	VERY CURVED	SLOW
MEDIUM	VERY CURVED	FAR	VERY CURVED	SLOW
FAST	STRAIGHT	NEAR	SLIGHTLY CURVED	MEDIUM
FAST	STRAIGHT	FAR	STRAIGHT	VERY FAST
FAST	SLIGHTLY CURVED	NEAR	SLIGHTLY CURVED	MEDIUM
FAST	SLIGHTLY CURVED	FAR	SLIGHTLY CURVED	FAST
FAST	CURVED	NEAR	CURVED	MEDIUM
FAST	CURVED	FAR	CURVED	MEDIUM
FAST	VERY CURVED	NEAR	VERY CURVED	SLOW
FAST	VERY CURVED	FAR	VERY CURVED	SLOW
VERY FAST	STRAIGHT	NEAR	STRAIGHT	MEDIUM
VERY FAST	STRAIGHT	FAR	STRAIGHT	VERY FAST
VERY FAST	SLIGHTLY CURVED	NEAR	SLIGHTLY CURVED	SLOW
VERY FAST	SLIGHTLY CURVED	FAR	SLIGHTLY CURVED	FAST
VERY FAST	CURVED	NEAR	VERY CURVED	SLOW
VERY FAST	CURVED	FAR	VERY CURVED	SLOW
VERY FAST	VERY CURVED	NEAR	VERY CURVED	SLOW
VERY FAST	VERY CURVED	FAR	VERY CURVED	SLOW

REFERENCES

[1] Autonomous Cars: Self-Driving the New Auto Industry Paradigm, MORGAN STANLEY RESEARCH, November 6, 2013.

[2] Overview of Autonomous Vehicle Sensors and Systems, Jaycil Z. Varghese, Professor Randy G. Boone, Proceedings of the 2015 International Conference on Operations Excellence and Service Engineering Orlando, Florida, USA, September 10-11, 2015.

[3] Multi-sensor data fusion for autonomous vehicle navigation through adaptive particle filter, Tehrani Nik Nejad Hossein, Seiichi Mita, Han Long, Intelligent Vehicles Symposium (IV), 2010 IEEE

[4] An Empirical Evaluation of Deep Learning on Highway Driving, An Empirical Evaluation of Deep Learning on Highway Driving, B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al., [arXiv], (2015).

[5] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns", US Patent 3,069,654, Ser. No. 17, 7156 Claims, 1962.

[6] "Real-Time Lane Detection for Driving System Using Image Processing", IRJET, Volume: 02 Issue: 05, Aug-2015.

[7] Allam Shehata Hassanein , Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab, "A Survey on Hough Transform, Theory, Techniques and Applications", Informatics Department, Electronics Research Institute, El-Dokki, Giza,12622, Egypt.

[8] S.ARUNADEVI, Dr. S. DANIEL MADAN RAJA, A Survey on Image Classification Algorithm Based on Per-pixel, International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014 ISSN 2091-2730.

[9] Vapnik (1995), The Nature of Statistical Learning Theory. Springer, Berlin.

[10] Jitendra Kumar, Image Classification using SVM-RBF in the field of Image Processing, International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences (IJIRMPS) Volume 1, Issue 2, December 2013.

[11] S. Agrawal, N. K. Verma, P. Tamrakar and P. Sircar, "Content Based Color Image Classification using SVM," 2011 Eighth International Conference on Information Technology: New Generations, Las Vegas, NV, 2011, pp. 1090-1094.

[12] Timothy J. Ross "Fuzzy Logic With Engineering Applications" Second Edition.

[13] J. E. Naranjo, M. A. Sotelo, C. Gonzalez, R. Garcia and T. D. Pedro, "Using Fuzzy Logic in Automated Vehicle Control," in IEEE Intelligent Systems, vol. 22, no. 1, pp. 36-45, Jan.-Feb. 2007.

[14] Design and Implementation of Autonomous Car using Raspberry Pi, Gurjashan Singh Pannu, Mohammad Dawud Ansar, Pritha Gupta, International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 9, March 2015.