ADES: AUTOMATIC DRIVER EVALUATION SYSTEM

by

Kemal Kaplan

B.S. in Computer Engineering, Boğaziçi University, 2000

M.S. in Computer Engineering, Boğaziçi University, 2003

A Thesis Proposal

submitted to the Thesis Supervisory Committee

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in

Boğaziçi University

2009

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

EU	European Union
GNP	Gross National Product
ADES	Automatic Driver Evaluation System
RF	Radio Frequency
RFID	Radio Frequency Identification
MHT	Multiresolution Hough Transformation
HMM	Hidden Markov Process
LDA	Linear Discriminant Analysis
SVM	Support Vector Machines
GA	Genetic Algorithm
LIDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GLONASS	The Global Navigation Satellite System

1. INTRODUCTION

Traffic accidents are one of the main causes of death and economic loss in most of the developed countries. According to the Road Safety Action Programme of European Commission, more than one million accidents a year cause more than 40 000 deaths and nearly two million injuries on the roads. In addition, the direct and indirect cost has been estimated at 160 billion euros, which is nearly two percent of the EU's GNP. Unfortunately these thrilling numbers are increasing every year. The number of road accidents in Turkey is approximately half million in 1998, where this number exceeds 720,000 in 2006. However, the most regretful fact is that, 97 percent of the accidents in our country is caused by driver mistakes.

The aim of the *ADES* project is developing a framework to reduce the rate of the accidents based on driver faults by detecting and reporting the traffic violations. If the driver violates rules, the system gives a negative response with full justification otherwise it should give a positive response.

1.1. Motivation

Although the vehicle manufacturers deploy more intelligence in their newest models, the current applications are usually focused on driver assistance and early warning systems. However, in the near future, intelligent vehicles will also enforce the traffic regulations. For example, speed limit and traffic light violations are going to be detected by cars. With this motivation, the ADES project is planned to be a framework for evaluating the drivers against the traffic rules. The applications of the resulting system include but not limited to the following items;

- Deter drivers from violating traffic rules
- Automation of driver license examinations
- Supervising the development of autonomous urban driving

2. PROBLEM STATEMENT

The focus of this work is developing a system which can be used for evaluating the actions of vehicle drivers especially in urban environments. Most of the accidents can be avoided if the users become more sensitive about the traffic rules. However, it is not feasible to place a police person at each corner even in the smallest city. Therefore, automation of driver evaluation is inevitable for forcing drivers to obey the traffic rules.

Similar to the structure of the designed solution, the statement of the problem can be divided into two major categories. The first category is about the acquisition of the related data as quickly, cheaply and accurate as possible. However, there are numerous difficulties even for the basic sensors. Naturally, the more advance the sensors, the more complicated the problems they have. For example, the RFID readers may suffer from tag collision, which is relatively easy to resolve. However, in vision, the system may suffer from lightning conditions or unrecognizable objects which requires more complicated solutions.

The second category of the problem is related with designing the inference engine. Even if the data acquisition modules provide the most accurate knowledge, the system may become too complicated to manage with simple if-then rules. Moreover, usually the sensors and data interpreting modules usually introduce considerably large amount of uncertainty to their outputs. As a result, the inference engine should be able to cope with both the complexity and the uncertainty of the environment.

In addition to these scientific difficulties, there are also some engineering problems. The system should process a large amount of data with limited computational power. And the quality of the data is closely related with the selected equipment. Moreover, the proper placement of the sensors to the vehicle may effect the system performance dramatically. The system may also require to integrate with different systems like existing traffic control agents or the vehicle itself. In addition, the cost of the sensors and additional peripherals, like RFID tags, should be considered carefully.

Although the expected cost of a single RFID tag is relatively small, thousands of RFID tags may become the most expensive part of the entire system. If the RFID tags are placed on cars, then the readers should be placed near the traffic signs. Otherwise all cars should be equipped with RF readers and traffic signs should have RFID tags which identifies them. In each case the number of required RF readers and RFID tags are quite large even for deploying the system in a small town. Since the RF readers are more expensive and require more maintenance, they may be placed in cars where it is relatively easy to obtain power source and keep the device under control. In addition, the cost of the reader may be charged to the vehicle manufacturers and can be subsidized by tax reductions.

3. PROPOSED APPROACH AND RELATED WORKS

Unlike the basic definition of the problem, which can be easily defined as revealing the guilty driver, it is not very easy to propose a fully detailed approach for this purpose. It is clear that any proposed solution should be aware of the actions of the driver and be able to judge the decisions of the driver. However, there are many alternative ways to accomplish or workaround every single subproblem which are roughly stated in the previous chapter.

In ADES project, the task is divided into two major parts. The first part acquires the sensor data and processes it for the use of the inference engine. The second part, which is the inference engine, uses the sensor values as facts, and comes up with the final decision according to its rules in the knowledge base. Since the rules in the knowledge base will be derived from common traffic rules for this application, they are going to be introduced by human experts. As a result, the inputs of the system are sensor values and predefined expert rules, and the output is the evaluation of the driver's moves as shown in Figure 3.1.

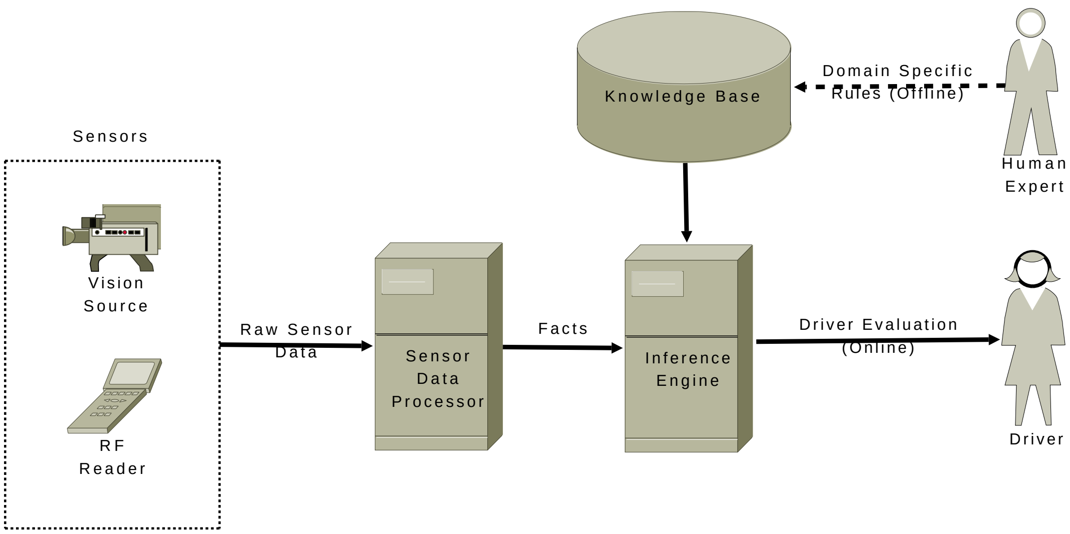

Figure 3.1. Basic system architecture of ADES project.

Although, to the authors' knowledge, there is no significant study available concerning the autonomous driver evaluation by the means of on vehicle systems, there are numerous works regarding autonomous driving, driver assistant systems, and autonomous traffic control systems. Obviously, these studies dealt with common problems and proposed remarkable solutions which are also applicable in this problem domain.

The decomposition of the proposed solution and the related work for each submodule are given in the next sections.

3.1. Data Acquisition and Processing

The most common sensors for mobile agents are video cameras, global navigation satellite systems, light and radio waves based detection and ranging systems. The proposed approach utilizes a single video camera which is placed in the front of the vehicle where the road and other important objects can be seen easily. In addition, the RFID technology also will be inspected in detail in order to reduce the possibility of the errors and increase the reliability of the system. Brief description of alternative sensors are also given in the last subsection.

3.1.1. Vision Module

Since the vision sensors usually provide valuable information about the environment, they are inevitable for most of the autonomous systems. ADES project also has a mono vision sensor, which is a video camera placed on the vehicle. The goal of the vision module is not only capture the image of the environment, but also process this image in order to provide valuable information for the inference engine. The proposed vision processing subsystem will be able to detect and track lane markings, traffic lamps and signs, and various obstacles like other vehicles and pedestrians.

In this section the details of the proposed lane and traffic sign detection approach is given. In addition, traffic lamp and obstacle detection techniques will be considered soon.

3.1.1.1. Lane detection and tracking. For lane detection, Hough Transform is one of the most common techniques. However, there are many other techniques in the literature for lane detection. Pomerleau et al. used neural networks in their ALVIN system. Dynamic programming is used for eliminating outliers from detected line segments by Kang et al. In addition, Wang et al. used B-splines in order to fit lane markings. Different techniques have also been proposed for tracking the detected lanes and modeling the road. Kalman filtering and particle filtering are the two most common tracking techniques used in lane tracking methods. A more detailed survey for lane detection strategies can be found on.

The classical Hough transformation approach processes the entire vision data in order to detect the lines. This scenario has two main drawbacks. First, the occluded lines (i.e. another car passing through the line) become noisy since the transformed relative intensity of the line decreases. Second, the relative intensity of the lines also decreases at the curves in the road. The proposed solution divides the road image into partitions, where the sizes of the partitions are inversely proportional to the distance of the partition to the vehicle.

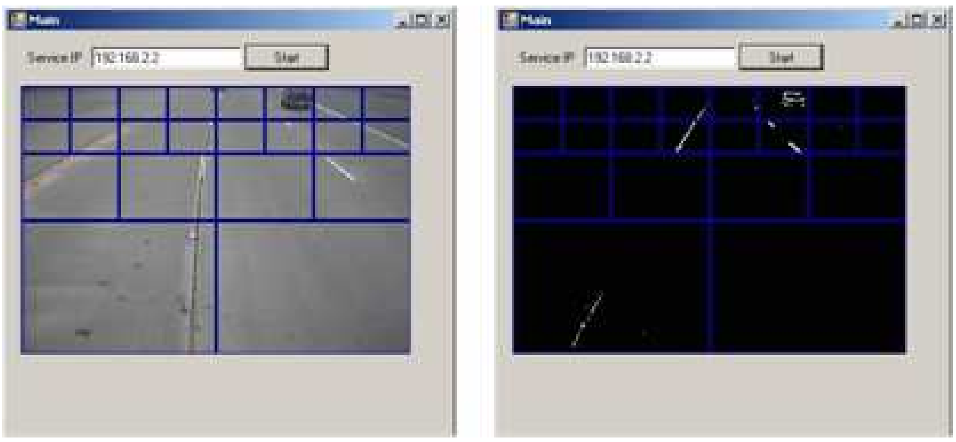

Figure 3.2. (a) Partitioned image, (b) Binary image.

The proposed approach employs MHT for lane detection, followed by two HMM models for radius and orientation of the candidate lanes. After the image is partitioned, a separate Hough transformation is applied to each single partition. The most intense line in each partition, which is the candidate line segment, is taken into consideration in order to find the global lanes in the image. Since the Hough lines are represented in polar coordinates *(r, θ)* instead of rectangular coordinates *(x, y)*, the candidate lines are grouped according to their slopes and distances to the center of the image as well as their intensities. The center of the frame is chosen as the bottom point.

Figure 3.3. (a) Candidate lines, (b) Transformed line, (c) Detected lines.

The transformation of the lines basically changes the center point of the polar coordinates for each transformed line which is achieved by the following translation

$$\begin{aligned} r' &= r + (x - x')\cos(\theta) + (y - y')\sin(\theta) \\ \theta' &= \theta \end{aligned} \tag{3.1}$$

where *(r', θ')* is the polar coordinates of the transformed Hough line *(r, θ)*. Note that the translation of the center of the Hough transformation is from *(x, y)* to *(x', y')*. After the lines are grouped, the most intense three clusters are assigned as the lanes. However, there may be less than three lanes if the sum of the intensities of the candidate lines is less than a threshold value.

For lane tracking, HMM is used to represent the relation between the current frame and its successor. Each line in a specific frame is represented by an individual *(r, θ)* pair. In the succeeding frame, the process will most probably observe the same line at *(r', θ')* which is not very far from the position of the line in the previous frame. The probability of observing *(r', θ')* pair in the next frame is modeled as an HMM problem. In addition, θ and r values are modeled by two different HMM. The θ value is discretized as (0, 1, 2, 3…178, 179) where the r value is discretized at the pixel level. This discretization schema is used in both transmission and emission matrices. The emission probability matrix shows the probability of observing θ' (or r') in the next frame, having observed θ (or r) in the current frame. In our implementation, the observation and state transition matrix values are derived from two Gaussian distributions with different deviations. The deviation of the transition matrix is assigned to a smaller value than the observation matrix, which means, the state transition matrix aims to preserve the current state where the observation matrix promotes the exploration behavior.

3.1.1.2. Traffic sign detection and tracking. There are numerous different methods for detection and recognition of traffic signs. Escalera et. al. used color thresholding and shape analysis for detection neural networks for recognition. Fang et al. developed a neural network based approach for detection and Kalman Filter for tracking the signs. Hsu et al. proposed a template matching based for detection and matching pursuit filters, which decompose patterns into two dimensional wavelet expansions, for recognition. Bahlmann et al. used Haar wavelet features obtained from AdaBoost training for detection and LDA followed by maximum likelihood approach for recognition. Loy et al. modified radial symmetry transform for detection. Bascon et al. used shape classification using linear SVMs for detection and Gaussian-kernel SVMs for recognition. Escalera et. al. and Soetedjo et al. proposed GA based methods for sign detection. Gao et al. uses shape and color based feature extraction methods for recognizing signs. Similarly, H. Fleyeh proposed a fuzzy approach for color detection and segmentation of traffic signs.

The proposed approach for sign detection and tracking in the ADES project is based on GA and a modified version of radial symmetric transform after an image binarization.

The GA implementation uses the coefficients of a geometric transformation applied to a set of points which describes the characteristics of any searched template. The formulation of geometric transformation, which includes affine and perspective transformations, is

$$\begin{vmatrix} u' \\ v' \\ w \end{vmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \tag{3.2}$$

$$u = u'/w \tag{3.3}$$

$$v = v'/w \tag{3.4}$$

where x and y are the coordinates of the sample point from the template describing set of points. u and v are the transformed point on the image. a, b, d, e provides rotation, scaling and shearing where c and f used for translation. In addition, g and h provides perspective transformation in two dimensions. These coefficient values or a subset of them can be used in the chromosome encoding of the GA.

The effect of the transformation can be visualized better with a simple example. Assume that a, b, c, d, e, and f coefficients are used in the encoding of the GA chromosome. In addition, g and h are left zero for simplicity. Also assume that 12 points are used in representing the characteristics of a circular object. For this scenario, we can conclude that a chromosome with the transformation coefficients in Equation 3.5 can yield to the transformed circle points in Figure 3.4. The points on the left of the figure are the characteristic points of the circular template, which are 12 equidistant points on the unit circle, and the right points are the translated, scaled, and rotated counter parties in the transformed domain.

$$\begin{vmatrix} u' \\ v' \\ 1 \end{vmatrix} = \begin{vmatrix} 2 & 1 & 100 \\ 1 & 2 & 50 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \tag{3.5}$$

These new points are used to calculate the fitness of candidate traffic sign location. The fitness value is formed as a function of the colors of the underlying pixels for each

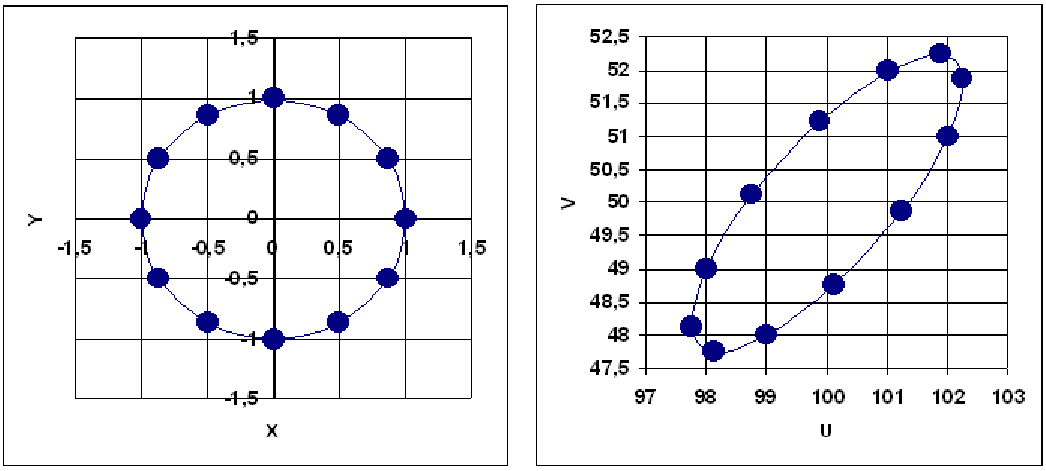

Figure 3.4. Template characteristic points in (x,y) domain, and (u,v) domain after geometric transformation.

BAND	Low Frequency	High Frequency	Ultra-High Frequency	Microwave
Typical RFID Frequencies	125-134 kHz	13.56 MHz	433, 865-956 MHz	2.45 Mhz
Read Range	<0.5m	<1.5m	<5m	<10m
Data Transfer Rate	1 kbit/s	25 kbit/s	30-100 kbit/s	<100 kbit/s
Sample Application	Car immobilizer	Contac-less travel cards	Logistics	Electronic toll collection

Table 3.1. Properties of most common RFID systems.

transformed point in the processed snapshot.

3.1.2. RF Module

The RF module is assumed to use information gathered from the RFID tags which are placed special points on the roads. An RFID system consists of an antenna and a transceiver which can read the radio frequency and transfer the information to a processing device (reader) and a transponder, or RF tag, which contains the RF circuitry and information to be transmitted. The antenna provides the means for the integrated circuit to transmit its information to the reader that converts the radio waves reflected back from the RFID tag into digital information that can then be passed on to computers that can analyze the data. The most common types of RFID modules and specific properties are given in Table 3.1.

<u>3.1.2.1. Sample applications.</u> The implementation of the RFID technology for the proposed application will contain a reader which is capable of reading tags up to two meters. If the RF reader can be placed somewhere near the front bumper, this distance is quite enough to read necessary markings. For the sample scenario given in Figure 3.5, there is an RFID tag which is placed very close to a no right turn sign. In order to detect the violation of the turning rule, another RFID tag is also placed a few meters ahead the crossroad in the restricted direction. According to this setup, the system can easily conclude that the driver violates the traffic rule if the RF reader on the car detects the first RFID tag and then the second tag in this specific order.

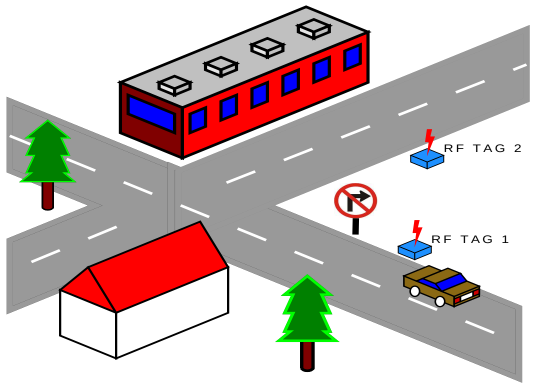

Figure 3.5. Sample RFID application scenario.

A slightly different scenario can be applied for the traffic lamps. However this time the first RFID tag should be an active one which is activated only when the red light is on. The proposed positions of the RFID tags are shown in Figure 3.6. The violation of traffic rule can be detected in the same way as it is done in the previous scenario. The easiness of producing similar scenarios for different traffic rules makes the RFID technology very promising for the ADES project.

3.1.3. Other sensor technologies

There are also other sensors in autonomous driving. One of the most common sensor is RADAR which is a system that uses electromagnetic waves to identify the range, altitude, direction, or speed of both moving and fixed objects. However, more

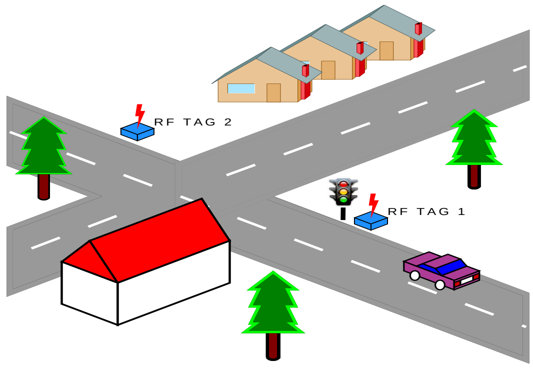

Figure 3.6. Sample RFID application scenario with traffic lamp.

recent applications use the laser scanning radar sensor LIDAR, which measures the relative position of the controlled vehicle with respect to its preceding vehicle. In addition to ranging sensors there are Global Navigation Satellite Systems (GNSS) which uses a version of triangulation to locate the receiver, through calculations involving information from a number of satellites. The most popular GNSS is probably the NAVSTAR GPS, maintained in the United States. There are other systems using satellites as references in the navigation process. Galileo is one GNSS system under development by the European Union (EU). Furthermore, Russia has a system under restoration called GLONASS.

3.2. Inference Engine

In order to propose a suitable reasoning method for ADES project, different planning techniques used in autonomous vehicles are investigated. Sukthankar et al. presented a distributed solution, which consists of a collection of reasoning objects that votes upon a set of possible actions for dealing with the complexity of tactical reasoning. Rosa et al. implemented a fuzzy expert system to organize traffic regulations in town area. Al-Shihabi et al. proposed a framework for modeling driver behavior which has perception, emotions, decision-making, and the decision-implementation units formed of fuzzy variables and if-then rules. Gao and Zhou developed a fuzzy rule based control strategy selection approach for mobile navigation which utilizes Fuzzy Reasoning Petri Nets for system modeling and parallel reasoning. Lattner et al. introduced a knowledge-based approach to behavior decision for intelligent vehicles which uses qualitative representation of the information perceived by the sensors. Hanford et al. presented a review of software platforms for autonomous vehicles and their artificial intelligence development capabilities. Ferguson et al. presented a reasoning framework, which consists of the mission, behavioral, and motion planning components perform the reasoning for an autonomous vehicle navigating through urban environments. Vacek et al. used case-based reasoning in order to predict the progress of the current situation and to select the appropriate behavior. Toit et al. described a finite state machine model to manage the complexity of the situational reasoning subsystem which recognizes the current situation to impose the correct traffic rules.

Since the acquired sensor values will be noisy, a modified version of fuzzy expert system is planned to be used in ADES project. However, the ultimate implementation may differ in the physical application.

3.2.1. Traffic Rules

The proposed inference system should be able to reason for the violations of the following traffic rules, or at least a major subset of them.

- Not following the rightmost lane unless there is an overriding sign. Has a penalty of 128 TL according to Turkish laws.
- Violating the red traffic light. Has a penalty of 128 TL according to Turkish laws.
- Not obeying the signs and land markings. Has a penalty of 61 TL according to Turkish laws.
- Exceeding speed limits up to 30 percent. Has a penalty of 128 TL according to Turkish laws.
- Exceeding speed limits more than 30 percent. Has a penalty of 265 TL according to Turkish laws.
- Not slowing down while entering crossroads, passing hills, crosswalks. Has a penalty of 61 TL according to Turkish laws.
- Entering forbidden roads. Has a penalty of 128 TL according to Turkish laws.

- Following the leading vehicle from an unsafe distance. Has a penalty of 61 TL according to Turkish laws.
- Driving too slow or decelerating unexpectedly. Has a penalty of 61 TL according to Turkish laws.
- Entering the crossroad and blocking the traffic. Has a penalty of 61 TL according to Turkish laws.
- Not obeying the STOP sign on the school buses. Has a penalty of 128 TL according to Turkish laws.

4. PROBLEM DOMAINS

The problem domains can be classified as simulated and physical applications.

4.1. Simulated Applications

Since the framework is developed by using Microsoft's Visual Studio.NET, the simulation environment is Robotics Studio which is also a product of Microsoft company. In addition, an urban challenge proposed by KIA provides a valuable simulation environment for autonomous urban driving. The simulated agent has stereo vision, GPS, bumper detectors and LIDAR sensors. And the actuators are modeled as differential drives for simplicity.

Figure 4.1. KIA Urban Challenge

4.2. Physical Applications

Current physical applications are based on precaptured visual information. However, the goal of the project is performing online image processing in addition to the information from RFID system.

Camera Position:	Front console of the car.
Resolution:	512x288
Frame Rate:	29.97
Length:	34 sec.

Table 4.1. Properties of video sequence.

Pixel Value	Red	Green	Blue
0-176	0	0	0
176-196	1	1	0
196-255	1	1	1

Table 4.2. Color remapping.

4.2.1. Lane detection and tracking experiments

The proposed approach is implemented and tested on a relatively short video sequence of an urban drive. In addition, the new approach is compared with the classical Hough transform where the entire image is processed and the most intense lines are accepted as candidate lines. The properties of the video sequence are given in Table 4.1.

As the first step of the experiment, the image is converted to a binary image by using color remapping. The mapping for each pixel from 24bit RGB value to binary value is given in Table 4.2. This binarization favors the white and yellow parts of the images. The values are manually crafted for the video sample. More discussions about improving the color remapping can be found in the next section. The next step is to determine the partitions of the image on which the Hough transforms will be applied. Although the image is 288 pixels high, only the bottommost 116 pixels are used since the road remains in this lower part of the image. The accuracy of this assumption may slightly differ depending on the slope.

The widths of the partitions are 32, 64, and 128 pixels from top to bottom. And the heights are 32, 42, and 42 pixels respectively as shown in Figure 4.2. These values are assigned according to the position of the camera. After the partitions are calculated, Hough transformation is applied to each partition as described in the previous section.

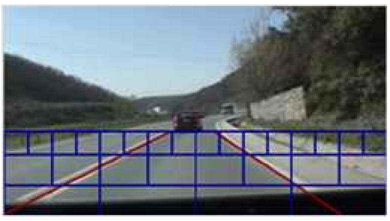

Figure 4.2. Image partitions.

r	0	1	2	...	281	282
0	0.3989	0.2420	0.0540	...	0.0000	0.0000
1	0.2420	0.3989	0.2420	...	0.0000	0.0000
2	0.0540	0.2420	0.3989	...	0.0000	0.0000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
280	0.0000	0.0000	0.0000	...	0.2420	0.0540
281	0.0000	0.0000	0.0000	...	0.3989	0.2420
282	0.0000	0.0000	0.0000	...	0.2420	0.3989

θ	0	1	2	...	178	179
0	0.3989	0.2420	0.0540	...	0.0540	0.2420
1	0.2420	0.3989	0.2420	...	0.0044	0.0540
2	0.0540	0.2420	0.3989	...	0.0001	0.0044
⋮	⋮	⋮	⋮	⋮	⋮	⋮
177	0.0000	0.0000	0.0000	...	0.0000	0.0000
178	0.0000	0.0000	0.0000	...	0.0000	0.0000
179	0.0000	0.0000	0.0000	...	0.0000	0.0000

Table 4.3. (a) Transmission matrix for r, (b) Transmission matrix for θ.

The most promising three lines are assigned as the candidate lane markers. But there may be less than three lines if the intensity of the calculated lines are less than an empirically assigned threshold. The experiment shows that the proposed approach usually detects only two lines most of the time.

After finding the lane markers, HMM method is used to track the lanes. The values of the emission and transition matrices are derived using Gaussian assumption. The deviation of the transition matrix is assigned as 1 and the deviation of the emission matrix is taken as 2. Two separate models are prepared for the θ and r values of the candidate lane markers. The transition and emission matrices are given in Tables 4.3 and 4.4. Since the θ values 0 and 179 are actually very close, the emission and transmission values are the same for 1 and 179 in θ matrices. In addition, the range of the r matrices is (0, 282) because the maximum possible distance for any detected line is 282 pixels where the height of the processed part of the image is 116 and width of the image is 512.

The proposed approach managed to detect and track at least one line in most of the sequence. In addition, false positives are reduced to an acceptable level. In order to validate the results, the proposed approach is compared with the classical Hough

r	0	1	2	...	281	282
0	0.1995	0.1760	0.1210	...	0.0000	0.0000
1	0.1760	0.1995	0.1760	...	0.0000	0.0000
2	0.1210	0.1760	0.1995	...	0.0000	0.0000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
281	0.0000	0.0000	0.0000	...	0.1995	0.1760
282	0.0000	0.0000	0.0000	...	0.1760	0.1995

θ	0	1	2	...	178	179
0	0.1995	0.1760	0.1210	...	0.1210	0.1760
1	0.1760	0.1995	0.1760	...	0.0648	0.1210
2	0.1210	0.1760	0.1995	...	0.0270	0.0648
⋮	⋮	⋮	⋮	⋮	⋮	⋮
178	0.0000	0.0000	0.0000	...	0.0000	0.0000
179	0.0000	0.0000	0.0000	...	0.0000	0.0000

Table 4.4. (a) Emission matrix for r, (b) Emission matrix for θ.

Transform approach. In this method, the same part of the image is processed using the Hough transform routine. The most intensive 10 lines are merged according to their r and θ values. Finally three or less candidate lines are selected as the lane markers. The major differences between the two approaches are shown in Figure 4.3. The images on the left hand side are the detected or missed lines by the classical approach. The right hand side images are the outputs of the new approach for the same frames which show that the new approach is more robust and accurate.

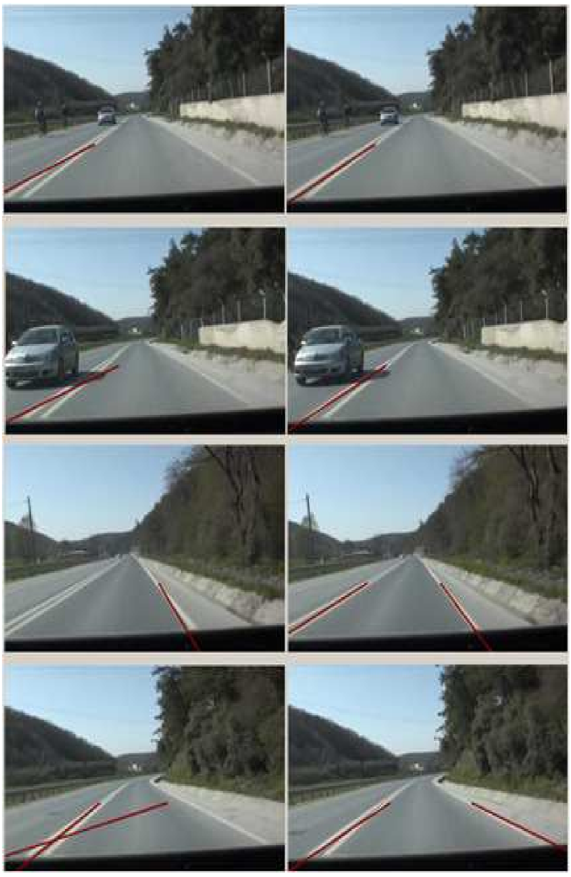

Figure 4.3. Differences between classical Hough transformation and proposed approach.

The computational cost of the proposed approach can be compared as follows. The average processing time is 21.25 milliseconds for a laptop PC with Intel T2050 processor at 1.6 GHz whereas the average cost of the classical approach is 15.29 milliseconds. The results of the experiments for both the proposed approach and the classical method can be found in the project website.

4.2.2. Sign detection and tracking experiments

Similar to the lane detection, sign detection process also starts with a image binarization process. However, this time a function of red, green, and blue values are used instead of a fixed color map as shown in Equation 4.1.

$$f(r, g, b) = \begin{Bmatrix} 1 \rightarrow \alpha.r > g, \beta.r > b \\ 0 \rightarrow o/w \end{Bmatrix} \tag{4.1}$$

where α and β are predefined constants. This function guarantees that the redness of the pixel is higher than other colors which is certainly true for the borders of the considered traffic signs. The original image and the binary image produced from it is given in Figure 4.4.

Figure 4.4. Original and binarized images.

The second step is creating the population for the GA process. The chromosome encoding is based on the coefficients of the geometric transformation matrix as defined in problem statement section. However, for simplicity, only the two translation and one scaling coefficients are included in the chromosome. The resulting transition matrix is given in Equation 4.2.

$$\begin{vmatrix} u \\ v \\ 1 \end{vmatrix} = \begin{vmatrix} a & 0 & c \\ 0 & a & f \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \tag{4.2}$$

And the crossover process is also a function of these coefficient as given in Equation 4.3

$$\begin{aligned} a_{newchromosome} &= \alpha.a_{chromosome1} + \beta.a_{chromosome2} \\ c_{newchromosome} &= \alpha.c_{chromosome1} + \beta.c_{chromosome2} \\ f_{newchromosome} &= \alpha.f_{chromosome1} + \beta.f_{chromosome2} \\ 1 &= \alpha + \beta \end{aligned} \tag{4.3}$$

And the parameters of the GA process are as follows,

- Population Size: 100
- Number of Iterations: 5
- Mutation Rate: 0.05
- Selection Rate: 0.9
- Selection Method: Elitist

In addition to these properties, half of the best chromosomes in a frame is transferred to the next frame in order to provide an initial knowledge about the image. Therefore the number of interations is kept small. And the fitness of the chromosome is evaluated according to the color of the transformed point (u, v) on the binary image. If the value of the pixel is one which means it is a red point on the original image, the fitness of the chromosome is increased. However, this method fails for completely red regions, therefore another set of template points are introduced in order to indicate the non-red points on the template. These points are also subject to the transformation. In this implementation these non-red points are selected to place in the inside of the unit circle. For the example in the problem statement section, if we add the non-red points to the Figure 3.4 we will conclude with Figure 4.5 where the red points increase the fitness values when they are white in the binary image, and the black points increase the fitness value when they are black in the binary image. If the expected color cannot be found then the fitness is decreased for each failed point.

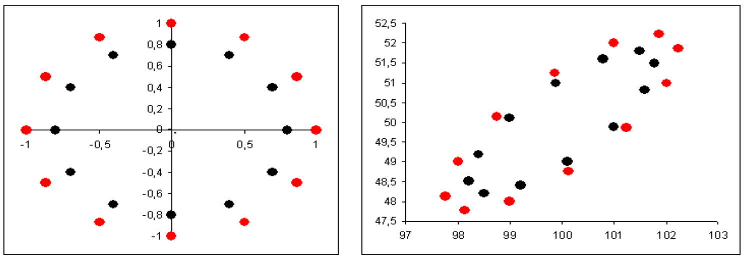

Figure 4.5. Red and non-red template points.

At each iteration the fitness values are calculated for each chromosome and at the end of the process the chromosomes are expected to converge around the circular sign as shown in Figure 4.6

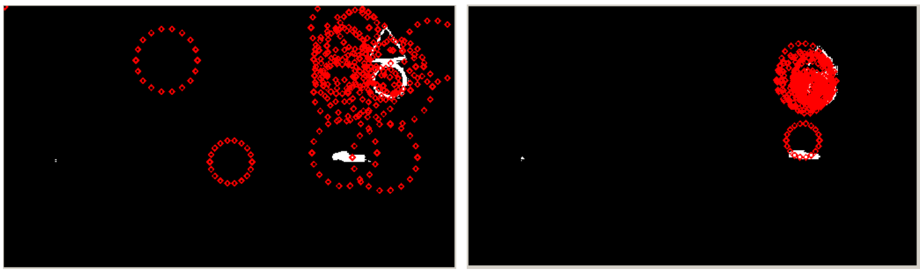

Figure 4.6. Initial and converged chromosomes.

And finally if the best chromosome has a fitness value greater than a threshold value, then the points of this chromosome should be on a circular traffic sign on the original image as shown in Figure 4.7. This complete process takes nearly 50 milliseconds when it is executed on the same hardware specified on the previous section.

Although only circular signs are described in this section, the existing implemen-

Figure 4.7. Detected traffic sign.

tation can also process triangular signs since the procedure is the same except the selection of the template characteristic points. Actually, any kind of sign with some constant properties can be processed by this algorithm.

5. PROJECT SCHEDULE

- **Up to now:**
  - Implementation of two vision processing items.
  - Inspection of reasoning methods.
- **Fall 2009:**
  - RF reader tests and implementation.
  - Implementation of inference engine.
- **Spring 2010:**
  - Tests on physical environment.
  - Journal paper.
  - Thesis writing.

6. CONCLUSION

As it is stated in the introduction, the social and economic damage of the traffic accidents are increasing everyday. Fortunately the number of institutes and efforts to prevent traffic accidents and compensate their damages are also increasing day by day. Most of the countries develop their traffic safety program according to the 4E's of safety which are engineering, enforcement, education, and emergency. Engineering step deals with the conditions of roads and highways, proper settlements of signs and other traffic indicators. Education step is the proper public access to the basic rules and important knowledge about traffic. Emergency part is the planning and improvement of the actions which should be taken after an accident occurs. And the enforcement step adapts the road traffic legislation and the way it is enforced. Since most of the accidents are caused by the drivers, education and enforcement becomes more important for preventing the traffic accidents. The ADES project proposes an application framework to help the enforcement of the rules, where different ideas for detecting and reporting driver faults can be easily combined.

REFERENCES