RC 35

Military Technical College Kobry El-Kobbah, Cairo, Egypt.



15<sup>th</sup> International Conference on Applied Mechanics and Mechanical Engineering.

# VISION BASED SAFE ROAD TRACKING SYSTEM FOR AUTONOMOUS GROUND VEHICLE

T. S. Abd El-Zaher\*, A. M. Bayoumy\*, A. M. Sharaf\* and Y. H. Hossam El-Din\*\*

# ABSTRACT

The past decade has witnessed a rapid evolution of the sensors technology as well as the embedded systems which in turn stimulates the development of a futuristic field such as the autonomous mobile systems. However, the development of an autonomous ground vehicle that is able to sense and avoid obstacles on its way represents a major challenge up till now. Various autonomous ground vehicles have been recently introduced which rely on wide diversity of detection systems such as; laser systems, acoustic systems and machine vision systems.

This paper proposes an advanced algorithm for an autonomous ground vehicle based on lane detection and tracking in addition to obstacle avoidance technique. Through this algorithm, the suitable car heading angle is calculated in order to follow the lane or avoid predefine obstacles. The lane and / or obstacles are being recognized based on vision system in which a real time data are collected by a single camera and analyzed using the color separation technique. An in-house numerical model is developed to estimate both the relative distance and direction of the obstacle. Consequently, further actions are considered to enable the vehicle to drive accurately down a predefined path and at a defined speed. The results reveal both the fidelity and the robustness of the introduced algorithm.

# **KEY WORDS**

Lane detection, Lane tracking, Obstacle detection, Collision avoidance system.

<sup>\*</sup> Egyptian Armed Forces.

<sup>\*\*</sup> Professor, Mech. & Mechatronics Engineering Dept., Higher Technological Institute, 10<sup>th</sup> of Ramadan, Egypt.



### INTRODUCTION

The major challenge for autonomous vehicle or mobile robot is to detect and avoid the encountered obstacles during maneuvers. Over the years, several versions of collision avoidance systems have been proposed varying from simple algorithms that detect an obstacle and stop the vehicle, through smart algorithms that enable the vehicle to steer around the obstacle [1]. The idea of having obstacles conceptually exerting forces onto a mobile robot-named potential field method (PFM), has been suggested by Khatib [2]. In this approach, obstacles exert repulsive forces onto the robot, while the target applies an attractive force to the robot. The sum of all forces specifies the direction and speed of travel. Krogh and Thorpe enhanced this method further by taking other parameters into consideration such as the robot's velocity and global path [3]. It should be noted that, these methods were not implemented on mobile robots with real-time input from sensors. Moravec and Elfes employed multiple wide-angle sonar range measurements to map the surroundings of an autonomous mobile robot [4]. In addition, Borenstein and Koren developed the virtual force field method (VFF) which combines the potential field method with a certainty grid to produce a powerful and robust control scheme for mobile robots [5]. While these methods were initially developed for sonar sensors, they can be applied to current laser sensors as well [6].

Obstacle detection is defined as the determination of whether a given space is free of obstacles for safe travel by an autonomous vehicle. A good obstacle detection system must be capable to correctly detect obstacles on a given space in good time and ignore ground features that may appear as obstacles [7]. Previous work into this field has been carried out using various types of sensors such as vision system, laser rangefinder, ultrasonic sensors, radar etc. Ultrasonic sensors are cheap but suffer from specular reflections and usually from poor angular resolution. Laser rangefinders and radar provide better resolution but are more complex and expensive. Vision systems are cheaper but require a textured environment to perform properly.

In the presented paper, obstacle detection is carried out using vision system with calibrated single camera as an input device such that 'Snap shots' are taken of the real scene and further image processing is carried via PC. Two scenarios are presented; the first is to detect the lane white lines and the second is to detect obstacles such as, for example, traffic cone. Furthermore, the heading angle of the car is predicted and the suitable avoidance consequence is proposed.

#### **KINEMATIC MODEL**

For the purpose of practical implications, the proposed algorithm is considered for a model car (or car like robot) with four rigid wheels as shown in Fig.1. The car is driven through 6-V DC electric motor which is installed at the rear-left wheel of the car without any differentiation between its wheels. The front axle of the car is steered through another 6-V DC electric motor which controls the direction of the car. The mathematical model of the car is simply expressed by its kinematics parameters (x, y, $\theta$ ), the car wheel base (L) and wheel track(d). The position vector

of the car  $(r = [x, y]^T)$  is defined at the center of the rear wheel axle, while the angular movement of the car around its vertical axis is defined by the heading or yaw angle ( $\theta$ ) with respect to the positive direction of the x-axis. The steering angle of the car with respect to its heading direction is denoted by ( $\phi \le 15^\circ$ ). The tangential velocity of the left wheel on rear axle is given by ( $V_L$ ). It should be noted that, the car dynamics is not considered and this may be accepted assuming that the car travels at low speed ( $\le 0.5 \ m/s$ ). It is assumed that all wheels roll without side-slipping. For the steady turning, the turning radius of the car (R) is given by:

$$R = \frac{L}{\tan\phi} \tag{1}$$

The angular rotation of the car about its vertical axis is given by:

$$\Omega = \dot{\theta} = \frac{V_L}{R'} \tag{2}$$

where: 
$$R' = \begin{cases} R - \frac{d}{2} & \phi > 0\\ R + \frac{d}{2} & \phi < 0 \end{cases}$$
(3)

$$R' = \begin{cases} \frac{2L - d \tan \phi}{2 \tan \phi} & \phi > 0\\ \frac{2L + d \tan \phi}{2 \tan \phi} & \phi < 0 \end{cases}$$
(4)

From equations (1-2) and considering that longitudinal speed of the vehicle  $(V = \dot{\theta}R)$ 

$$V = \frac{V_L L}{R' \tan \phi} \tag{5}$$

$$\dot{x} = V\cos\theta = \left(\frac{V_L L}{R'\tan\phi}\right)\cos\theta \tag{6}$$

$$\dot{y} = V \sin \theta = \left(\frac{V_L L}{R' \tan \phi}\right) \sin \theta$$
 (7)

#### **CAMERA CALIBRATION**

For a given camera configuration, the process of camera calibration involves establishing the relationship between 3D scene points and their corresponding 2D image coordinates. Its accuracy involves performing 3D metric measurements from





images. For the presented research, a low cost webcam with 320X240 pixels is employed as a vision sensor to detect obstacles. Further image processing is carried out using MATLAB and Simulink program along with Image Acquisition Toolbox, Image Processing Toolbox, and Video and Image Processing Block set which provide a flexible environment for solving complex imaging problems in a wide range of machine vision. This research is based on a technique of using calibrated cameras to detect obstacles with vision sensor as shown in Fig.2.

(a) Using camera calibration to find distance to pixel ratio for selected camera type.

(b) Using a Camera calibration arrangement with known dimension and compare distance in paper with number of pixels parallel in image.

The following features of calibration target are used:

1- Length D= 83 cm.

- 2- Width W = 67 cm.
- 3- Circle no. 1 with R=1cm.
- 4- Circle no. 2 with R=5 cm.
- 5- Rest of circles with equal space radius= 4 cm.

The calibrated camera is used to shoot several photos of a calibration target for various ranges,  $(H_{cal})$  (100, 145, and 200) cm. Consequently, distance to pixels ratio (E/P) is obtained as shown in Fig.3.

For calibration at  $(H_{cal} = 145)$  the results are shown in Table 1. The best ratio is RX= 0.254 and RY= 0.283. The calibrated distances  $(A_{cal}, B_{cal}, C_{cal})$  and angles  $(\theta x, \theta y)$  as shown in Fig.4 of any point in the image can be easily determined by number of pixels of  $F_1$ ,  $F_2$ , and  $X_{center}$  the center of point in x-direction.

Distances:  

$$Center \begin{bmatrix} C_{cal} = -\frac{D}{2} + \frac{D}{X_{max}} X_{center} & in \ cm \end{bmatrix}$$

$$width \begin{bmatrix} A_{cal} = F_1 \times R_x & in \ cm \end{bmatrix}$$

$$height \begin{bmatrix} B_{cal} = F_2 \times R_y & in \ cm \end{bmatrix}$$

$$(8)$$

$$Height \begin{bmatrix} B_{cal} = F_2 \times R_y & in \ cm \end{bmatrix}$$

$$\{ \begin{bmatrix} \theta_x = \tan^{-1} \frac{C_{cal}}{H_{cal}} \end{bmatrix} \begin{bmatrix} \theta_y = \tan^{-1} \frac{B_{cal}}{H_{cal}} \end{bmatrix} \}$$

$$(9)$$

#### LANE DETECTION

Lanes are normally marked by white lines which can be identified and captured by the robot's camera in 24bit RGB format. Each pixel is broken down into a set of three 8bit values according to the intensity of red, green and blue colors [8]. The captured image's resolution is 320×240 pixels and stored in a 320×240×3 matrix whose values are between 0 and 255. Further images processing is necessary incorporating extraction of the pixels of the lane and eliminating other possible inaccuracies.



## Lane Extraction

During this process, the white lines of the lane are extracted from the image Region of Interest (ROI). In this paper, two methods are implemented and the final result is obtained from their combinations.

#### (a) White Pixels Extraction

The image pixels are eliminated except for the white pixels that have certain values of red, green and blue components. For the presented experimental work, the best result is obtained by extracting pixels with (R>190, G>190 and B>200) as shown in Fig.5 and Fig.6. It should be noted that, the presented technique is simple, requires little processing time and power and provides good results in a one-step process. Furthermore, if the color of the lane marker is changed, the adjustments required in the code are minimal. However, the results are not satisfying under various lighting conditions. In addition, on a sunny day, there will also be a lot of reflections that need to be filtered out. Under low-light conditions, the white lane marker will appear closer to asphalt. This will require the tolerance level to be increased.

### (b) Channel Mixer

Chanel is referred to the image's color component (R, G, B), reducing each pixel to a single value from the original set of three values shrinks the number of operations by two thirds and further simplifies its processing. This process is known as converting to gray scale, as the resulting image pixels will be stored as a single 8bit value, often represented by a gray value between black and white. This value is usually called the luminescence value since it represents how bright the pixel is.

Four methods are used in this paper to calculate the luminescence of each channel as illustrated in Fig.7. The simplest method takes the average of the three color values for each pixel. However, it does not always produce the best result and often lacks in contrast. The PAL television standard uses a ratio that provides the best contrast suitable for human viewing [8]. For the purpose of comparison, Intel's Image Processing Library is introduced [9].Since the asphaltic road has a high blue and green luminescence; the grayscale image can be by refined using a combination of color channels with certain ratio such that both the green and blue channels are subtracted from the red channel in a certain ratio. Experimental results show that the formula (c = R - 0.4(G+B)) provides the best results with minimal calculations involved. Using the mixed channel formula with an average threshold value (0.3922), the image is converted from grayscale into a black-and white monochromatic binary image as shown in Fig.8. This method provides accurate results with minimal computation. Furth more, it is not affected by varying lighting conditions since there are no hard-coded color values and thresholds.

The presented research combines the aforementioned methods to get the best results from comparison between the final B/W binary images and obtaining the final processed image with common white pixels. In the final B/W binary images each image is subtracted from the other and the errors of white pixels. This method has combined the privilege of channel mixer and white pixel extraction and eliminates the white pixel errors resulted from each method alone.



### Image Enhancement and Noise Removal

While the previous step removes most of the noise in the image, further noise such as parts of cars, sky, pavement, and trees need to be removed to make the lane detection process faster and more accurate. For this purpose, ready-made library of functions are used under the MATLAB environment. Using the mixed channel formula with noise, Fig.9 illustrates the grayscale to B/W image conversion. Four steps are followed to obtain the black and white binary image of lanes only without noise as shown in Fig.10.

# Line Analysis

Lane detection involves finding the equation of a straight line that corresponds to the lane marker in the image. The image has two-dimensional binary matrices that can be considered as a set of points in a space. The algorithm attempts to find a heading which has the least resistance. To get the appropriate heading the algorithm goes through the following steps:

- 1- Identify the line equation of the left lane by chose any point in this lane  $(x_L, y_L)$  and the mean point of the rest of points of the lane  $(x_m, y_m)$  and by two points, get the equation of the line and its inclination (m1) as shown in Fig.11.
- 2- Identify the line equation of the right lane as previous with its inclination.  $(x_R, y_R)$   $(x_m, y_m)$  and (m2)
- 3- Determine the car line heading by get the line equation that represent the inclination with value  $(m_1 + m_2)/2$ , and the point that result from the intersection between the left and right lanes.

Since the camera is located at the middle front of the car, the actual car heading is determined as a line divide the captured image at mid-point in x-axis with orientation 90 degrees as shown in Fig.12.

$$\Delta\theta = \tan^{-1} \left( \frac{m_1 + m_2}{2} \right) - 90$$
 (10)

$$\Delta y = (P_2 - P_1) \cdot R_x \tag{11}$$

# **OBSTACLE DETECTION**

The presented paper investigates the detection of a traffic cone as an obstacle example in terms of fixed parameters such as color, height, and width. The distance of an obstacle and orientation, viewed from the camera center, relies on the camera calibration which introduced previously in section 3.

# **Obstacle Extraction**

In this process the traffic cone with a red color is extracted from the image such that the best result is obtained empirically by extracting obstacle pixels with (R>100 & R/G>1.5 & R/B>1.5). The obstacle pixels satisfy these values will take value of one and the rest pixels take value of zero, see Fig.13. Further enhancement of resulted

B/W image is considered by removing noise and smoothing the obstacle to make the demanded processing to determine the relative obstacle distance and direction. So B/W image is filtered with aid of MATLAB toolbox as shown in Fig.14.

#### **Obstacle Analysis**

The obstacle is recognized by its height  $(F_2)$ , width  $(F_1)$  and center (x, y) as shown in Fig.15. By using distance to pixels ratio  $(R_y)$  obtained from camera calibration and defined height in pixels getting the relative height in centimeters  $(B_{cal})$  which lead to angle of obstacle in Y-direction  $(\theta_y)$ .

The real distance from the camera to obstacle and obstacle width, Fig.16 & Fig.17are calculated as follow:

The relative distance from camera to obstacle: 
$$r_{ob} = \frac{\theta}{\tan \theta_y}$$
 (12)

The obstacle width: 
$$A = r_{ob} \left( \frac{A_{cal}}{H_{cal}} \right)$$
 (13)

Obstacle orientation:  $\theta_{ob} = \theta_x$ 

(14)

# SCHEME AND SIMULATION RESULTS

#### Lane Tracking Control

The vision system, image acquisition and image processing, determine the lane heading with respect to car by calculating the relative distance  $\Delta y$  and relative angle  $\Delta \theta$  from the car. To guide the car to track the lane as shown in Fig.18, and the steering angle is calculated as follows:

$$\phi = K_1 \cdot \Delta y + K_2 \cdot \Delta \theta \tag{15}$$

where  $(K_1, K_2)$  are proportional control gain coefficients.

#### **Obstacle Avoidance Control**

whenever the vision system detects an obstacle, it sends a flag to the flag switch to switch the steering angle from lane tracking control to obstacle avoidance control, and the steering angle,  $\phi$ , will follow the following equations:

$$\phi = K_1 \cdot \theta_t = K_1 \cdot \tan^{-1} \left( \frac{R_t}{r_{ob}} \right)$$
(16)

$$R_t = 1.5 \left( \sqrt{L^2 + d^2} + \frac{A}{2} - r_{ob} \cdot \tan \theta_{ob} \right)$$
(17)



After the car avoids the obstacle the flag switch switches to the lane tracking control to guide the car to lane again as shown in Fig.19.

# CONCLUSIONS

This paper presented a new approach to detect and avoid an obstacle while driving a small car like robot. This approach is based on machine vision system in which both lane and obstacles like traffic cone can be recognized. Furthermore, control algorithm is proposed to avoid the prescribed obstacles. It should be noted that the obtained results accuracy is subjected to the in hand sensors such as low cost PC cam. The future works will embrace further enhancements such as the dynamic mathematical model, 3D vision system, and the incorporation of laser range finder sensor.

# REFERENCES

- [1] Modi, S., Comparison of three obstacle avoidance methods for an autonomous guided vehicle. 1998, University of Cincinnati.
- [2] Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. 1985: IEEE.
- [3] Krogh, B. and C. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. 1986: IEEE.
- [4] Moravec, H. and A. Elfes. High resolution maps from wide angle sonar. 1985: IEEE.
- [5] Borenstein, J. and Y. Koren, Real-time obstacle avoidance for fact mobile robots. Systems, Man and Cybernetics, IEEE Transactions on, 1989. **19**(5): p. 1179-1187.
- [6] Hancock, J., M. Hebert, and C. Thorpe. Laser intensity-based obstacle detection. 1998: IEEE.
- [7] Singh, S. and P. Keller. Obstacle detection for high speed autonomous navigation. 1991: IEEE.
- [8] Florczyk, S., Robot vision. 2005: Wiley Online Library.
- [9] Intel Corp., Image Processing Library Reference Manual, 200, Available: <u>http://www.cc.gatech.edu/dvfx/readings/iplman.pdf</u>.

RC 43



Fig. 1. Kinematic model of the car.



Fig. 2. Camera calibration arrangement.



Fig. 3. The distance to pixels ratio (D/P).





Fig. 4. the calibrated distances and angles.



Fig. 5. Color component values for pixels at various areas of the image



Fig. 6. Image converted to B/W binary image by filtering white pixels.





(a) Average color value (c = 0.33R + 0.33G + 0.33B)



(c) Intel formula (c = 0.212671R + 0.715160G + 0.072169B)



(b) NTSC formula (c = 0.30R + 0.59G + 0.11B)



(d) Mixed channel formula (c = R - 0.4(G+B))

Fig. 7. Image converted to grayscale image using various methods.





Fig. 8. Image converted to B/W binary image after converting to grayscale.





Fig. 9. the grayscale to B/W image conversion using the mixed channel formula with noise.



Step (1) Remove all connected objects that have fewer than certain pixels.



Step (2) Join the white pixels in the image together by filling in the gaps between them and by smoothing their outer edges to create separate objects.



Step (3) Clear the image borders.



Step (4) Measure the eccentricity of the ellipse for each connected object to get lanes only.





Fig. 11. Mathematical equation of the lane lines.



Fig. 12. the actual car heading and calculated heading.





Fig. 13. the original image converts to B/W image has traffic cone with white pixels.



Fig. 14. the B/W image after enhancement.



Fig. 15. Obstacle detection.



Fig. 16. Car-obstacle layout.



Fig. 17. Car-obstacle avoidance orientation



Fig. 18. Lane tracking simulation results.



Fig. 19. Obstacle avoidance simulation results as obstacle in left, middle and right of the car.

| No. | Е    | Pixels         |                | E/Pixels       |                |
|-----|------|----------------|----------------|----------------|----------------|
|     | (cm) | In X direction | In Y direction | R <sub>x</sub> | R <sub>Y</sub> |
| 1   | 10   | 39             | 35             | 0.256          | 0.286          |
| 2   | 18   | 69             | 62             | 0.261          | 0.290          |
| 3   | 26   | 99             | 91             | 0.263          | 0.286          |
| 4   | 66   | 252            | 233            | 0.262          | 0.283          |

**Table 1.** Results of distance to pixels ratio (E/P).

