15th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 15 – May 28 - 30, 2013, Email: asat@mtc.edu.eg ,
Military Technical College, Kobry Elkobbah, Cairo, Egypt,
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

Vision-Based Road Tracking of Wheeled Mobile Robot

M. AbdElrahman[*], A. M. Bayoumy[†], G. Elbayoumi[‡], M. Bayoumi[§]

Abstract: Wheeled mobile robot (WMR) is not a new subject but it has gained interest in recent years due to its high applicability in civil industries and military field. Autonomous robots can perform desired tasks in unstructured environments without continuous human guidance. A fully autonomous robot has the ability to gain information about the environment and take decision according to this information independently. Vision is the human most powerful sense. In this paper, a vision-based lane detection algorithm is presented for tracking the road for autonomous wheeled mobile robot. The presented algorithm relays on feature based detection which detects the interested pixels by a mixture between image thresholding and edge detection. This mixture gives advantages of execution time reduction, overcomes shadow and dark areas problem and works with different day night time. Hough transform is used for line detection. The required heading angle is calculated by using these detected lines after the adjustment of the path following control law. This angle is used for controlling the WMR. Decision trees are utilized to increase the robust of robot behavior and have the benefit of solving the breaks in marking lines problem. The lane detection algorithm along with the decision trees are tested experimentally and they give high accuracy in different environment conditions. Finally at controller design step, WMR mathematical kinematic molding is hired to characterize the controller gains.

Keywords: Wheeled mobile robot, lane detection, Hough transform, vision system, path-following.

1. Introduction

Autonomous robots can perform desired tasks in unstructured environments without continuous human guidance. Many kinds of robots have some degree of autonomy. Different robots can be controlled autonomously in different ways. A high degree of autonomy is particularly desirable in fields such as space exploration, cleaning floors, mowing lawns, and waste water treatment [1].

One important area of robotics research is to enable the robot to cope with its environment whether this will be on land, underwater, in the air, underground, or in space.

A completely autonomous robot would require all of the autonomous technologies such as lane detection to aid in passing slower vehicles, obstacle detection to locate other robots,

[*] Egyptian Armed Forces, Egypt, fastmada@yahoo.com

[†] Egyptian Armed Forces, Egypt, dramgad@mep-ls.com

[‡] Head of Aerospace Department, Faculty of Engineering, Cairo University, Egypt, Egyptgelbayoumi@yahoo.com

[§] Associate Professor, Faculty of Engineering, Cairo University, Egypt, msb0100@gmail.com

pedestrian, animals, etc., collision avoidance to avoid hitting obstacles in the way, and lateral control to maintain the robot's position on the roadway. So, sensors will be a major component to develop these technologies.

The main properties that must be possessed by a solution of the lane marking or boundary detection are [3]:

- The detection should not be affected under shadow condition. These shadows can be cast by trees, buildings, etc.
- It should be capable of detecting and processing painted or unpainted roads.
- It should handle curved road rather than assuming straight road.
- It should use the parallel constraint as a way to improve the detection of both sides of lane markings or boundaries in the face of noise in the images.
- It should produce an explicit measure of the reliability of the result it has produced.

A distinction can be made between the problems of lane detection and lane tracking. Lane detection involves determining the location of the lane boundaries in a single image without strong prior knowledge regarding the lane position. Lane tracking involves determining the location of the lane boundaries in a sequence of consecutive images, using information about the lane location in previous images in the sequence to constrain the probable lane location in the current image.

The task of detecting and tracking the lane marking is of great importance of several reasons:

- It can be used to estimate the direction in which the robot is heading.
- It shows the position of the robot in the road.
- It tells how wide the road is, in terms of number of lanes and the road physical geometry.

In principle, vision-based lane detection can be categorized in three main classes: feature-based, region-based and model-based.

At present many different vision-based lane detection algorithms have been developed. They depend on different road models (2D or 3D, straight or curve) and different techniques (Hough, template matching, neural networks, etc.). Some of these approaches:

Tang-Hsien Chang [2] proposes algorithm using image processing techniques for lane detection (adaptive thresholding, defining ROI, noise filters, detecting lines, etc.). Template matching is utilized for obstacle avoidance. The radial basis probability network is used for solving the scenario of many uncertainties.

A new Catmull-Rom spline based lane model describes the perspective effect of parallel lines is constructed for generic lane boundary or marking. It is able to describe a wider range of lane structures than other lane models such as straight and parabolic models. The lane detection problem is formulated by determining the set of lane model control points. A maximum likelihood method is used to measure the matching between the model and the real edge image[3].

Shengyan Zhou [4] introduces another lane detection method by combining lane geometrical feature and lane model matching method. This algorithm divided into three stages. The first one is the off-line calibration which transforming the model to the image plane using the calibration parameters calculated at [5]. The second stage is lane parameters estimation and candidates some parameters where the algorithm uses Hough transform and Canny edge detection methods. The last stage is finding the remaining parameter and applying the matched model. But this algorithm based on planner and pained roads assumptions.

The LOIS (Likelihood of Image Shape) lane tracking algorithm has been shown to robustly find lane markers even in the presence of shadowing, occlusion, and varied lighting conditions. The LOIS algorithm provides a parametric family of shapes for the objects in the image and a function measuring the matching between object in images and real world. Then the algorithm finds the shape parameters which maximize the likelihood function. LOIS used to find the vehicle's offset with respect to the right and left lane markings and examined as a function of time. A Kalman filter is used to predict the future values of these offset parameters, based on past observations [6].

Wei Liu [7] introduces an algorithm which allows detection and tracking of multiple lane markings. Edge points cue is used to detect the lane marking. A road orientation estimation method is used to delete the edge lines which are impossible attribute to lane markings. In order to select the candidate lane marking, a Confidence Measures method is proposed. A finite-state machine decides whether or not a lane marking is really detected by fusion multi-frame detection results. Specifically, a particle filter is used to predict the future values of the lane marking model parameters, based on past observations.

The presented road tracking algorithm is composed of three parts: image pre-processing, lane detection and heading determination.

2. Road Tracking Algorithm

The algorithm described in this paper is quite unique as it uses a mixture of image thresholding and edge detection then using Hough transform as a feature extraction steps.
A COMS camera is fixed on the front of the experimental car to capture the road scene. The algorithm first is down-sampling the image, converts the image to a grayscale. Due to non-uniform lighting distribution, image contrast enhancement is applied. Median filter is utilized for noise reduction. The image is split into four quarters. Every quarter of the image is submitted to a line extraction process. In the feature extraction process a mixture between image thresholding and edge detection is used to produce an edged image. At this step the output is typically a small set of data (small number of pixels). This data is sent to the line detector for detecting the lane boundary segments. These segments are submitted to decision trees for providing the proper heading according to the robot behavior, as shown in Fig. 1.

2.1 Image Pre-Processing

This part includes a group of steps that process the acquired data to assure certain assumptions. The loop execution time is reduced by down-sampling the image to quarter the original image then converting to grayscale. The following formula is used for converting each pixel to grayscale.

$$\mathrm{Gray} = 0.299\,\mathrm{R} + 0.27\,\mathrm{G} + 0.431\,\mathrm{B} \qquad (1)$$

where (R, G, B) are the red, green and blue pixel value respectively. Several experiments are performed to obtain this formula which gives a more accurate representation of the original image, especially in image contrast and overall brightness. These values are obtained experimentally.

Second, the image subjectively improved by modifying the intensity to make the required feature (lane lines) easier to see. Then the image is split into four quarters. The remaining road tracking algorithm steps are implemented to each quarter. The curvature estimation problem is handled by image down sampling and image splitting by transforming the curvature lane to piecewise lines.

As presence of noise has an effect on edge detection and image thresholding, noise removal is very important. The median filter used because of its sensitivity and the ability to remove the outliers without reducing the sharpness of the image, see Fig. 2.

2.2 Lane Detection

The steps of image pre-processing prepare the image to the image processing phase (lane detection phase). During the line extraction process, the equation of the most dominant straight line in an image is calculated. In image processing literature; before using Hough transform for line extraction, the image will be subjected to edge detection or thresholding to convert to binary image. But in the present algorithm a mixture of the two methods are used.

As a first, image adaptive thresholding where a simple approach was implementing an intensity threshold. The adaptive thresholding determines the threshold automatically for each individual frame rather than imposing a certain fixed threshold where brighter pixels are considered as a part of a line and darker pixels are free space. The intensity threshold method assumes that the painted line will generally be the brightest part of the image, which is an acceptable assumption as the lane line color almost, be white or yellow.

In this step, the algorithm will search for the edges. Canny method is used as it is one of the most widely used methods and may be their most accurate one. Canny method is used where the texture of bright intensity will be extracted. If there is a dark object (such as a dark car), it will be eliminated in the previous step. The algorithm will not notice that there is a car at the current step.

One of the biggest problems in lane detection is the existence of shadow areas. So using the edge detection only, the boundary of the shadow will appear as edges. But a mixture with an adaptive thresholding, the shadow and dark area problem is handled in the present work.

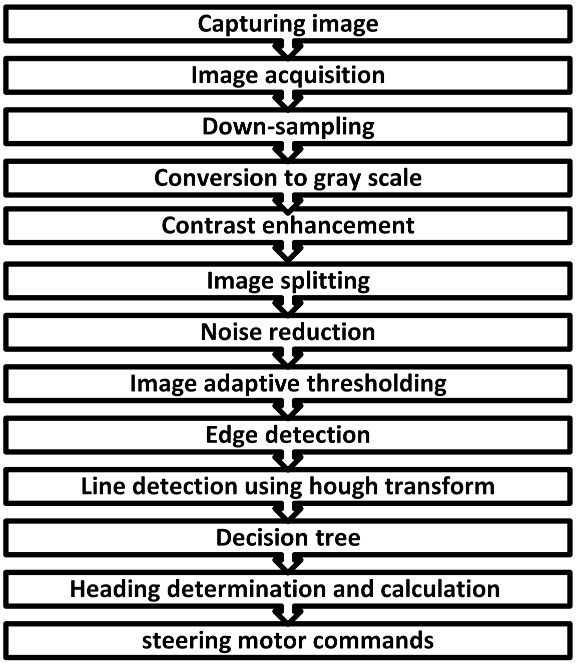


Fig. 1 Road tracking algorithm structure

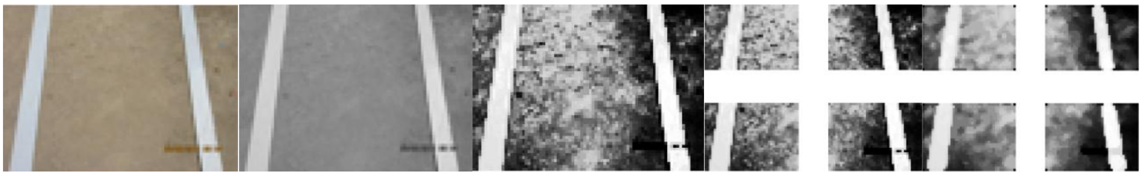


Fig. 2 Image pre-processing steps

Finally Hough transform is used to find the most dominant line in each part with a thresholding score of the number of pixels that is producing a line, see Fig. 3.

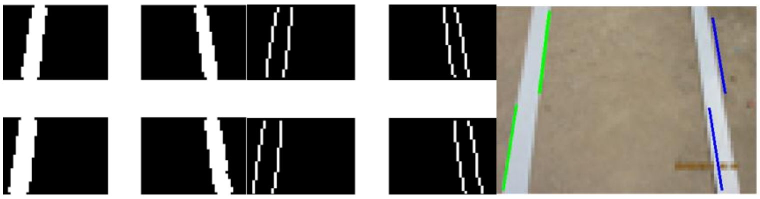


Fig. 3 Lane detection steps

2.3 Heading Determination

The line information is broken down into cases that are solved using distinct methods. This strategy causes the vehicle to execute different behaviors depending on the characteristics of the lines. Changing the vehicle behavior for different situations solved many performance problems.

Once the boundary lane lines have been extracted, a decision tree is used to interpret the situation and selects the appropriate heading line behavior. Each behavior interprets the line information in a unique way and produces a heading. Two decision trees are used, one for the two lower parts and the other for the two upper parts. These decision trees are a case study, created and modified during experiments. Figure 4 represents the decision tree for the lower parts and Fig. 5 represents the decision tree for the upper parts.

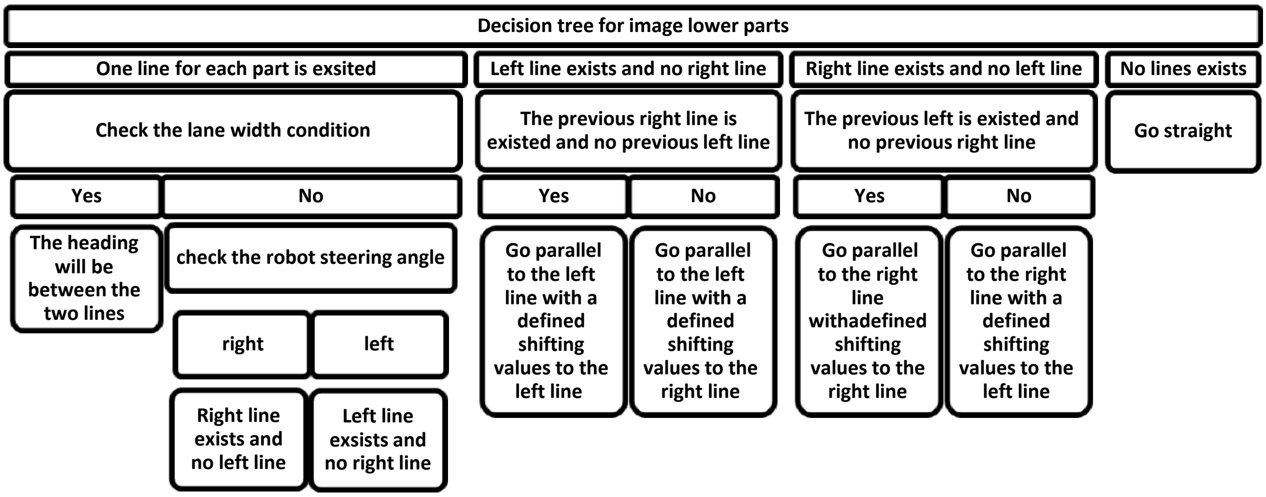

Decision tree for image lower parts
One line for each part is exsited
Left line exists and no right line
Right line exists and no left line
No lines exists
Check the lane width condition
The previous right line is existed and no previous left line
The previous left is existed and no previous right line
Go straight
Yes
No
Yes
No
Yes
No
The heading will be between the two lines
check the robot steering angle
right
left
Right line exists and no left line
Left line exsists and no right line
Go parallel to the left line with a defined shifting values to the left line
Go parallel to the left line with a defined shifting values to the right line
Go parallel to the right line withadefined shifting values to the right line
Go parallel to the right line with a defined shifting values to the left line

Fig. 4 Decision tree for the lower parts of the image

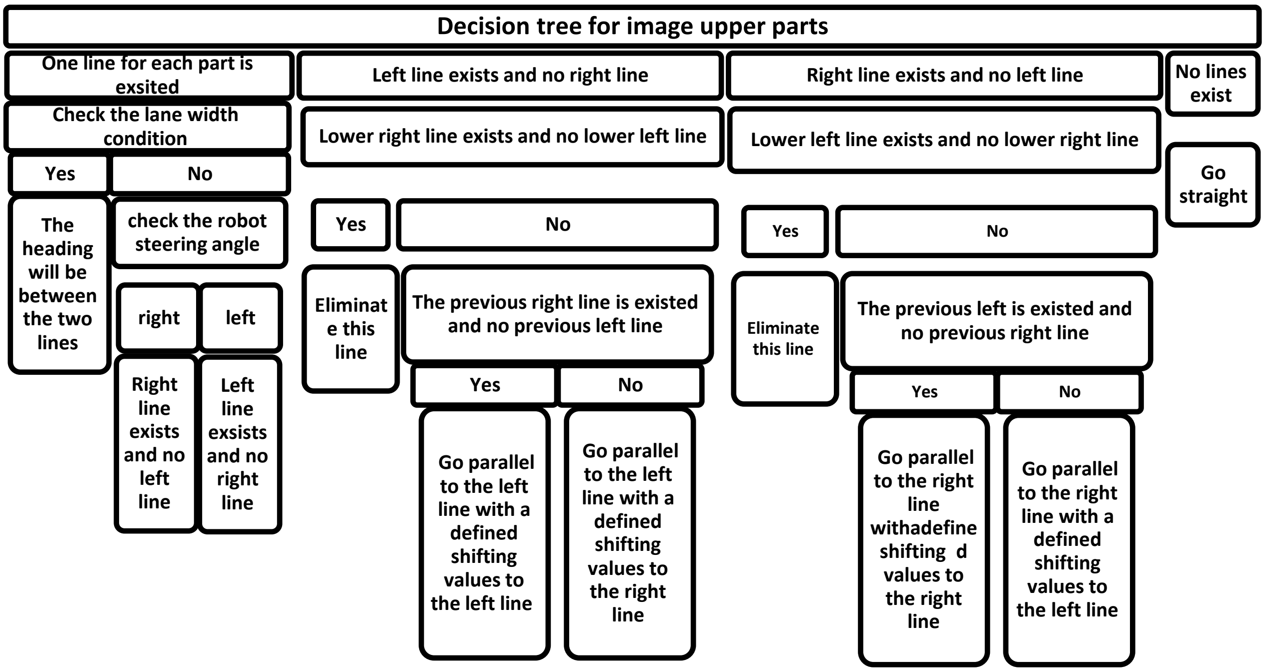

Decision tree for image upper parts
One line for each part is exsited
Left line exists and no right line
Right line exists and no left line
No lines exist
Check the lane width condition
Lower right line exists and no lower left line
Lower left line exists and no lower right line
Go straight
Yes
No
Yes
No
Yes
No
The heading will be between the two lines
check the robot steering angle
right
left
Right line exists and no left line
Left line exsists and no right line
Eliminate this line
The previous right line is existed and no previous left line
Yes
No
Go parallel to the left line with a defined shifting values to the left line
Go parallel to the left line with a defined shifting values to the right line
Eliminate this line
The previous left is existed and no previous right line
Yes
No
Go parallel to the right line withadefine shifting d values to the right line
Go parallel to the right line with a defined shifting values to the left line

Fig. 5 Decision tree for the upper parts of the image

The first decision factor examines whether there is a line or not. If both parts have a line, then the distance between the two lines are measured for checking the lane width condition. This condition is checked to solve a problem of false detection that comes from objects around lane course or non-uniform lighting. If the condition is false, the robot steering angle will be checked. If the robot is turned to left, the right detected line will not be taken into account in the heading calculation and vice versa.

If only one line is detected in the image, the decision tree will check if the robot is still within the lane boundary or not. This check is vital when the robot could not take the proper action in the last step and it may be about leaving outward the lane boundary or the robot is approaching a curvature (e.g. left curvature).

If this lane line appears in the left part at the present step and in the previous step there was a single line appearing in the right part, then the decision will be to move the robot parallel to the detected line but in the left side. But if the condition is false, the robot heading will be parallel to the line but in the right side

If no line exists, it means the robot still within the lane and the decision is moving forward as shown in Fig. 6.

For the two upper parts, they will have the same decision tree of the two lower parts. But one difference exists, see Fig. 6, to check if the detected line appears in the upper left part and the right lower part or vice versa.

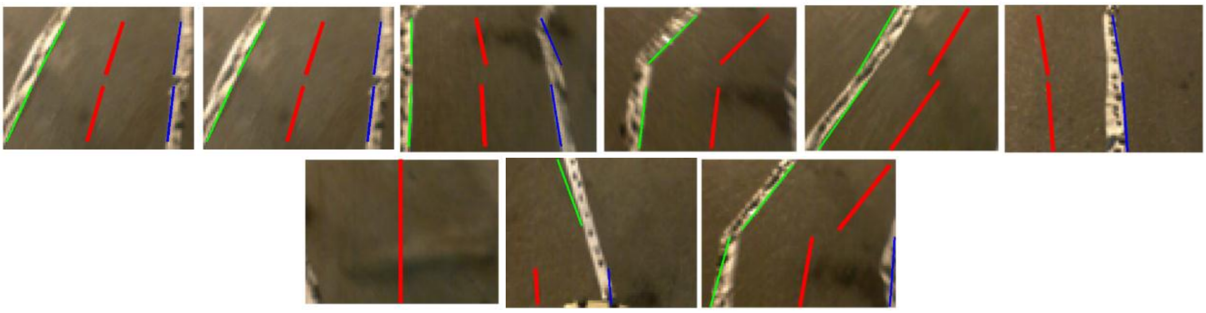


Fig. 6 Examples of the algorithm behavior after passing thorough the decision trees. (As a reference the green is the left detected line, the blue line is the right detected line and red line is the calculated required robot heading)

This condition is checked because that line may be an extension to the detected line in the opposite lower part (curved lane line). In case of true condition, the heading of upper part will not be taken into account in heading calculation step (Fig. 6 the most right image). But if the condition is false, the decision tree will follow the same conditions that are already checked of the lower parts. Figure 7 represents real experiment with the proposed algorithm.

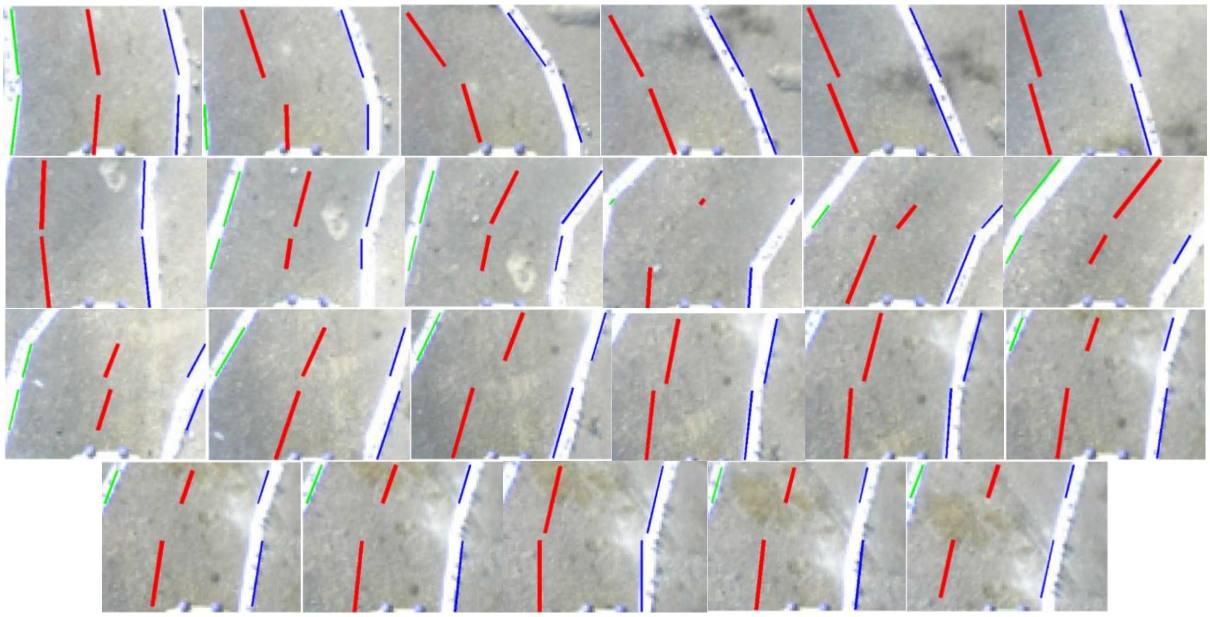


Fig. 7 Real experiment with the proposed algorithm

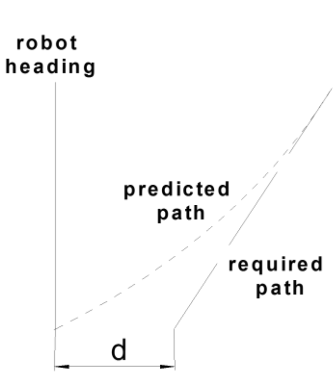
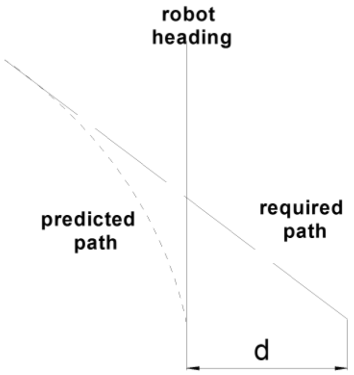
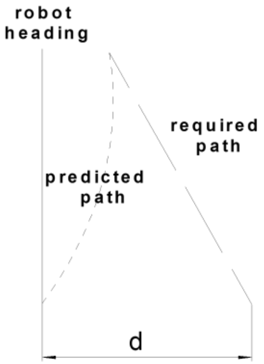
The desired heading angle will be calculated as the average slopes of the two slope's lines. But if only one line exists, the desired heading slope will be equal the line's slope. In case of no line exists, the robot should move forward with zero heading.

In lane detection literature, in most cases the lower part is only analyzed for reducing the execution time. But the present algorithm uses also the upper part of the image as a prediction guide of the required heading angle. It has an effect on reducing the control effort. For example, if the robot needs to turn right 10 degrees and at the next step it needs to turn left 5 degrees. Using only the lower part output analysis the robot will turn right 10 degrees then 15 degrees left at the next step. But in the present algorithm, where the upper part of the image is taken into consideration the total heading angle is calculated as 80 percent of the calculated heading angle in the lower part plus 20 percent of the calculated heading angle in the upper part. The control effort will be 7 degrees right then the 12 degrees left. So the robot will perform 19 degrees in total instead of 25 degrees which leads to energy saving.

At this step the required heading angle will be adjusted by the path following law as shown in Table 1**Error! Reference source not found.**. This adjusted value is calculated as a gain multiplied by the distance between the longitudinal axes of the WMR and the tangent of the detected line (cross track error), see Fig. 8.

The longitudinal axe of the WMR is considered as the vertical line passing through the image center. The image center point (principal point) is an intrinsic parameter. This parameter is measured using a self-calibration technique [5].

Table 1 Path following equations

 robot heading predicted path required path d $\theta_r = \theta_d + 0.23d$	 robot heading predicted path required path d $\theta_r = \theta_d - 0.08d$	 robot heading required path predicted path d $\theta_r = \theta_d - 0.16d$

3. Mathematical Modeling of WMR

In this section the mathematical modeling of WMR is presented, Fig. 9. The WMR can be represented by kinematic model or dynamic model. In robotics field with small robot velocity and consequently small acceleration and small forces; the kinematic model is good enough to represent the WMR motion [8].

The experimental car has two motors: front steering motor and rear left wheel driving motor.

The robot angular velocity will be

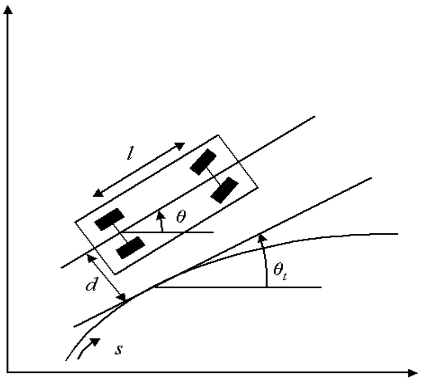


Fig. 8 The actual robot heading and the required path

In left turn

$$\dot{\theta} = \Omega = \frac{v_l}{R - {}^{b}\!/_{2}} = \frac{v_l}{\left({}^{l}\!/_{\tan\varphi} \right) - {}^{b}\!/_{2}} \tag{2}$$

In right turn

$$\dot{\theta} = \Omega = \frac{v_l}{R + {}^{b}\!/_{2}} = \frac{v_l}{\left({}^{l}\!/_{\tan\varphi} \right) + {}^{b}\!/_{2}} \tag{3}$$

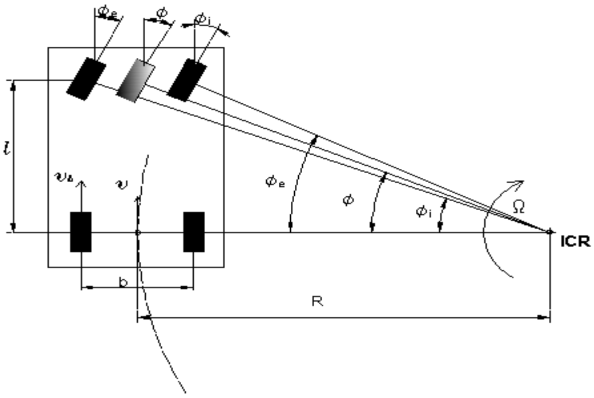


Fig. 9 WMR dimensions

where: v_l is the longitudinal velocity of the rear left wheel, b is the lateral distance between the rear wheels, l is the distance between the two axels, R is the robot radius of rotation, $\dot{\theta}(\Omega)$ is the robot angular velocity and φ is the steering angle. The steering DC motor transfer function is represented as first order $\frac{\dot{\phi}}{V} = \frac{K}{Ts+1}$ where k is the DC motor gain. The motor is powered for certain time and the actual change in the steering angle is measured. This difference is divided by the time to give the motor speed. This procedure is repeated for different values of input voltage (V). The slope will be the motor gain (K).

T is the motor's mechanical time constant (the electrical time constant is ignored in this practical case i.e. the armature inductance L is considered zero), and can be measured for certain applied volt by measuring motor speed change with time.

4. Controller Design

The mathematical model of the robot is used after linearization with the transfer function of the motor as a plant transfer function, see Fig. 10. The plant has two integrators, one in WMR kinematic model transfer function and the other with the DC motor transfer function. PD controller is utilized to add zero to attract the poles in the left side of the S domain. The PD controller is tuned by genetic algorithm as shown in Fig. 11.

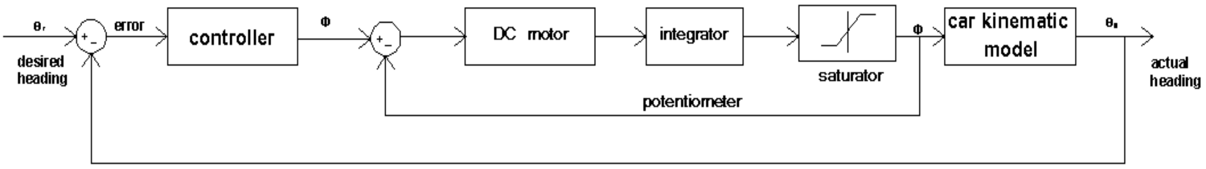


Fig. 10 Simulink model of WMR and DC motor models inside the control loop

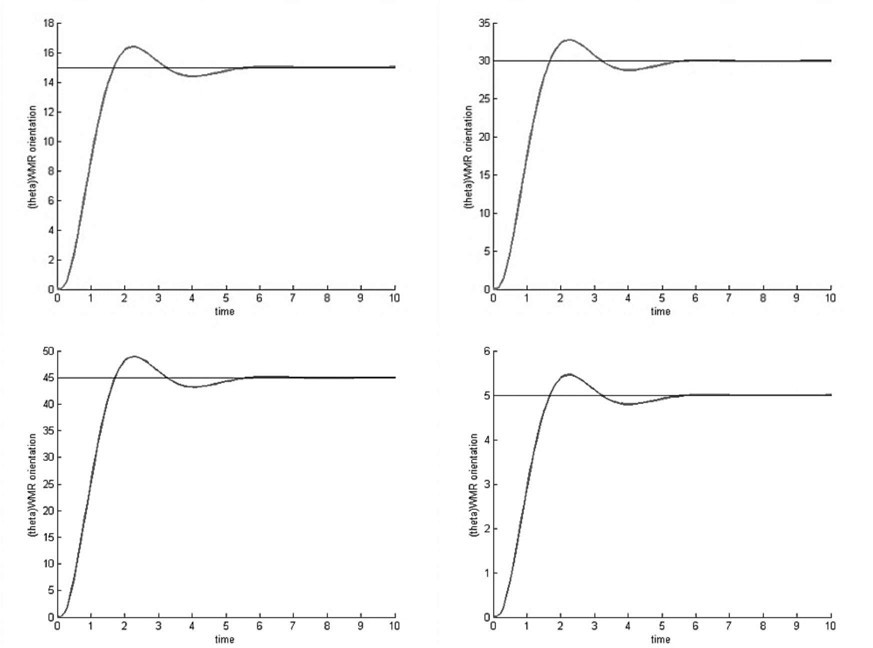


Fig. 11 WMR responses at different degrees step input with PD controller

5. Experimental Results and Analysis

The potentiometer is used as a feedback sensor for the steering angle measurement. According to the Ackermann steering condition, the normal lines of the wheel planes will be intersected at ICR, see Fig. 9. But in practical case this condition is achieved at one angle only[9]. For a specific steering angle, the left and right front wheel steering angles are measured. The robot radius of rotation of each angle is calculated using equations (4) and (5).

$$R_{ex} = \frac{\tan(\ \phi_{\mathrm{ex}})}{l} - \frac{d}{2} \tag{4}$$

$$R_{in} = \frac{\tan(\ \phi_{\mathrm{in}})}{l} + \frac{d}{2} \tag{5}$$

$$\phi = \tan^{-1}(l\ /\ R_{av}) \tag{6}$$

The average robot radius of rotation of the left and right wheel is used to recalculate the virtual center steering angle equation(6). The above steps are repeated several times at several angles and plotted against the measured potentiometer’s count as shown in Fig. 12.

Camera height should be selected to give the best forward view of the robot relaying on the camera field of view angles and the required observed horizontal distance, Fig. 13. The camera field of view (FoV) angles are measured using a calibrating object; Fig. 14, with known coordinates using the following equation

$$FoV\,angle = 2\,\tan^{-1}\left(\frac{\mathrm{s}/2}{\mathrm{D}}\right) \tag{7}$$

where: (D) is the distance between the calibrating object and camera and (s) is the real distance between the two observed lines at image border.

The measured horizontal FoV angle is 36.8 degrees and the vertical FoV angle is 28 degrees. The required observed horizontal width is 160 cm. This width can be achieved at camera height (2.4 m) with zero camera inclination angles. Camera height could be reduced by increasing the inclination angle (equation 8). But the distance from the first observed horizontal line to the robot will be increased (Table 2).

$$h = \cos(\theta - \phi\,/\,2)L = \cos(\theta - \phi\,/\,2)b_l\,/\;\;2\tan(\beta\,/\,2) \tag{8}$$

where: (h) is the camera height, (θ) the camera inclination angle measured from the camera z axis to the vertical line, (Φ) the vertical camera field of view angle, (β) the horizontal camera field of view angle, (W) the rear horizontal width line of the camera field of view projected on the ground and (b_l) is the required width.

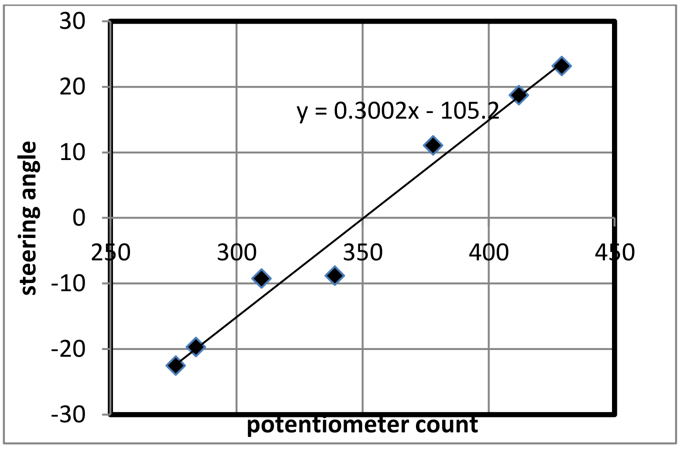


Fig. 12 Virtual steering angle and potentiometer count relationship

Due to increase in the inclination angle, the distance from the robot is increased. The decision of the autonomous robot is based on the future rather than the current situation (this will represent a problem in curved road tracking). So after this distance, the steering command should be carried out. But during maneuver in high curvature lane (low radius), the lanes may be not covered by the camera field of view, see Fig. 15. So the robot will build its decision relaying on unwanted scene (may be with no lane line).

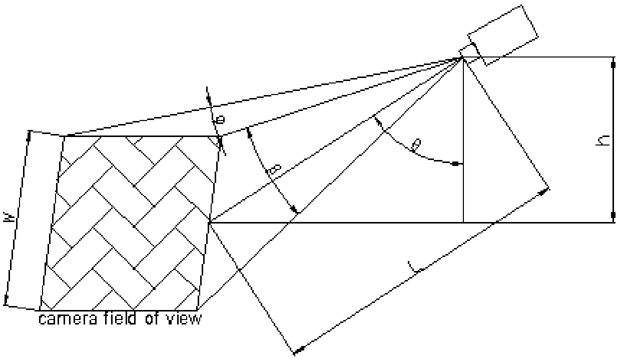


Fig. 13 Field of view parameters

The above problem will be solved by increasing the camera height, so the unseen forward distance will be decreased. The robot can take quickly the proper action even in high curvature lane area , see Fig. 16, The selected height is h=250 cm with 8 degrees inclination angle

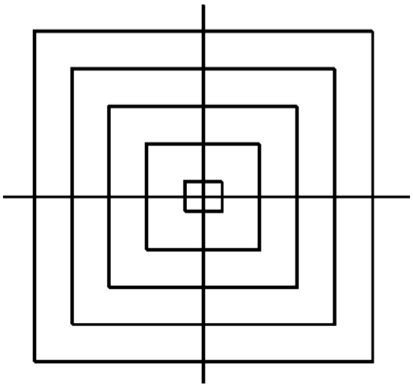


Fig. 14 Calibration object

Table 2 Height and distance results for different inclination angles

θ (degrees)	h (cm)	d=tan(θ) h (cm)
28	233.3	58.1
35	224.5	86.1
45	206	123.7
60	167	172.9
70	134.4	199.2

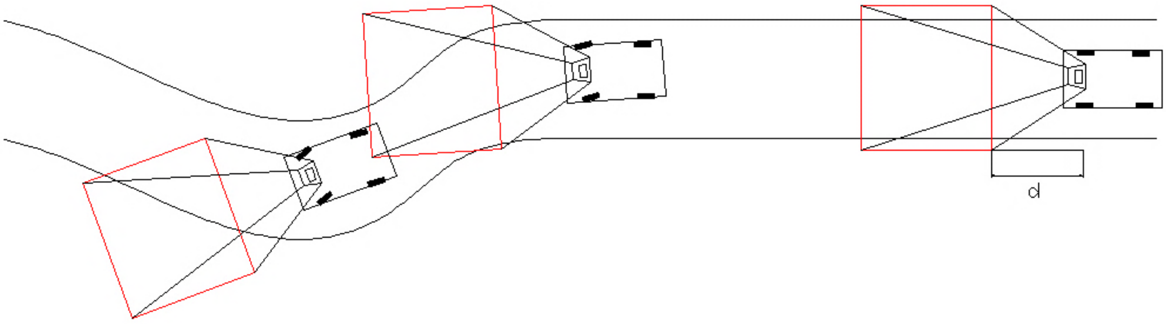


Fig. 15 High inclination field of view angle

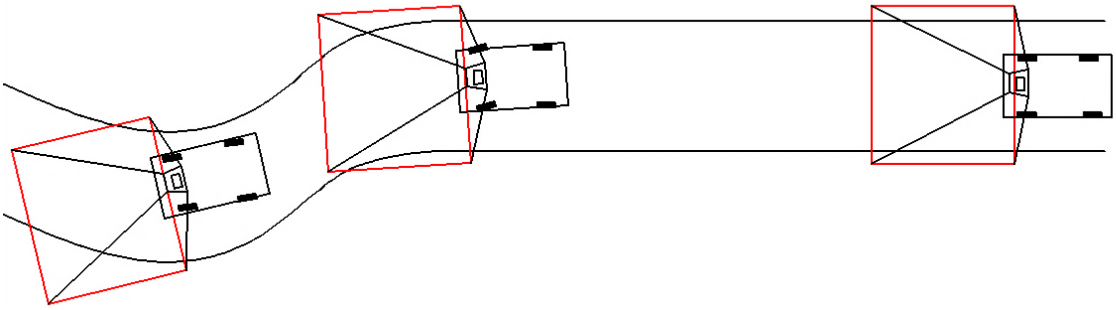


Fig. 16 Low inclination field of view angle

The experimental results in this paper, the vision algorithm was implemented in a HP Compaq nx7300, Intel(R) Core (TM) Duo CPUT2450 @2.00GHz and RAM: 2.99 GB using Matlab version 7.10.0.0(R2012a). All video was recorded at normal roads, dashed markings, straight and curved roads in different environmental conditions (sunny, cloudy, night time, and shadowing). There are numerous results of performance interest, relating to execution time results of the vision model is shown in Table 3, and vision model accuracy is shown in Table 4.

Table 3 Vision algorithms execution time

No	Algorithm features			
	Conversion to Gray	Down-sampling	Image part	Execution Time
1	Without	Without	whole	0.67 sec
2	With	Without	lower part	0.10 sec
3	With	With	lower part	0.04 sec
4	With	Without	whole	0.22 sec
5	With	With	whole	0.062 sec

Table 4 Lane detection accuracy during different times with different road types

Road type	Accuracy			
	Cloudy Day	Night	Shinny day	shadow
Continuous marking	95 %	80 %	88 %	92 %
Breaks in marking	85 %	65 %	80 %	85 %

Figure 17 represents a part of pavement used in experimental during test. Without any modification on the present vision algorithm, the robot can handle the discontinuity in road marks and track it efficiently; see Fig. 18.

6. Conclusion

In this paper, a road tracking algorithm was implemented for WMR for guidance based on video sequences taken from a robot driving on a pavement way. This video sequence is discretized individually and analyzed by utilizing the image processing algorithm that has been described.

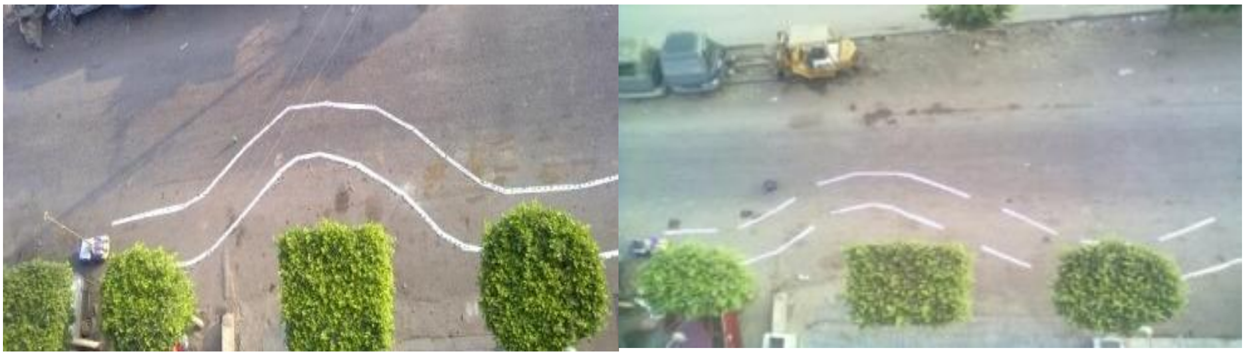


Fig. 17 Experimental routes

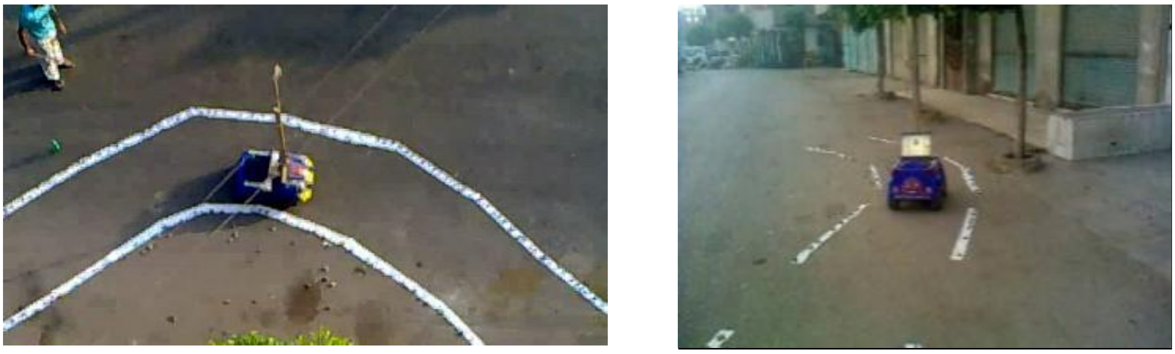


Fig. 18 Experimental car during tests

The mixture of adaptive thresholding and Canny edge detector gives advantages of execution time reduction, overcomes shadow and dark areas problem and works with different day night times. The lanes are detected using Hough transformation.

Decision trees are utilized to increase the robustness behavior of the robot and have a benefit of solving the breaks problem in marking lines. The lane detection with the decision trees is tested experimentally and they give high accuracy in different environment conditions.

The heading angle is calculated depending on the whole image. The upper parts is used as a predication step which gives the benefit of power saving. Then the heading angle is adjusted by the path following control law. The wheeled mobile robot (WMR) modeling was accomplished by the kinematic model as the kinematic model is adequate in case of low speed WMR.

The camera height is selected according the derived generic equation which depends on the camera field of view angles and the required observed width (road width) and tested successfully.

7. References

[1] http://en.wikipedia.org/wiki/Autonomous_robot .

[2] C.H.Tang, H.L.Chun, S.H.Chih, J.W.Yao, A Vision-Based Vehicle Behavior Monitoring and Warning System, IEEE, 2003.

[3] W.Yue, S. Dinggang, K.T. Eam, Lane Detection Using Catmull-Rom Spline, IEEE International Conference on Intelligent Vehicles, 1998.

[4] Z.Shengyan, X.Junqiang, G.Jianwei, X. Guangming, C. Huiyan, A Novel Lane Detection based on Geometrical Model and Gabor Filter, 2010.

[5] Z.Zhengyou, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11):1330–1334.

[6] K.Chris, L.Sridhar, K.Karl Kluge, A Driver Warning System Based on the LOIS Lane Detection Algorithm, 1998.

[7] L.Wei, Z.Hongliang, D.Bobo, Y.Huai, Z.Hong Zhao, Vision-Based Real-Time Lane Marking Detection and Tracking, IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008.

[8] P.Romain Pepy, L.Alain Lambert, M.Hugues Mounier, Path Planning using a Dynamic Vehicle Model, IEEE, 2006.

[9] R.Danny, M.Phillip Mckerrow, Dynamics of the Titan four-wheel drive mobile robot with floating Ackerman steering, Australia, 1999.