A Real-Time Lane Extraction System for the Wheelchair Robot

Xiaojun Qi and Yinbing Ge

Computer Science Department, Utah State University, Logan, UT 84322-4205
xqi@cc.usu.edu and yinbing@cc.usu.edu

Abstract

This paper presents a robust and reliable lane extraction algorithm to navigate the wheelchair robot in real time in outdoor environments. Seven components have been cooperatively utilized to extract lane boundaries on roads with different conditions. Specifically, three features are efficiently extracted to represent two lane boundaries using the lane boundaries in the previous frame. Three weights related to the three features are computed offline by the genetic algorithm for more accurate lane extraction. Furthermore, the average road intensity in all previous frames is also used to aid the robot in steering to the right direction if the inference-based verification system indicates consistent discrepancies between the current and previous frames. The system has been extensively tested and experimental results demonstrate our system can accurately guide the wheelchair robot in the navigating tasks in real time.

Keywords: Lane extraction, genetic algorithm, inference system, and wheelchair robot.

1. Introduction

Lane extraction is a crucial enabling technique in a number of intelligent vehicle applications such as autonomous driving, lateral control, lane excursion detection and warning, and intelligent cruise control. The robustness and reliability is the paramount requirement for lane detection systems since erroneous findings will generate wrong steering commands which may jeopardize safety. In this paper, we present a robust and reliable lane extraction system for the wheelchair robot, which can guide the visually impaired users to safely walk on the sidewalks of main quad of Utah State University (USU).

A few vision-based lane detection algorithms or systems are briefly reviewed. Jochem *et al.* [5] and Conrad and Foedisch [1] respectively propose a modular neural system and a support vector machines based system for autonomous vehicle navigation. Both systems can transparently navigate different road types by incorporating knowledge stored in pretrained networks or machines. The LANA system [8] combines a set of edge strength and orientation related frequency domain features with a prior deformable template to detect the lane markers. Lee *et al.* [9] use Hough transform and evolutionary strategy to find the superior lane candidate in road features. Ma *et al*. [10] apply a Bayesian multi-sensor image fusion method to locate the lane and pavement edges in the optical and radar images by using the empirical maximum a prior (MAP) estimate.

However, all these systems are not suitable and robust in real-time applications because: 1) A large number of distinct road images are required for training. 2) Complicated feature vectors and statistical models are used for finding lane characteristics. As a result, several more efficient systems are proposed. Gaspar *et al.* [2] present an efficient vision-based navigation system in the indoor environment by applying long-distance/low-precision topological navigation and short-distance/high-accuracy visual path following on the bird's eye view images obtained from the omni-directional camera. Jeong *et al.* [4] start a quick lane search without using any prior road information and recognize the lane in a changeable searching range based on the prediction derived from the previous frame. However, higher errors occur under sharp changes of the road conditions. Wang *et al.* [11] propose a robust B-Snake based lane model to describe a wider range of lane structures using the perspective effect of parallel lines under the conditions of noise, shadows, and illumination variations. Other model-based techniques [3, 6, 7] use straight lines or parabolic curves to represent the lane shape for the detection. However, these model-based approaches only focus on certain road shapes and lack the flexibility in detecting the road with arbitrary shape.

In this paper, we propose a novel feature-based lane detection algorithm for the wheelchair robot. The algorithm is robust and reliable for real-time applications under the different conditions. The remainder of the paper is organized as follows. Section 2 introduces the proposed real-time lane extraction algorithm. Section 3 presents the experimental results. Section 4 draws conclusions.

2. Proposed Approach

A sequence of frames (images) is taken at the rate of 10 frames per second from the camera installed on the wheelchair robot, which is shown in Fig. 1. The seven processing components of the proposed system are shown in Fig. 2.

Fig. 1: Sunrise Medical Quickie P300 power wheelchair

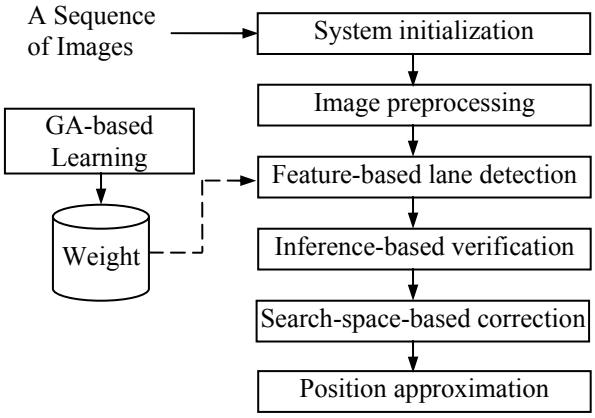


Fig. 2: The block diagram of the proposed system

2.1. System initialization

This initial process is to locate the two lane boundaries and estimate the vanishing point of the two lanes to remove the remote scene in the first frame of the sequence without using any prior road information. Five steps are involved in the process: 1) Apply Canny edge detector to extract the lane edges from the grayscale image which is converted from the RGB color space. 2) Use Hough transform to obtain the two lane boundaries. 3) Estimate the vanishing point by finding the intersection of the two lanes. 4) Remove the remote scene based on the estimated horizontal line which is a few pixels below the vanishing point. 5) Save the left and right lanes into feature vectors, which will be elaborated in section 2.3. Fig. 3 shows the final lane boundaries in red on a grayscale image after removing the remote scene.

Fig. 3: Two lane boundaries in the initial process

2.2. Image preprocessing

This step uses the road information in the previous frame to quickly remove the remote scene and find all possible lane edges. A horizontal line, which passes through a few pixels below the vanishing point found in the previous frame, is used to remove the remote scene. A partial image resulted from this removal procedure is shown in Fig. 4(c). The K-means algorithm is further applied to the histogram of this partial image to estimate the minimum intensity value (i.e., threshold) for separating the image into road and non-road regions. The centers of the two clusters in the previous frame are stored and used as the starting points of the K-means algorithm in the next frame. Finally, Sobel edge detector is applied on the thresholded binary image to locate all possible edges. Fig. 4 shows the intermediate result of each step.

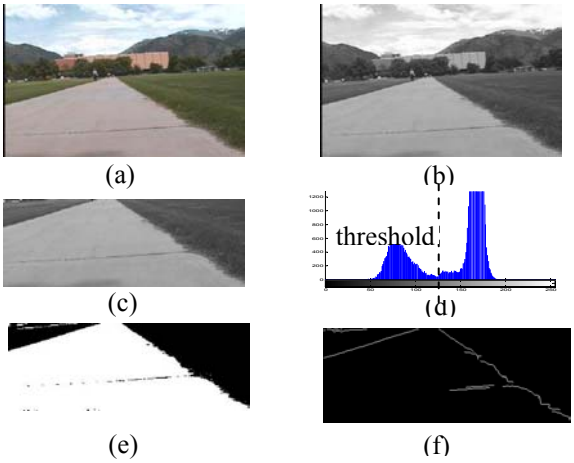


Fig. 4: Illustrations of the intermediate results of the preprocessing (a) The original image. (b) The grayscale image. (c) The remote scene removed partial image. (d) The histogram of the partial image and the threshold (e) Road and non-road regions (f) Possible lane edges

2.3. Feature-based lane detection

This lane detection process uses three features to quickly extract lane boundaries based on the possible lane edges. In order to accomplish a quick search, an upper portion of the resultant image obtained from the preprocessing is chosen as the search base. For each point p_i taken at every 3 pixels at the first line of the search base, 2 sets of three features may be derived by respectively using the left and right regions of the dividing red line crossing p_i (Fig. 5). Each set can be represented as a vector of the form $C_i = [p_i, I_i, A_i]$ where I_i is the maximum number of edge points along $p_i l_j$ or $p_i r_j$ as defined in Fig. 5, and A_i is the angle of $p_i l_k$ or $p_i r_k$ which has the maximum number of edge points at the left or right regions. The

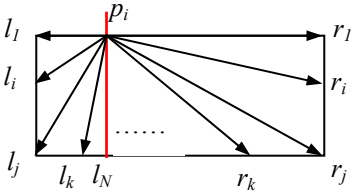


Fig. 5: Feature construction scheme

features will be saved in the left or right candidate feature vector based on the region used.

In order to ensure a fast search, the 8-neighborhood of each p_i is used to decide whether the search is needed. That is, if there are no lane edge points around p_i, the detection will directly jump to p_{i+1}, which is 3-pixel away. Otherwise, the detection resumes. This 8-neighborhood search validates the sampling scheme at the first line of the search base.

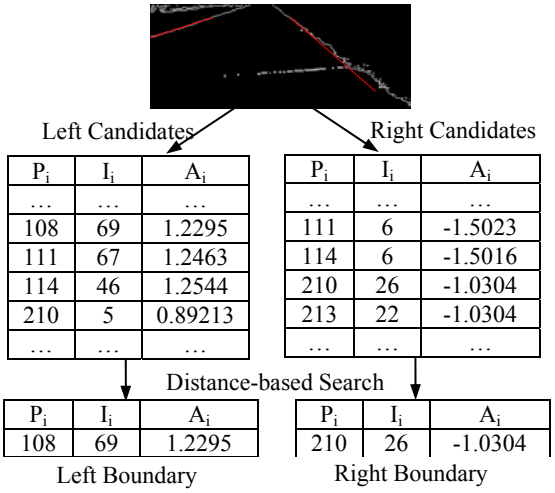
2.4. Inference-based verification

This verification process uses the lane boundaries in the previous frame to find the optimal lane boundaries in the current frame. The left and right candidate feature vectors yielded from the lane detection are compared with the corresponding lane boundaries in the previous frame to find the optimal lane boundaries. The distance formula used for such findings is:

$$\lambda_i = K_p \mid p(C_i) - p(L_{n-1}) \mid + K_I \mid I(C_i) - I(L_{n-1}) \mid + K_A \mid A(C_i) - A(L_{n-1}) \mid \quad (1)$$

where $p(C_i)$, $I(C_i)$, and $A(C_i)$ are the candidate features of the current frame, $p(L_{n-1})$, $I(L_{n-1})$, and $A(L_{n-1})$ are the lane features of the previous frame, and K_P, K_I, and K_A are the weights for each feature and are offline calculated using the genetic algorithm (GA) on road pictures taken under various environments.

In general, the 3-feature vector with the smallest distance represents the optimal lane boundary. However, if this distance is greater than a threshold, which is set to be 200 based on K_P=1.94, K_I=6.74, and K_A=48.59 obtained via training, a large difference between the previous and current lanes occurs. In such case, the lane boundaries in the previous frame are used as current lane boundaries. Fig. 6 shows the results of the lane detection and verification.



P_i	I_i	A_i
…	…	…
108	69	1.2295
111	67	1.2463
114	46	1.2544
210	5	0.89213
…	…	…

P_i	I_i	A_i
…	…	…
111	6	-1.5023
114	6	-1.5016
210	26	-1.0304
213	22	-1.0304
…	…	…

P_i	I_i	A_i
108	69	1.2295

P_i	I_i	A_i
210	26	-1.0304

Fig. 6: Feature-based lane boundaries

2.5. Search-space-based correction

The correction process takes place if the inference system keeps using the information in the previous frame to update the lane boundaries in the current frame for 5 times. Consequently, this correction will use a specific search space to direct the wheelchair robot to the road regions.

Additional road information in each frame is used in the correction process. That is, the average road intensity of the five search windows in front of the robot with the size of 15×15 (Fig. 7) is calculated for each frame. The accumulative average road intensity for all the previous frames (i.e, V_{road}) is also computed. For the current frame where a correction is considered, the average intensity of each window (i.e., V_{window}) is separately compared with V_{road} to find the possible road directions. This comparison is applied to another set of 5 windows constructed by moving the previous set outward in their corresponding directions by 20 pixels. This comparison procedure is repeatedly applied to the moving set of 5 windows until all the moving windows reach the border of the frame. The direction with the maximum number of matching windows (i.e., V_{window} is close to V_{road}) will be considered as the right steering direction for the wheelchair robot.

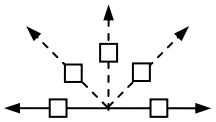


Fig. 7: Heuristic search map

2.6. Position approximation

Position approximation process estimates the position of the robot on the road. Fig. 8 shows the mechanism for such approximation. In general, $|(\theta_L + \theta_R) - 180^\circ|$ measures the degree of deviation of the robot from the road center, the proportional position of the wheelchair robot, i.e., LP/LR or PR/LR, determines the deviation direction (right or left), and MP determines the exact distance to the center.

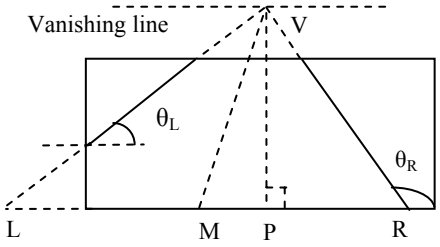


Fig. 8: Finding the position of the robot

2.7. GA-based learning

The genetic algorithm (GA) is applied offline for optimizing the three weight parameters K_P, K_I, and K_A in (1) to achieve the minimum misclassification rate. A total of 200 images with various road conditions, such as different illuminations, shadow of trees, crack

surface, debris, etc, have been used as training images. The left and right boundaries in each training image are manually marked. The input variables for the GA is $K = [K_P, K_I, K_A]$ and the cost function is

$$f(K) = Q\sum_{n=1}^{M}\left(|\, p(C_n) - P(L_n)\,| + |\, I(C_n) - I(L_n)\,| + |\, A(C_n) - A(L_n)\,|\right) + \sum_{n=1}^{M}\left(\frac{\lambda_n^1}{\lambda_n^2}\right) \quad (2)$$

where n is the image index, M is the total number of training images, L_n denotes the manually labeled left/right lane vector in frame n, C_n is the closest candidate vector to L_{n-1}, λ^1_n and λ^2_n are the distance from the current top two closest candidate vectors to L_{n-1}, and Q is the relative weight and is chosen as 10.

3. Experimental Results

A variety of experiments have been performed using the wheelchair robot on the sidewalks of main quad of USU. The laptop processor is Pentium III (2 GHZ). The CCD camera is installed on the wheelchair robot for capturing images. The size of each image is 303×397 in the 24-bit RGB format.

The lane boundaries are extracted using image preprocessing, feature-based lane detection, and inference-based verification techniques, which effectively use the prior lane boundaries to achieve real-time performance. The extraction results on three frames are shown in Fig. 9.

(a) Frame 1 (b) Frame 61 (c) Frame 120

Fig. 9: The lane boundaries of some frames

The intersection can also be corrected identified by our system based on the "abnormal" relative position between left and right lanes, where the left lane appears on the right side as shown in Fig. 10.

(a) Frame with a corner (b) Lane boundaries

Fig. 10: Corner detection

Table 1 illustrates the lane extraction accuracy of the proposed approach and several variants. The columns 2 through 4 indicate the accuracy obtained from the feature-based lane detection using equal weights, the feature-based lane detection using the optimal weights got from GA yet without using the inference technique, and the feature-based lane detection and the inference-based verification with the optimal weights (i.e., our approach), respectively. It clearly shows that our approach outperforms its variants and runs in real time with the average run time for each frame is less than 0.2 seconds.

Table 1: The lane extraction accuracy of different methods

Images	Variant 1	Variant 2	Proposed	Average run time (sec/frame)
200	85.5%	94.5%	100%	0.1744
239	77%	92.9%	100%	0.1450
83	89.2%	92.7%	100%	0.1674
140	92.1%	94.3%	100%	0.1669
138	90.6%	93.5%	100%	0.1732
275	92.4%	94.9%	100%	0.1685
123	86.2%	89.4%	100%	0.1792

4. Conclusions

This paper presents a robust and reliable lane extraction method for outdoor navigation tasks in real-time. Specifically, no prior road information is needed for extracting lanes. The main contributions are:

- Three-feature based lane extraction
- Inference-based verification
- Search-space-based correction
- Genetic-algorithm-based learning

5. References

[1] P. Conrad and M. Foedisch, "Performance Evaluation of Color Based Road Detection Using Neural Nets and Support Vector Machines," *32nd Applied Imagery Pattern Recognition Workshop*, pp. 157-160, 2003.

[2] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based Navigation and Environmental Representations with an Omni-directional Camera," *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 6, pp. 890-898, 2000.

[3] D.Grimmer and S. Lakeshmanaan, "A Deformable Template Approach to Detecting Straight Edges in Radar Images," *IEEE Trans. on PAMI*, Vol. 18, pp. 438-443, 1996.

[4] S. G. Jeong, C. S. Kim, D. Y. Lee, and S. K. Ha, **"**Real-time Lane Detection for Autonomous Vehicle," *IEEE Int'l Symposium on Industrial Electronics (ISIE)*, Vol. 3, pp. 1466-1471, 2001.

[5] T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe, "MANIAC: A Next Generation Neurally Based Autonomous Road Follower," *3rd Int'l Conf. on Intelligent Autonomous Systems*, IAS-3, 1993.

[6] D. Jung Kang, J. Won Choi, and I. S. Kweon, "Finding and Tracking Road Lanes Using Line-Snakes," *Proc. of Intelligent Vehicle*, pp. 189-194, 1996.

[7] A. Kaske, D. Wolf, and R. Husson, "Lane Boundary Detection Using Statistical Criteria," *Int'l Conf on Quality by Artificial Vision*, pp. 28-30, 1997.

[8] C. Kreucher and S. Lakshmanan, "LANA: a Lane Extraction Algorithm that Uses Frequency Domain Features," *IEEE Trans on Robotics and Automation*, Vol. 15, No.2, pp. 343-350, 1999.

[9] J. W. Lee, J. H. Kim, Y. J. Lee, K. S. Lee, "A Study on Recognition of Lane and Movement of Vehicles for Port AGV Vision System," *IEEE ISIE*, Vol. 2, pp. 463-466, 2002.

[10] B. Ma, S. Lakshmanan, and A. O. Hero, **"**Simultaneous Detection of Lane and Pavement Boundaries Using Model-Based Multisensor Fusion," *IEEE Trans. on Intelligent Transportation System*, Vol. 1, No. 3, pp. 135-147, 2000**.**

[11] Y. Wang, E. K. Teoh, and D. Shen, "Lane Detection and Tracking Using B-snake," *Image and Vision Computing*, Vol. 22, pp. 269-280, 2004.