2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Concurrent visual multiple lane detection for autonomous vehicles

Rachana Ashok Gupta, Wesley Snyder, and W. Shepherd Pitts

Abstract— **This paper proposes a monocular vision solution to simultaneous detection of multiple lanes in navigable regions / urban roads using accumulator voting. Unlike other approaches in literature, this paper first examines the extent of lane parameters required for continuous control of any vehicle manually or autonomously. The accumulator-based algorithm is designed using this fundamental control knowledge to vote for the required lane parameters (position of lanes and steering angle required) in the image plane. The novel accumulator voting scheme is called "Parametric Transform for Multi-lane Detection." This paper not only adapts predictive control in the image plane, but also detects multiple lanes in the scene concurrently in the form of multiple peaks in the accumulator. This method is robust to shadows and invariant to color, texture, and width of the road. Finally, the method is designed for dashed/continuous lines.**

I. Introduction

"Autonomous vehicles" are not a new concept. They have been proposed for the long-term goal of having vehicles driving autonomously in an unknown environment for the purpose of human safety, convenience, and ease of life. Currently, most autonomous research vehicles depend on relatively expensive LIDAR, Radar, and undependable GPS [10][12][22]. Other approaches have been tried such as autonomous driving using dead reckoning and road profile recognition [13] using on-board sensors, automated highway systems (AHS) with lanes equipped with a magnetic infrastructure, and a central communication system. These approaches have obvious disadvantages such as the need to thoroughly map the drivable terrain beforehand, and infeasibility for large public road networks due to costs. Thus, it is desirable to use computer vision systems to achieve low cost, low maintenance autonomous vehicle navigation. Recent results from vision research suggest that visual control may now be feasible and will be a major step in conversion of the entire autonomous guidance system to a vision-based one.

A. Background

There are four principle components to road detection in literature: (1) With **AI and machine learning**-based approaches the vision system is extensively trained for road and non-road regions using a neural network or other machine learning technique using a combination of features such as texture, color, luminance and coordinates of the image pixels, etc [15][21]. (2) **Color Processing** is another technique used for urban scene understanding and lane detection [1][12][14]. Very common assumptions are painted lane markings, distinct paved and marked road boundaries [10][14]. The colors and the presence of the road lane markings can vary between different geographical regions. Thus, these techniques may fail in unstructured environments and un-marked roads. (3) There have been research efforts to detect unmarked and unpaved roads using **vision area-based and stereo processing** techniques [6]. However, once the road region or boundary is classified within the desired accuracy, the boundary and road parameters must be determined to control the vehicle. Researchers mostly rely on filtering techniques to extract road edges and boundaries. There are other techniques available such as maximum entropy thresholding [19] to segment the marked road edges from road surface. (4) **Curve fitting techniques** are used in many cases to approximate the road boundaries [9][17] and surfaces[3][20]. Dickmanns and Mysliwetz present 3-D modeling of the road in [3].

According to Kluge and Thorpe [4], three are three basic criteria that are very important: (1) the accuracy of the model to fit the road, (2) computational cost of the model fitting and extraction of road parameters, and (3) the robustness and stability of the model and the fitting process in presence of noise. They discuss, in brief, different techniques used for recovering road parameters. Back-projection techniques assume constant road width [2], and thus the estimated road shape can have arbitrary errors. Road shapes have been modeled as straight lines [9][18][19], parabolae [6][17], circles [4], cubic splines [16][20] and Taylor series approximation of clothoids [3][11], etc.

Accumulator-based methods are popular as they are easy to implement and are robust to noise. Because the computational complexity of the algorithm increases exponentially with the number of parameters being estimated; these methods have been primarily used in past for straight lines [5] and constant curvature (circle) detectors [4]. Statistical fitting techniques are useful in case of fitting higher degree curves to the road, as they are computationally more efficient [4]. However, unlike statistical fitting techniques, accumulators allow detection of multiple instances of objects. This work uses an accumulator-based technique to achieve robustness and concurrency; but it modifies the accumulation technique to deal with curved roads reducing the number of parameters to be estimated to that of straight lines.

In a perspective transform, where the camera is on the vehicle looking at the road at a particular angle, the road boundaries in the image are not parallel to each other and represent two different curves and therefore two sets of parameters. Techniques like RALPH [7][12] take the inverse

R. A. Gupta is with the Department of Electrical Engineering, North Carolina State University, Raleigh, USA. ragupta@ncsu.edu
W. Snyder is with the faculty of Department of Electrical Engineering, North Carolina State University, Raleigh, USA. wes@ncsu.edu
W. S. Pitts is with Lotus Engineering Inc., Ann Arbor, Michigan, USA. shep@ncsu.edu

perspective transform to build the top view of the road in the image to determine the curvature of the road, which is now the same for both road boundaries. The advantage of this technique is the detection of both road boundaries with one parameter set. However, the inverse perspective transform can be tricky when other vehicles/objects are in the scene and when the terrain is not flat. RALPH determines the empty region of the road using laser sensors and therefore processes only the empty road region for discovering the curvature.

B. Motivation

All vision approaches mentioned above deal with detecting road/non-road regions or the boundaries as correctly as possible. Very few of them deal with vehicle control. However, for an autonomous vehicle application the most important information needed is immediate control action (for example, speed, steering angle, rate of steering change etc). Therefore, all road information detected in any form (curve equations/segmented regions/boundary locations) has to be converted into vehicle control parameters. Hence, this work adopts a control-motivated philosophy. Therefore, the problem is broken down into following sub-parts and executed in that order:

1) Determine the type of lane parameters needed to calculate the correct immediate vehicle controls to keep the vehicle in its lane.
2) Detect the lanes and the lane parameters from the scene image.
3) Use the lane parameters to calculate the local controls for the vehicle.

Other than approaching the road following problem from control perspective, the novelties of the proposed algorithm are: it is designed to intrinsically provide the lane parameters for multiple lanes (if they exist) concurrently in the image scene and is therefore called "Parametric Transform for multi-lanes," and the algorithm achieves vehicle control in pixel co-ordinates - "predictive control in image co-ordinates." The proposed algorithm also relaxes some of the assumptions required by other algorithms such as presence of painted lane markings, prior knowledge of road textures, prior learning by the vehicle controller, etc.

This paper is organized as follows: Section 2 explains the motion control strategy. Section 3 introduces the parametric transform for multi-lane detection and its mathematical theory. Navigation and predictive control in image plane is explained here. Section 4 presents the results, analysis and discussions of the failed cases. This section also compares this technique with other state-of-the-art techniques and discusses the advantages of the novel technique proposed here. Finally, Section 5 concludes the paper.

II. Motion Control Strategy

Fully functional autonomous navigation has many aspects: road detection, other vehicle detection, sign detection, intersection, exit, obstacle detection, etc. All of these aspects are equally important to achieve a fully functional autonomously navigable vehicle in a real urban setting. Therefore, the whole autonomous system needs to be aware of the entire scene. In this paper, the primary focus is designing a lane detection algorithm, which simplifies/reduces the calculations of required car control parameters (such as steering angle, speed and rate of change of steering) in order to maintain the vehicle's position within the lane boundaries. This suggests that instead of processing the complete lane boundaries in the scene, it is necessary to take a bottom-to-top approach; i.e., to know the lane parameters closest to the vehicle first.

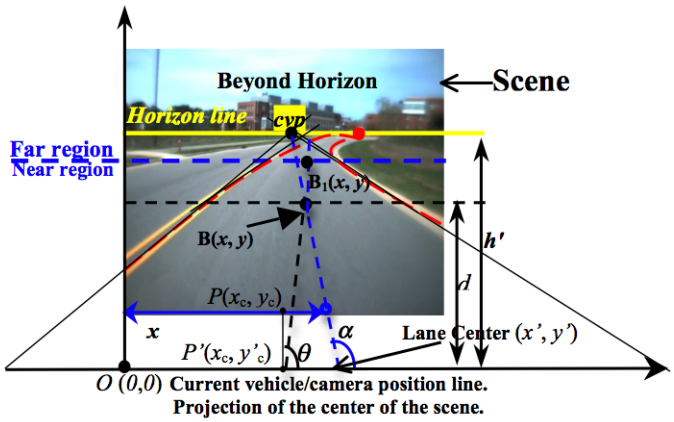


Fig. 1. Urban road scene explaining predictive motion strategy

In a lot of scenarios in an urban scene containing other vehicles and objects, the far away road region (near the horizon) is not sufficiently visible to make any significant control navigations decision. Therefore, the scene is divided into different regions as shown in Fig. 1. The "near region" consists of most of the road region starting from the bottom of the image to 3/4 of the region below horizon. The ratio can be changed as per requirements. The region above that is considered as "far region" which consists of mostly the sky and the road near the horizon. Thus the road region within the "far region" has very little information about the road parameters and therefore it is ignored.

Fig. 1 shows a simple urban road scene. Other than the fact that there are speed limits in the urban environment, vehicle speed is intuitively controlled by either the speed of another vehicle in front and/or the bending of the road. For the vehicle to follow the near region, the vehicle should always point towards the current look-ahead point **B**. Thus, the steering angle (θ) is corrected by looking at the **B**(x, y). The near region in the image plane has a point or a series of points predicting the next position(s) of the vehicle. The rate of change of steering angle (ω) can be estimated by looking at these multiple look-ahead points next to **B** if available. Thus, to find controls to reach the look-ahead point **B**, Fig. 1 shows tangents (solid black lines) approximating the road boundaries in the near region as straight lines. Therefore, the distance d to the nearest look-ahead point has to be chosen such that the lane boundaries within distance d can be safely approximated to straight lines. To fulfill this, d is defined as some percentage p of the near region, where p can be changed dynamically from frame-to-frame as a function of curvature of the road (κ) (κ is defined in a later subsection). This corresponds to the philosophy that look-ahead distance

will decrease with sharper turns Eq (1), [8].

$$d = p \cdot h', \text{where} p \propto \frac{1}{1+\kappa} \text{and} \lim_{\kappa \to 0} p \to 1 \quad (1)$$

In the perspective transform, road boundaries meet at the horizon line. The meeting point is called *vanishing point (vp)*. The point of intersection of two tangents shown in Fig. 1 can thus be called as the *"current vanishing point (cvp)"*; i.e., the point where the current closest road boundaries will meet assuming the closest road extends to infinity with zero curvature where the tangents are drawn (Eq (2)).

$$\lim_{\kappa \to 0} cvp \to vp \quad (2)$$

The vehicle might not track the center of the lane and therefore the current steering angle (θ) depends on (1) the current location of the vehicle, $P'(x_c, y_c')$ (P' is the projection of the center of the scene P, assuming that the camera is installed at the center of the vehicle) and (2) the lane angle, α, which the lane center (x', y') makes looking at the current vanishing point. θ can be calculated as per Eq. (3) and (4) if the vehicle is tracking the center of the lane correctly then $\theta = \alpha$.

$$\theta = \tan^{-1}\left(\frac{y}{x' - x_c}\right) \quad (3)$$

$$\theta = \tan^{-1}\left(\frac{d\tan\alpha}{d + (x' - x_c)\tan\alpha}\right) \quad (4)$$

Thus, the lane parameters representing every lane uniquely are (1) lane center x', (2) lane angle α, and (3) lane width W. The scene could have multiple lanes. Therefore, to distinguish different lanes, this paper considers storing the location (x') and the angle (α) in an accumulator.

This is the main principle used to design the accumulator based parametric transform to detect (x', α, W) for the lanes in the scene and predictive control in image co-ordinates.

III. Parametric Transform for Multi-Lane Detection

The multi-lane detection algorithm explained here is designed to obtain the following three lane parameters: lane center at the bottom of the scene (x'), the angle that the lane forms with the x-axis (α) and lane width W. The region beyond horizon is ignored, as it does not have any lane/road information. In this discussion, it is assumed that the "bottom" row or row 0, is at the bottom of the image, unlike many image analysis configurations which put row 0 at the top. Given the camera's pose and its projective transformation, it is straightforward to determine the location of the horizon (h').

Let $I \subset \Re^2$ denote the set of image coordinates visible in the camera's view, and let I_L denote those coordinates below the horizon. Furthermore, let x' denote a column number on the bottom row of the image. We seek a particular value of x' which will be the center of a lane. There may be more than one such x'. Let ∂ denote the set of all points in I_L which could possibly be a lane edge. Several criteria may be used to find these points. Typically, they are points of high gradient magnitude, and with some additional restrictions on the brightness or color of the regions they are dividing.

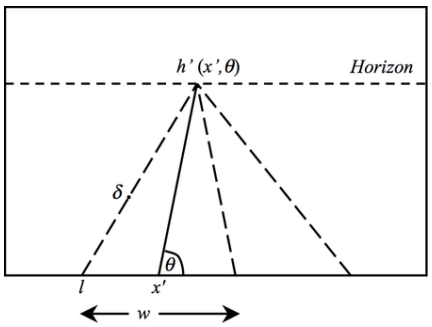


Fig. 2. Coordinate system used, showing the possibility that two lanes are visible in the image

Finally, let $h'(x', \theta)$, $h' : \Re^2 \to \Re$ denote the point on the horizon intersected by the vector through x' in the direction θ. These definitions are illustrated in Fig. 2. To determine the existence and parameters of the lane, choose a value for x' and θ, which determines $h'(x', \theta)$. Then, for each point $\delta \in \partial$, compute the intersection of the line through $h'(\cdot)$ and δ with row 0 (note that this point may not lie in the image), and call that point l. Then, if x' is indeed the center of the lane, and δ is actually a point on the lane edge, $W = 2|x' - l|$.

Define a 3-D accumulator A. Loop over all values of x', θ, computing W, and incrementing the point $A(x', \theta, W)$;

$$A(x', \theta, W) = \sum_{\delta \in \partial} p(\delta)\,, \quad (5)$$

where $p(\delta)$ is a weight, $0 \le p(\cdot) \le 1$, representing the confidence that δ is a road edge. Each distinct peak in the accumulator will represent a lane in the image.

This version of the algorithm is computationally heavy, and in the next section, we trade computation in the image domain for computation in the accumulator domain, in order to obtain the speed necessary for real-time control of a vehicle.

Instead of voting at every edge point in the image as a potential road edge point in the accumulator A for x, θ, W, voting by pairs of lines as potential set of lane boundary segments in the accumulator A for x', α is proposed. This is achieved by first using the simplest and classic Hough transform to find straight-line components in the near region. Criteria and constraints defined such that the lines having highest potential to form objects which look like lanes are used in the voting scheme. Thus avoiding the edges created by shadows and other noise.

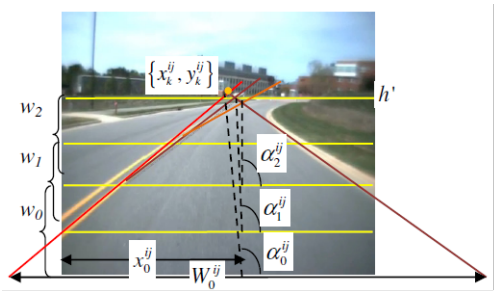


Fig. 3. Sliding window approach from bottom to top

The near region of the image is split into overlapping windows, as shown in Fig. 3; where 50% of each window w_{i+1} overlaps with window w_i and w_{i+2} (See Fig. 3). In each

window, the potential road boundaries are detected using the Hough transform on the Canny edge-detected image. The number of windows is chosen such that window height is d which is function of κ (Eq. (1)). Thus, the small segments of the road boundaries in each window can be safely approximated (by Hough transform) as straight lines. However the minimum number of windows is $K = 3$. The Hough output in window w_k is a set of lines $L_k = \{L_k^1, L_k^2, L_k^3 ..., L_k^n\}$. The goal is to locate a pair of lines $\{L_k^i, L_k^j\}$ in window w_k such that $L_k^i, L_k^j \in L_k$ and $L_k^i, L_k^j \in \partial S$, where S is a possible navigable road region with boundary ∂S. Therefore, a set of pairs of lines, L, is formed as per Eq (6).

$$L = \{\{L_k^i, L_k^j\} \Big| L_k^i \in \mathrm{L}_k \text{ and } L_k^j \in \mathrm{L}_k \text{ and } i \neq j\} \tag{6}$$

The following criteria must be satisfied by $\{L_k^i, L_k^j\}$ for them to match with possible ∂S.

Criterion 1: *cvp* on the horizon satisfies

$$y_k^{ij} > 0 \tag{7}$$

$$\varepsilon_k^{ij} = \left| y_k^{ij} - h' \right| \approx 0 \tag{8}$$

Eq (8) requires that the intersection point $\{x_k^{ij}, y_k^{ij}\}$ of a pair of lines $\{L_k^i, L_k^j\}$ should be as close to *cvp* (cvp_x, cvp_y), i.e. horizon line (h') as possible considering the region is flat (see Fig. 3). h' cannot be negative i.e. h' cannot be below the bottom of the image (Eq (7)).

Criterion 2: Sufficient width.

This criterion requires that the width of the lane should be greater than a pre-determined limit, W_0 (in pixels), which can be determined from camera resolution (pixels/meter) and the car width. This ensures that two lines very close to each other will not be considered as navigable lane boundaries. This criterion helps filter noisy edges, lines formed due to shadows, lines due to other anomalies on the road such as arrows, etc. W_k^{ij} is the width between lines $\{L_k^i, L_k^j\}$ measured in pixels at the bottom of the image plane (Eq (9) and Fig 3).

$$W_k^{ij} > W_0 \tag{9}$$

Criterion 3: Smoothness factor.

$$\begin{aligned} s_k^{ij} &= \int_{S_k^{ij'}} \nabla G \\ &\text{Where,} S_k^{ij'} \in w_k, L_k^i, L_k^j \in \partial S_k^{ij'} \\ &\nabla G = \nabla Gx + \nabla Gy \\ &\nabla Gx = |S_E(x,y) - S_E(x+1,y)| \\ &\nabla Gy = |S_E(x,y) - S_E(x,y+1)| \end{aligned} \tag{10}$$

This criterion requires that the gradient (change in the intensity) integrated over the region $S_k^{ij'}$ in the gray scale image S_E be very small. $S_k^{ij'}$ is the region confined by lines L_k^i, L_k^j, the lower and the upper row of the window w_k (see Fig. 3). Thus, s_k^{ij} is called the "smoothness factor" for region $S_k^{ij'}$. The lower the value of s_k^{ij}, the smoother, and thus the more navigable is the region.

After using criteria (1), (2), and (3) to filter to obtain the potential lane boundaries; the following objective function J_k^{ij} is calculated for each potential pair $\{L_k^i, L_k^j\}$ of road boundaries (Eq (11)).

$$J_k^{ij} = Ke^{-\left(\varepsilon_k^{ij} + s_k^{ij}\right)} \tag{11}$$

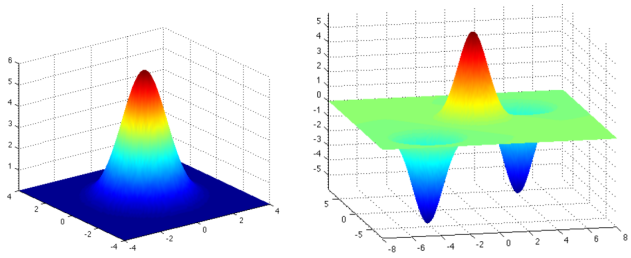
J_k^{ij} is called the voting strength for the pair $\{L_k^i, L_k^j\}$. Given h', lane center resulting from each potential pair of lines $\{L_k^i, L_k^j\}$, potential boundaries of the lane, can be uniquely represented by two parameters α_k^{ij}, the lane angle and $x_k^{'ij}$, x-coordinate of the center of two lines (See Fig. 3). Therefore, a 2-D accumulator is constructed and a voting system is used to vote for each pair of lines with a voting strength of J_k^{ij} in A, with $\{x', \alpha\}$. The accumulator is updated for all K windows in each frame. Therefore, if a navigable road/region exists in the image, all the overlapping windows should have a pair of boundaries with almost the same α_k^{ij} and $x_k^{'ij}$ forming a peak in the accumulator A at $\{x_k^{'ij}, \alpha_k^{ij}\}$.

$$\begin{aligned} &\mathrm{A}(x', \alpha) = \sum_{k=0}^{K} \sum_{i}^{n} \sum_{j(i \neq j)}^{n} J_k^{ij} \cdot f(x_k^{'ij} - x', \alpha_k^{ij} - \alpha) \\ &\text{Where,} \\ &f(a,b) = \begin{cases} e^{\frac{-(a^2+b^2)}{2\sigma^2}}, & \forall -x_L \leq a \leq x_L \text{ and } -\alpha_L \leq b \leq \alpha_L \\ 0, & \text{otherwise} \end{cases} \\ &\alpha_k^{ij} = \tan^{-1}\left(\frac{y_k^{ij}}{x_k^{'ij} - x_k^{ij}}\right), x_k^{'ij} = \left(\frac{x_k^i + x_k^j}{2}\right) \\ &J_k^{ij} = \text{Voting strength,} \\ &K = \text{Number of Windows in the images} \end{aligned} \tag{12}$$

A small neighborhood of $\{\pm x_L, \pm \alpha_L\}$ around each $\{x_k^{'ij}, \alpha_k^{ij}\}$ is increased using a Gaussian blur kernel with standard deviation σ, for noise removal and smoothed peaks. See Eq (12), Fig. 4(a).

A. Negative voting scheme

As the voting scheme considers all the possible potential pairs of lines to vote in the accumulator, there is a possibility of falsely detecting a lane at the location of the edge shared by two lanes. This is illustrated in Fig. 5(a). The accumulator will have multiple peaks representing: (1) lane 1 due to line pair AB, AC, (2) lane 2 due to line pair AC, AD, and (3) a false lane 3 due to line pair AB, AD as this pair too satisfies all the criteria for a lane. To avoid this, we use the

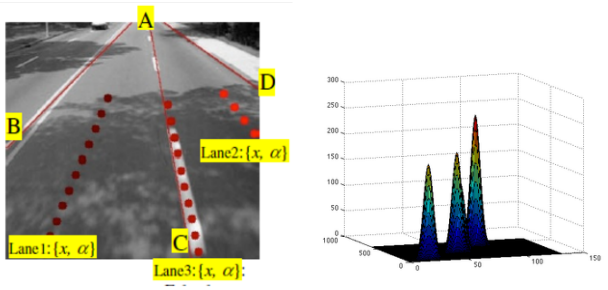


(a) Original $f(a,b)$ (b) New $f(a,b)$

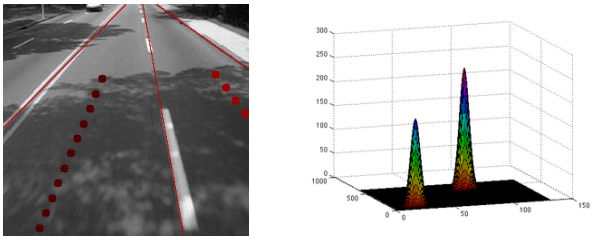
Fig. 4. Negative Voting Scheme: change in $f(a,b)$

negative voting scheme as follows: If a potential lane center with $\{x_k^{'ij}, \alpha_k^{ij}\}$ is formed with voting strength J_k^{ij} due to line pair $\{L_k^i, L_k^j\}$, then none of the lines (lane edges) $\{L_k^i, L_k^j\}$

can be lane centers and each should be voted with equal in magnitude but with negative voting strength $-J_k^{ij}$ in the same accumulator A. Thus, $f(a,b)$ in Eq. (12) is changed to new $f(a,b)$ shown in Fig. 4(b); where negative peaks represent $\{x',\alpha\}$ of lane edges $\{L_k^i, L_k^j\}$. The positive peak represents $\{x_k'^{ij}, \alpha_k^{ij}\}$. The variance of the Gaussian blur depends on the width of the lane W_k^{ij}. Later, the negative values in the accumulator are pulled to zero to get only positive peaks representing the true lane centers, see Fig. 5(a), Fig. 5(b)).



(a) False lane detection. Original road figure is from [16]



(b) False lane is avoided by negative votes in A

Fig. 5.

B. Predictive control in image co-ordinates

It is demonstrated below that the vehicle control parameters can be calculated from the image plane co-ordinates; thus the transformation to the ground plane co-ordinates is not necessary. Let $\{x',\alpha\}_t$ be the lane parameters in

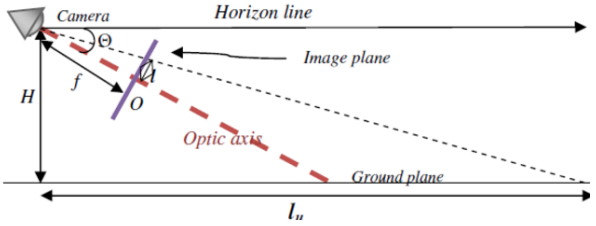


Fig. 6. Camera Configuration

current frame/window F_t and $\{x',\alpha\}_{t+1}$ be the corresponding lane parameters in the next frame/window F_{t+1}. "Predictive control" is analogous to a human driver's behavior where he/she approximately predicts the next point B_t for the vehicle to reach, while at point P_t in F_t, and adjusts the speed and the steering in order to reach that point. A human driver will change the steering angle to match the new angle θ_t at the rate of ω as per accumulator peak from F_t, Eq (4).

Let the distance l (pixels) travelled between 2 frames in image plane correspond to the distance l_u in the ground plane (Fig. 1 and 6). From the perspective transform, we have Eq (13); where H is the camera height from the ground plane, Θ is the camera angle with respect to the horizon, and f is the focal length of the camera, [11] (See Fig. 6).

$$l_u = \frac{H(\cos\Theta - l\sin\Theta)}{f(l\cos\Theta + \sin\Theta)} \tag{13}$$

Let υ and υ_u be the vehicle speed in image plane and in the ground plane respectively, defined in Eq (14).

$$\upsilon = \frac{l}{\Delta t},\ \upsilon_u = \frac{l_u}{\Delta t},\ \text{where } \Delta t = (\text{frames per second})^{-1} \tag{14}$$

Thus, assuming either l or l_u is known, the curvature of the road in image plane, κ, is defined in terms of α and l. The vehicle speed υ can be calculated as a function of κ and a constant maximum speed for the vehicle υ_{max} in pixels. Constants υ_{max} and β are to be determined experimentally to suit the vehicle dynamics, see Eq (15). The philosophy is that the vehicle speed is inversely proportional to the κ, [8].

$$\kappa_t = \frac{\alpha_{t+1} - \alpha_t}{l} \text{ and } \upsilon = \frac{\beta\,\upsilon_{max}}{(1+\kappa)} \tag{15}$$

Let ω be the rate of change of steering angle, which will depend on θ, Eq (16).

$$\omega = \frac{\theta_{t+1} - \theta_t}{\Delta t}, \tag{16}$$

If the vehicle tracks the center of the lane correctly, it will match with the center of the image plane and thus, $\alpha = \theta$. However, the vehicle might not always follow the center of the lane and therefore, steering control will be dependent on the calculation of θ. Differentiating Eq (13) gives,

$$\begin{aligned} \frac{dl_u}{dt} &= d\left(\frac{H(\cos\Theta + l\sin\Theta)}{f(\sin\Theta - l\cos\Theta)}\right)/dt \\ \frac{l_u\cdot\omega}{(\theta_{t+1}-\theta_t)} &= \frac{H\upsilon}{f(\sin\Theta - l\cos\Theta)^2} \end{aligned} \tag{17}$$

At point P_t, the final steering angle required is θ_{t+1}, Eq (4). Once a safe speed υ_u for the vehicle is calculated from l_u for a given curvature using Eq (13), (14) and (15); the corresponding steering rate ω can be calculated recursively from current frame F_t to F_{t+1} using Eq (17). Thus, the relationship between image plane and ground plane velocity is derived in terms of image plane parameters, the vehicle control parameters υ_u, ω, θ can be computed directly from the image plane, and a transformation to the ground plane is not necessary. Moreover, as θ and υ are functions of $\{x',\alpha\}$; the lane parameters $\{x',\alpha\}$ can be used for the vehicle control.

IV. Result and Analysis

The algorithm in this work is designed to detect multiple lanes simultaneously in the form of multiple peaks in an accumulator. This is similar to any parametric transform where multiple instances of the object and their parameters can be detected simultaneously using accumulator voting. Straight line Hough is a classic example.

The algorithm is tested on road images in this paper in addition to some of the road images from the previous lane detection literature, see Fig. 5 and 7. The red dots

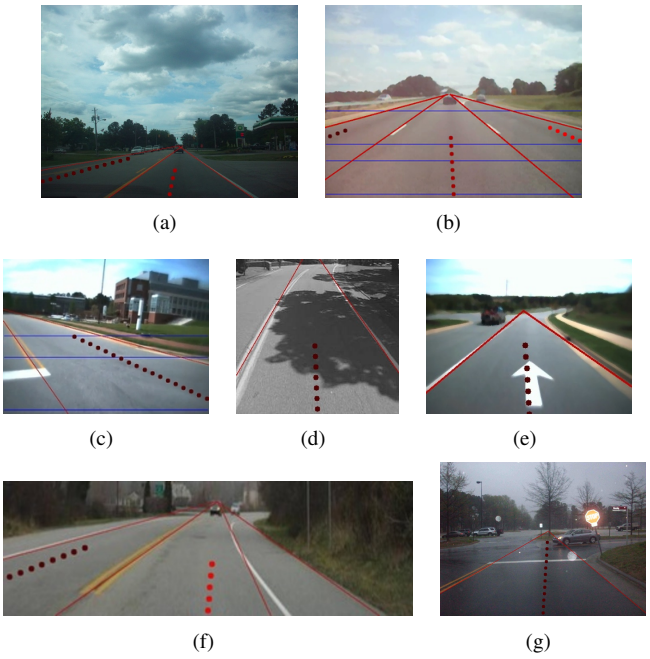


Fig. 7. Concurrent multi-lane detection results. Original road image in 7(d) are taken from [16]

predict the direction of vehicle motion as well as the future position of the vehicle. Red solid lines are the approximately predicted locations of the road boundaries in the "near region" extended up to the horizon.

It is observed that irrespective of the lighting conditions, horizon location on the scene image, width of the road, shadows, and road markings; the algorithm detects the multiple lanes. It does not depend on whether the camera is pointing downwards towards the road - getting almost all the image with road in it (Fig. 7(d)) - or it is pointing towards horizon with the road in less than half of the scene image, Fig. 7(a), 7(b), 7(g). As the algorithm does not consider the color of the line yet, even the lane with opposite traffic is marked as additional navigable lane. Currently, the algorithm takes on an average 1.5 to 2 frames/second to detect the lanes on a laptop (dual core 2.8 GHz) with room for improvement.

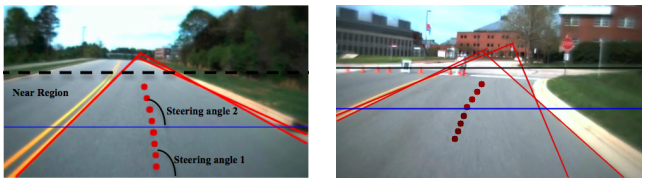


Fig. 8. Multiple lanes suggesting the change in the steering angle within the same lane in the near region

The algorithm splits the near region into multiple windows. Therefore, in case of higher curvature roads, the multi-lane detector highlights 2 pairs of lane boundaries in the same lane vertically apart (See Fig. 8). This is useful in predictive control of steering angles in the next frames as explained in section III. More research has to be done regarding this.

A. Discussion on failed cases

The proposed algorithm depends on detecting strong edges that potentially represent lines (painted, unpainted, physical, etc); which match the shape of the road boundaries. Additionally, the algorithm filters the detected lines based on the previously mentioned criteria such as enough width, smoothness of the region, etc (See section III). Therefore the algorithm fails in cases where the correct road edges are not strong enough to be detected, are absent or other non-road strong lines are present in the scene satisfying the same criterion such as painted arrows, pedestrian cross-walk segments, (Fig. 9(c)), cars (Fig. 9(a)), etc. In some cases, the algorithm detects a wide pedestrian side-walk as another lane because it does not distinguish the difference between a curb and a line, Fig. 9(d).

Some of these problems can be solved by introducing lane edge tracking, current lane tracking or temporal filtering such that the algorithm will filter anomalies and also work with disappearance of road-edge for a few frames. Temporal filtering and tracking will also increase the processed frames per second as the algorithm will be processing small parts of the scene (corresponding to the road edges present in the previous frame) as opposed to the whole frame. This will be addressed in future work.

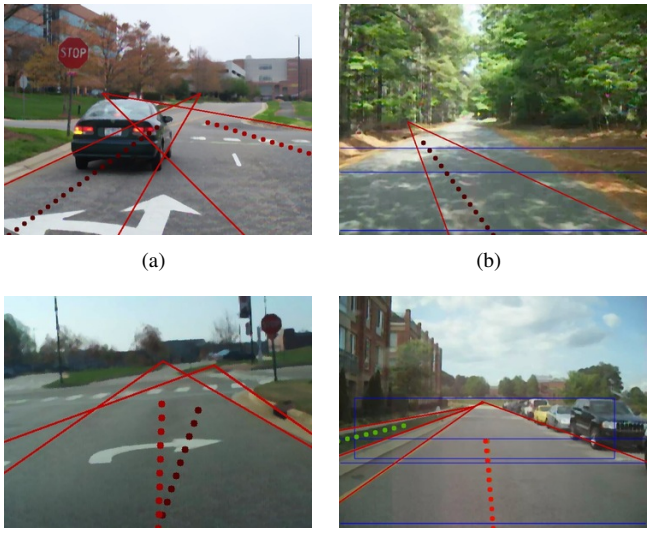


Fig. 9. Some failed results

B. Comparison with other techniques

So far, lane detection algorithms in the literature do not detect multiple lanes in the scene simultaneously [7][16][19]. Wang et al. [16] used accumulator voting prior to B-spline fitting; however, they vote for the vanishing point. As the vanishing point is same for all the lanes in the scene, their accumulator does not represent multiple lanes, Fig. 5(b). This algorithm intrinsically develops multiple peaks in the accumulator without additional processing for multiple lanes. Yu et al. [19] used a splitting window approach. However, they split the image into fixed ratios and do not consider the possible look-ahead distance d to decide the near region

boundary. Additionally, the algorithms in the literature which use the Hough transform to detect road edges do not deal with curved roads [19]. The proposed method uses the path tracking control knowledge to detect the required road angle and therefore can deal with curved roads without knowing the complete road boundaries. The results presented in this paper are achieved in the image coordinate space; therefore, there is no need for an inverse perspective transform inorder to navigate properly.

V. Conclusion

This work proposed a monocular vision solution to detection of multiple lanes in navigable regions / urban roads. The paper first formulated an opinion on the lane parameters needed to control any vehicle autonomously/manually in real-time. This control knowledge was then used to design an accumulator voting scheme that determines the location and the angle of multiple lanes (if they exist) in the image scene; in the form of multiple distinct peaks in an accumulator - "Parametric transform for multi-lanes." The parametric transform design is modified with some practical road edge constraints to reduce the accumulator dimensions and thus making it useful in real-time operation. This paper adapts control in the image plane from ground plane control - "Predictive control in the image coordinates." Further, the method is robust to shadows and invariant to color, texture, and width of the road. Finally, the method is applicable to both straight and curved roads.

Acknowledgement

The authors acknowledge Lotus Engineering Inc. USA for supporting this research. The opinions expressed are those of the authors and do not necessarily reflect the views of Lotus Engineering Inc. USA. Authors also acknowledge following sponsors for their contribution, help and/or research support: Analog Devices, Automation Direct, BWI Eagle, Classic Auto Air, Comtrol, Four Brothers Auto Air, Kinetik Audio, North Carolina Center for Automotive Research, NCSU, Powertrain Control Systems, Prosilica, 1stVision, Revware, Tamron, TRW, and Vicor.

References

[1] Crisman, J. D. and Thorpe, C. E. *Color Vision for Road Following*. In *Vision and Navigation: The CMU Navlab, C. Thorpe (Ed*, pp. 9–24. 1988.

[2] Turk, M. A., Morgenthaler, D. G., Gremban, K. D., and Marra, M. *VITS-a vision system for autonomous land vehicle navigation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(3):342–361, May 1988.

[3] Dickmanns, E. D. and Mysliwetz, B. D. *Recursive 3-D Road and Relative Ego-State Recognition*. IEEE Trans. Pattern Anal. Mach. Intell., 14(2):199–213, 1992.

[4] Kluge, K. and Thorpe, C. *Representation and recovery of road geometry in YARF*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 114–119. Jun 1992.

[5] Suzuki, A., Yasui, N., Nakano, N., and Kaneko, M. *Lane recognition system for guiding of autonomous vehicle*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 196–201. Jun-1 Jul 1992.

[6] Kluge, K. *Extracting road curvature and orientation from image edge points without perceptual grouping into features*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 109–114. Oct 1994.

[7] Pomerleau, D. *RALPH: rapidly adapting lateral position handler*. In *Proceedings of the Intelligent Vehicles Symposium*, pp. 506–511. Sep 1995.

[8] Yoshizawa, K., Hashimoto, H., Wada, M., and Mori, S. *Path tracking control of mobile robots using a quadratic curve*. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 58–63. Sep 1996.

[9] Taylor, C. J., Koseck, J., Blasi, R., and Malik, J. *A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving*. International Journal of Robotics Research, 18:442–453, 1999.

[10] Hofmann, U., Rieder, A., and Dickmanns, E. D. *Radar and Vision Data Fusion for Hybrid Adaptive Cruise Control on Highways*. In *Proceedings of the Second International Workshop on Computer Vision Systems*, pp. 125–138. London, UK, 2001. ISBN 3-540-42285-4.

[11] Southall, B. and Taylor, C. J. *Stochastic road shape estimation*. In *Proceedings. of Eighth IEEE International Conference on Computer Vision*, vol. 1, pp. 205–212. 2001.

[12] Rasmussen, C. *Combining Laser Range, Color, and Texture Cues for Autonomous Road Following*. In *In IEEE International Conference on Robotics and Automation*, pp. 4320–4325. 2002.

[13] Holzapfel, W., Sofsky, M., and Neuschaefer-Rube, U. *Road profile recognition for autonomous car navigation and Navstar GPS support*. IEEE Transactions on Aerospace and Electronic Systems, 39(1):2–12, Jan 2003.

[14] He, Y., Wang, H., and Zhang, B. *Color-based road detection in urban traffic scenes*. IEEE Transactions on Intelligent Transportation Systems, 5(4):309–318, Dec 2004.

[15] Hu, M., Yang, W., Ren, M., and Yang, J. *A vision based road detection algorithm*. In *IEEE Conference on Robotics, Automation and Mechatronics*, vol. 2, pp. 846–850. Dec 2004.

[16] Wang, Y., Teoh, E. K., and Shen, D. *Lane Detection and Tracking Using B-Snake*. International Conference on Information, Intelligence, and Systems, 0:438, 2004.

[17] Jung, C. R. and Kelber, C. R. *An Improved Linear-Parabolic Model for Lane Following and Curve Detection*. In *Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*, p. 131. Washington, DC, USA, 2005. ISBN 0-7695-2389-7.

[18] Schreiber, D., Alefs, B., and Clabian, M. *Single camera lane detection and tracking*. In *Proceedings of IEEE Intelligent Transportation Systems*, pp. 302–307. Sept 2005.

[19] Yu, T., Wang, R., Jin, L., Chu, J., and Guo, L. *Lane mark segmentation method based on maximum entropy*. In *Proceedings. IEEE Intelligent Transportation Systems*, pp. 177–181. Sept 2005.

[20] Wang, H. and Kearney, J. K. *A parametric model for oriented, navigable surfaces in virtual environments*. In *Proceedings of the ACM international conference on Virtual reality continuum and its applications*, pp. 51–57. New York, NY, USA, 2006. ISBN 1-59593-324-7.

[21] Sha, Y., Zhang, G.-Y., and Yang, Y. *A Road Detection Algorithm by Boosting Using Feature Combination*. In *IEEE Intelligent Vehicles Symposium*, pp. 364–368. June 2007.

[22] Peterson, K., Ziglar, J., and Rybski, P. E. *Fast feature detection and stochastic parameter estimation of road shape using multiple LIDAR*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 612–619. Sept 2008.