

Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision

Radu Danescu and Sergiu Nedevschi, *Member, IEEE*

Abstract—Accurate and robust lane results are of great significance in any driving-assistance system. To achieve robustness and accuracy in difficult scenarios, probabilistic estimation techniques are needed to compensate for the errors in the detection of lane-delimiting features. This paper presents a solution for lane estimation in difficult scenarios based on the particle-filtering framework. The solution employs a novel technique for pitch detection based on the fusion of two stereovision-based cues, a novel method for particle measurement and weighing using multiple lane-delimiting cues extracted by grayscale and stereo data processing, and a novel method for deciding upon the validity of the lane-estimation results. Initialization samples are used for uniform handling of the road discontinuities, eliminating the need for explicit track initialization. The resulting solution has proven to be a reliable and fast lane detector for difficult scenarios.

Index Terms—Cue fusion, lane detection, particle filtering, stereovision, tracking.

I. INTRODUCTION

LANE/ROAD detection has been a fertile research field for decades due to the great significance of accurate and robust road description results in any driving-assistance system. The algorithms have increasingly become complex, as the targeted scenarios increasingly became difficult. From the highway scenario, the lane-detection systems moved to city and country roads. With this move, the initial emphasis on lane-delimiting features, such as lane markings, was replaced by the emphasis on model parameter estimation techniques, which use static and dynamic knowledge-based probabilistic constraints to counteract possible noisy features and smooth the result. These constraints lead to probabilistic reasoning in the form of tracking, which is traditionally achieved by the use of the Kalman filter (KF). The use of KF tracking has the advantage of reducing the search space, eliminating the detection outliers, and smoothing the result.

The features that make the KF solutions smooth and efficient are the very features that cause problems when the road is not continuous. Sharp turns, lane changes, and atypical road geometries pose problems to a tracker that represents the lane probability density as a Gaussian function, and the reduction of the search space around the past results makes it difficult to han-

dle new hypotheses and causes detection errors to accumulate if the search regions are drawn toward false delimiters.

Particle filtering is a novel technology for probability-based tracking, allowing multiple-hypothesis tracking, simple measurement, and faster handling of road discontinuities.

This paper describes a lane-detection system that combines the advantage of particle filtering, stereovision, and grayscale image processing to achieve robust lane-estimation results in difficult scenarios of city, highway, and country roads.

II. PROBABILISTIC FOUNDATIONS OF LANE TRACKING

While there is no universal definition of tracking, we can regard it as the process of reasoning about the state of a time-evolving entity, given a sequence of observations. In particular, lane tracking can be defined as the process of reasoning about the position and geometry of the lane, given a sequence of image-derived feature sets.

The goal of tracking as probabilistic inference is to evaluate $P(\mathbf{X}_i | \mathbf{Y}_0 = y_0, \dots, \mathbf{Y}_i = y_i)$, i.e., to compute the conditional probability density of state \mathbf{X}_i , given the sequence of measurements from the past and current frames.

Due to the fact that the tracking process must deliver results at each frame and that a tracker should be able to function in mostly the same way for an indefinite period of time, the process of estimation of $P(\mathbf{X}_i | \mathbf{Y}_0 = y_0, \dots, \mathbf{Y}_i = y_i)$ has to be written in a recursive manner, such that the results of the past frames can be reused in the estimation for the current frame. To achieve this, four concepts are used.

- 1) *Dynamic model*: $P(\mathbf{X}_i | \mathbf{X}_{i-1})$ is the probability of reaching some value of random variable \mathbf{X}_i , given the past state \mathbf{X}_{i-1} , under the assumption that only the immediate past matters.
- 2) *Prediction*: Computation of the conditional probability density of the current state, given the past sequence of measurements, i.e., $P(\mathbf{X}_i | \mathbf{Y}_0 = y_0, \dots, \mathbf{Y}_{i-1} = y_{i-1})$. Given the simplification assumption that only the immediate past matters, the prediction probability values can be recursively computed, given the past results and the dynamic model, i.e.,

$$P(\mathbf{X}_i | y_0, \dots, y_{i-1}) = \int P(\mathbf{X}_i | \mathbf{X}_{i-1}) P(\mathbf{X}_{i-1} | y_0, \dots, y_{i-1}) d\mathbf{X}_{i-1}. \quad (1)$$

- 3) *Data association*: At each frame i , there may be several measurements that are available, and not all of them are equally useful. Denoting by y_i^r the r th measurement

Manuscript received April 10, 2008; revised January 5, 2009 and January 28, 2009. First published April 28, 2009; current version published June 2, 2009. The Associate Editor for this paper was R. Hammoud.

The authors are with the Technical University of Cluj-Napoca, 400 020 Cluj-Napoca, Romania (e-mail: Radu.Danescu@cs.utcluj.ro; Sergiu.Nedevschi@cs.utcluj.ro).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2009.2018328

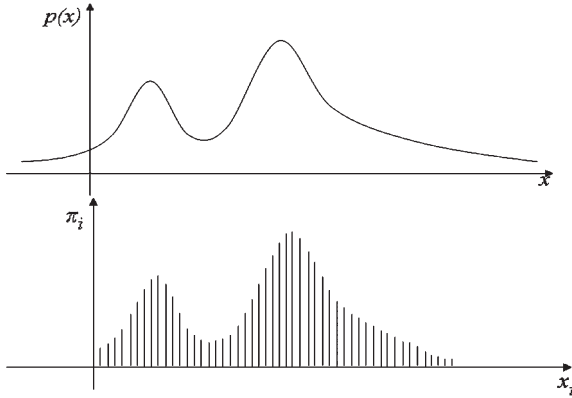


Fig. 1. Analogy between a probability density function and a set of weighted samples.

of frame i , the probability of this measurement being useful is expressed as $P(\mathbf{Y}_i = y_i^r | y_0, \dots, y_{i-1})$. If each measurement is conditionally independent of the others (the measurement independence assumption is taken), the usefulness of each measurement can be computed as

$$P(\mathbf{Y}_i = y_i^r | y_0, \dots, y_{i-1}) = \int P(\mathbf{Y}_i = y_i^r | \mathbf{X}_i) P(\mathbf{X}_i | y_0, \dots, y_{i-1}) d\mathbf{X}_i. \quad (2)$$

4) *State update*: State probability density $P(\mathbf{X}_i | \mathbf{Y}_0 = y_0, \dots, \mathbf{Y}_i = y_i)$, which is the end result of the tracking process, is computed using Bayes' rule, i.e.,

$$P(\mathbf{X}_i | y_0, \dots, y_i) = \frac{P(y_i | \mathbf{X}_i) P(\mathbf{X}_i | y_0, \dots, y_{i-1})}{\int P(y_i | \mathbf{X}_i) P(\mathbf{X}_i | y_0, \dots, y_{i-1}) d\mathbf{X}_i}. \quad (3)$$

The equations of tracking as probabilistic inference are too complex to apply in the general case. Furthermore, the probability densities involved are impossible to analytically represent most of the time and are therefore approximated. Approximating means either coercing them to a known probability density function, such as a Gaussian, or maintaining a discrete numerical representation throughout the whole process. The Gaussian representation leads to the well-known KF solutions, and the representation as discrete samples leads to the particle-filtering solutions.

III. PARTICLE FILTERING

A practical approach to tracking general probability density functions, i.e., particle filtering, is described in [3]. Instead of trying to analytically approximate an unknown function, their system uses N discrete values called "samples" or "particles." At each given time t , a particle i is defined by a value \mathbf{x}_t^i and a weight π_t^i , with the sum of all weights being 1 (see Fig. 1).

The problem of tracking becomes the problem of evaluating the values and the weights, given a dynamic model and an observation density function.

For algorithm optimization purposes, a parameter is added to each particle, changing the particle representation to $\{\mathbf{x}_t^i, \pi_t^i, c_t^i, i = 1, \dots, N\}$. This parameter is defined as the sum

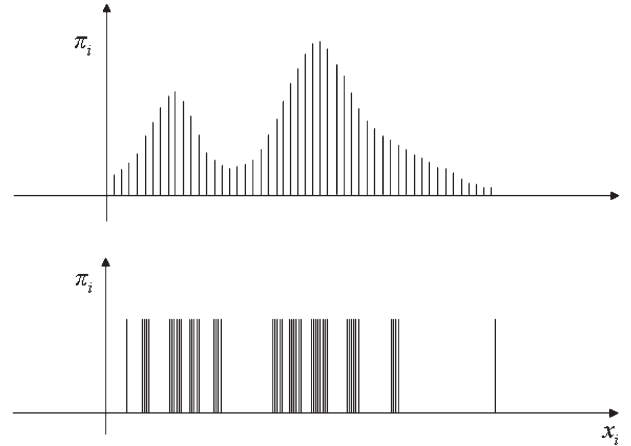


Fig. 2. Same probability density function approximated by weighted and weightless particles.

of the weights of each particle from 1 to i (a cumulative histogram). Each iteration of the CONDENSATION algorithm has the aim of evaluating a new set of particles, given the previous set, the dynamic model, and the measurements.

The first step of the algorithm is resampling. A weighted sample set is transformed into a new set of samples of equal weight but uneven concentration through the domain of values of \mathbf{x} . This is achieved by performing N random draws from the particle set using the particle weights as probabilities for particle selection. A particle having a larger weight will be selected several times, whereas a particle having a low weight may not be selected at all. The new set of weightless particles and the weighted set approximate the same density function (see Fig. 2).

Prediction is the next step of the CONDENSATION algorithm. In a general form, this is achieved by sampling from the dynamic model density function. This function describes the likelihood of each possible current state, given the assumption that the past state is described by the value of weightless particle i . A more pragmatic approach is to assume that the new state is derived from the past state partly by a deterministic process, which is described by a function or a linear transformation, and partly by a random factor.

Each weightless particle that resulted from the resampling step is subjected to a deterministic transformation, which will take into account the state transition equations of the system, and a stochastic diffusion, which will account for the random events that may change the state (see Figs. 3 and 4).

The final step of the algorithm is the *measurement/update* process. In the general formulation of the tracking problem as probabilistic inference, updating means applying Bayes' rule to get the posterior probability density, given the prior probability density and the measurement. The prior state probability density is, at this point, completely encoded in the distribution of the values of the weightless particles through the domain of possible state values. The posterior probability density function is obtained by simply weighing the particles using the likelihood of observation $p(y_t | \mathbf{x}_t = \mathbf{x}_t^i)$. Several cues can be combined in this step by multiplication using the cue conditional independence assumption, if applicable (see Fig. 5).

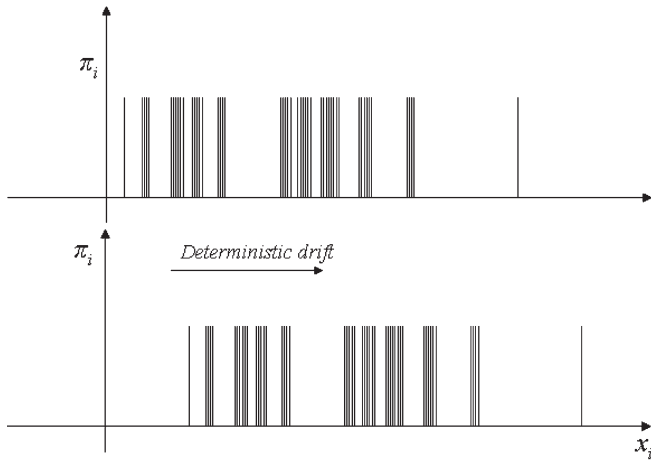


Fig. 3. Deterministic drift using weightless particles.

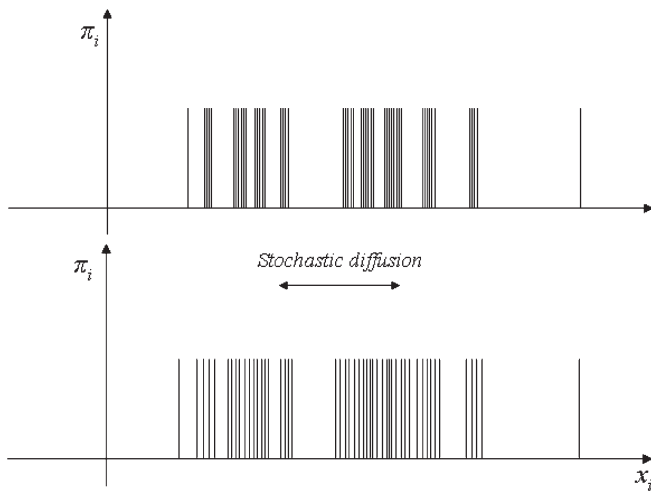


Fig. 4. Stochastic diffusion using weightless particles.

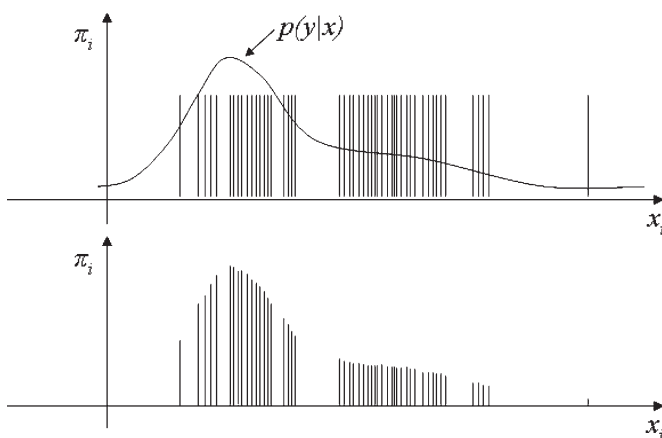


Fig. 5. Weightless particles are weighed by measurement.

IV. RELATED WORK

Lane estimation through Kalman filtering was pioneered by Dickmanns and Myśliwetz [1], and since then, many researchers have devised working solutions, such as [2]–[7]. The Kalman-filter-based lane tracking relies on the model-based prediction for establishing search regions for detection and uses the detection results to update the state. This approach expects

a continuously varying road situation, and the discontinuities are usually handled by reinitializing the tracking process. The solution presented in [6] handles some particular case of road discontinuities by using two instances of the road model, but it is clear that the KF is not the best choice for tracking discontinuous roads.

A shift toward particle filtering for lane estimation is currently taking place. A particle-based lane solution usually starts with particle sampling, followed by drifting and measurement. The measurement step is considerably simpler, in comparison to the KF, because it usually consists of a comparison between the particle and the image data, from which a weight is derived; therefore, no complex detection algorithms are required. However, the measurement step is executed for each particle, and therefore, the simplicity is essential for adequate time performance. Southall and Taylor [10] presented a lane detector based on a condensation framework, which uses lane-marking points as measurement features. Each point in the image receives a score based on the distance to the nearest lane marking, and these scores are used to compute the matching score of each particle. The system uses partitioned sampling (two-step sampling and measurement using subsets of the state space, achieving a multiresolution effect), importance sampling, and initialization samples (completely random samples from the whole parameter space), which cope faster with lane discontinuities. In [4], we find a lane detection system that uses the particle-filtering framework to fuse multiple image cues (color, edges, and Laplacian of Gaussian). For each cue, a comparison method between the image data and the particle is designed, the likelihood is computed, and then, the likelihoods are combined by multiplication. This solution also uses initialization samples for faster lane relocation and additional sampling around the best-weighted particles for improvement of accuracy.

The much simpler way in which a particle filter handles the measurement information allows the use of a wider range of cues. Such is the case of the lane detector for country roads, as presented in [5], where the image space is divided into road and nonroad areas and each pixel in these areas contributes to the final weight by its intensity, color, edge, and texture information. The likelihood of each feature value to belong to either road or off-road areas is computed using trained histograms, thus allowing a non-Gaussian multimodal probability density not only for the lane state but for the measurement as well. The work presented in [11] also shows the value of the particle-filtering technique for heterogeneous cue fusion, when image information is fused with a Global Positioning System and map information for long-distance lane estimation. In [12], the authors described a system that uses a hybrid approach, i.e., combining lane border hypotheses generated using a Random-Sample-Consensus (RANSAC)-type algorithm with hypotheses from a particle filter and then using further probabilistic reasoning to choose the best border pair to delimit the lane.

V. SOLUTION OUTLINE

The system continuously evaluates the state of the lane by means of a set of particles. There is no initialization phase; therefore, each cycle is run exactly in the same way, as shown in

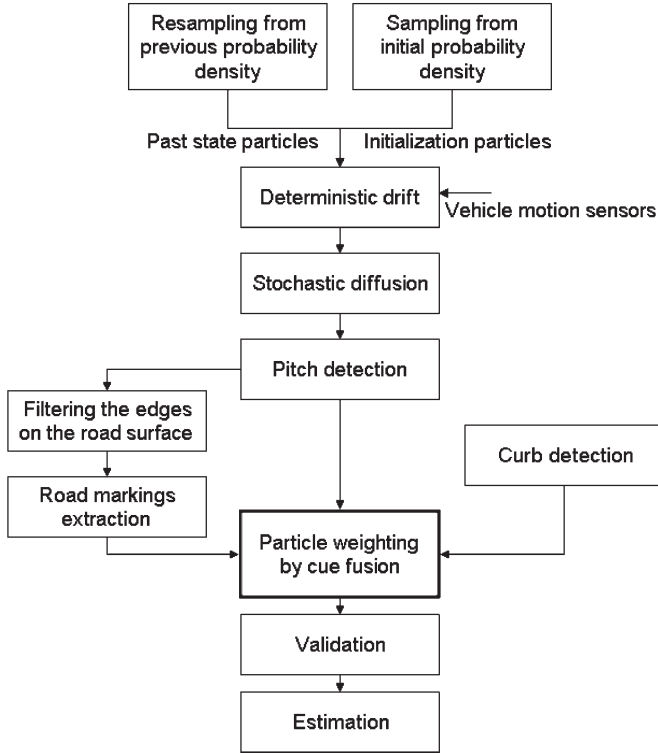


Fig. 6. Lane-detection algorithm outline.

Fig. 6. The cycle starts with particle resampling, which is done partly from the previous particle distribution and partly from a generic distribution that covers all lane geometries to cover the possible discontinuities that may arise. The deterministic drift is applied to all particles, taking into account ego motion parameters, such as speed, yaw rate, and frame timestamps; then, stochastic diffusion will alter each particle in a random way.

Pitch detection is done independently of the particle system using a probabilistic approach. The value of the detected pitch is set to each particle. The pitch value is also used to select the road features, which are then used to weigh the particles.

A validation step ensures that the particles are locked on a lane, and if this step succeeds, a lane representation is estimated.

VI. ALGORITHM DESCRIPTION

A. Lane Particles

The lane state probability density is described at a given time t by a set of N weighted particles $p(\mathbf{x}) \approx \{\mathbf{x}_t^i, \pi_t^i, i = 1, \dots, N\}$. Particle value \mathbf{x} is a lane state hypothesis in the form of a lane description vector.

The coordinate system that is used has the point of origin on the ground in front of the ego vehicle, relatively centered to the width of the car. The X -axis is positive toward the right of the vehicle, the Y -axis is positive toward the ground, and the Z -axis is positive along the forward direction. The lane is a surface that stretches forward and is bounded by two delimiting curves. The X coordinate of the delimiting curves depends on the lane parameters, the chosen distance Z , and the delimiter type t (left or right), i.e.,

$$X = h(Z, t).$$

We will denote the preceding equation as the horizontal profile of the lane. The lane parameters that affect function h will be denoted as horizontal profile parameters (such as the horizontal curvature).

In the same way, we can describe the variation of the Y coordinate of each of the delimiters, with the equation of the vertical profile of the lane, i.e.,

$$Y = v(Z, t).$$

The lane-tracking system was designed in a modular fashion, with the equations for the vertical and horizontal profiles being easily configurable. The measurement function is independent of the 3-D model, as long as the sets of 3-D points for the delimiters are available. We have found that, for the majority of the cases, the following set of parameters was sufficient:

$$\mathbf{x}_t^i = \begin{bmatrix} W - \text{lane width} \\ C_H - \text{horizontal curvature} \\ C_V - \text{vertical curvature} \\ X_C - \text{lateral offset} \\ \alpha - \text{pitch angle} \\ \gamma - \text{roll angle} \\ \psi - \text{yaw angle} \end{bmatrix}.$$

Due to the configurable nature of the system, we have been able to experiment with several other models and parameter sets. A model that included a width variation parameter has successfully been tested in highway scenarios (the results section includes the tests done with this model), but the simpler model previously described has proven to be more reliable in urban scenarios. A quite powerful argument against the use of a very complex lane representation model is that the visibility range is quite limited due to the camera focal distance and the complexity of city traffic.

B. Prediction

Before prediction can be applied, the past state described by the particle set has to be resampled into particles of equal weight. A fraction $R = 0.1N$ of the particles will be selected from a uniform probability distribution spanning the whole range of possible lane parameters. These particles account for the probability that the currently tracked lane can be erroneous or that a better lane candidate appears, such as in the case of lane change or road forking.

Each particle is transformed via prediction, which is achieved by applying the following equation:

$$\bar{\mathbf{x}}_t^i = \mathbf{A}_t \hat{\mathbf{x}}_{t-1}^i + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t$$

$$\mathbf{A}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{s_t^2}{2} & 0 & 1 & 0 & 0 & s_t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -s_t & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{B}_t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{s_t^2}{2} \\ 0 \\ 0 \\ s_t \end{bmatrix}$$

$$\mathbf{u}_t = c_t. \quad (4)$$

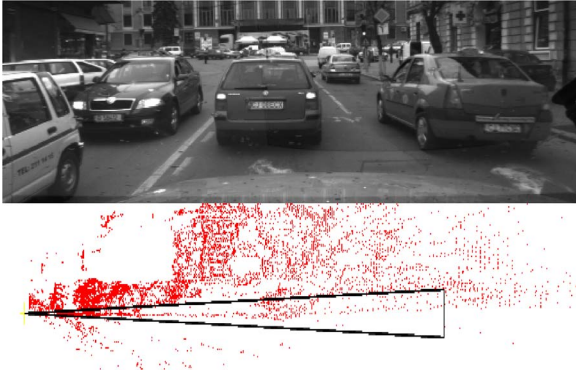


Fig. 7. Complex city scene with road, cars, and walls and a side view of the reconstructed 3-D points. The possible domain of pitch variation is highlighted.

Matrix \mathbf{A}_t is the linear transformation that encodes the way the lane evolves in time in the absence of any input from the driver, and \mathbf{B}_t is the matrix that relates the driver input to the lane evolution. The input consists of c_t , which is the curvature of the vehicle's trajectory derived from the yaw rate. Matrices \mathbf{A}_t and \mathbf{B}_t depend on space s_t traveled by the vehicle between measurements.

Part $\mathbf{A}_t \hat{\mathbf{x}}_{t-1}^i + \mathbf{B}_t \mathbf{u}_t$ is the deterministic part of the prediction, when motion laws are applied and each possible past lane configuration is clearly mapped onto a present configuration. In addition to the deterministic part, each particle's position is altered by a random value \mathbf{w}_t , which is drawn from a Gaussian distribution of zero mean and covariance matrix \mathbf{Q}_t . Covariance matrix \mathbf{Q}_t is obtained by scaling a fixed matrix \mathbf{Q}_0 , which is calibrated for a time of 100 ms between frames, with the actual elapsed time between measurements (as the frame rate is not fixed). This is natural, as a longer time between measurements allows the lane to deviate more from the predicted configuration.

C. Pitch Detection

Pitch detection has to be handled somehow differently outside of the particle filtering framework due to the following reasons: The pitch does not track well (is not very predictable), and pitch selection influences the measurement data, which are selected from the 3-D setpoints knowing the pitch angle (see Fig. 7).

Assuming that the origin of the center of coordinates is at ground level, immediately in the front of the car, it can be assumed that, for about 10–20 m, the road seen from one side will be a line passing through this origin. This line is defined by the pitch angle alone. Similarly to our previous version of the stereovision-based lane detection [7], the process of pitch detection starts by building a polar histogram that counts the points along each line passing through the origin in the lateral projection (distance–height plane). The lines correspond to discrete values of the pitch angle, which are spaced at 0.1° and range from -2° to 2° . The algorithm for polar histogram building is given here.

Initialize polar histogram $H(index)$ to 0 for each index.

For each 3-D point p

If $distance(p) > Limit$, go to the next point.

Find the angle of the line passing through p and the origin, i.e.,

$$\alpha_p = \tan^{-1} \frac{height(p)}{distance(p)}. \quad (5)$$

If $\alpha_p > 2^\circ$ or $\alpha_p < -2^\circ$, go to the next point.

Find the index of α_p in the polar histogram, i.e.,

$$index_p = \frac{\alpha_p + 2^\circ}{0.1^\circ}.$$

Increment the polar histogram by a variable amount, taking into account the variability of the point density with the distance, i.e.,

$$H(index_p) = H(index_p) + \frac{distance(p)^2}{K}. \quad (6)$$

End For

The difference from the previous pitch detection method is how we process this polar histogram. Previously, we found the maximum of the histogram and then scan the histogram bottom up until a value that is greater or equal to two thirds of the maximum was found. The reasoning behind this approach is that the road is the first structure of a substantial number of points encountered scanning the scene from bottom up, and the “substantial” part is relative to the scene. The problem with the previous approach is that it is hard to justify its correctness, and one can imagine some rare situations when it would fail. For the current lane detection algorithm, a probabilistic approach is used, which describes better relations between the real-world and possible pitch values. This means that, for each of the pitch candidates α_{index} , we will approximate probability density $p(\alpha = \alpha_{index})$, given the available information.

There are several assumptions that will govern the process of probability computation. The first assumption is that pitch history does not matter, as the pitch variation is mostly due to imperfections in the road surface, which are not easy to predict. (One can argue that an oscillatory model of the pitch variation can be used, but it would introduce a constraint that can lead to wrong estimations if not properly calibrated.) This means that the pitch probability density will be derived from current measurements alone, i.e.,

$$p(\alpha|y_1, y_2, \dots, y_t) = p(\alpha|y_t). \quad (7)$$

The second assumption is that there is no prior probability density; therefore, the probability density of the pitch variable is directly proportional to the measurement likelihood, i.e.,

$$p(\alpha|y_t) \propto p(y_t|\alpha). \quad (8)$$

The measurement is composed of two cues, which were derived from two assumptions about the 3-D road points seen in the lateral projection.

- 1) The road points should be nearly collinear.
- 2) Most of the points in the 3-D space are above the road surface.

