

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

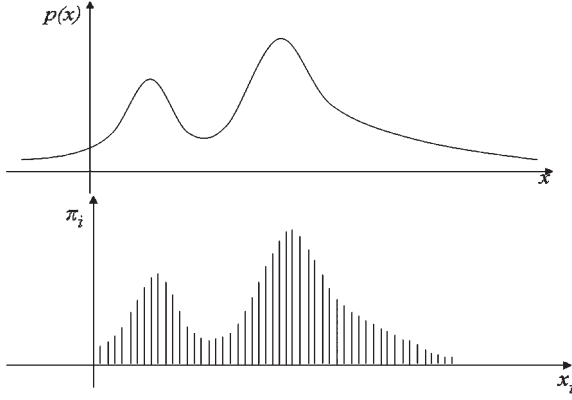


Fig. 1. Analogy between a probability density function and a set of weighted samples.

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

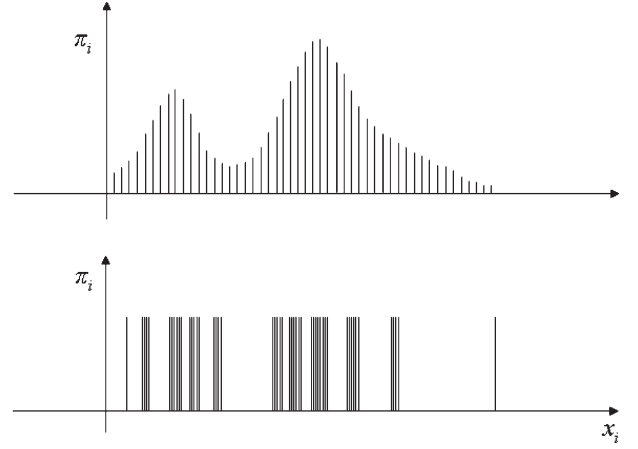


Fig. 2. Same probability density function approximated by weighted and weightless particles.

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).

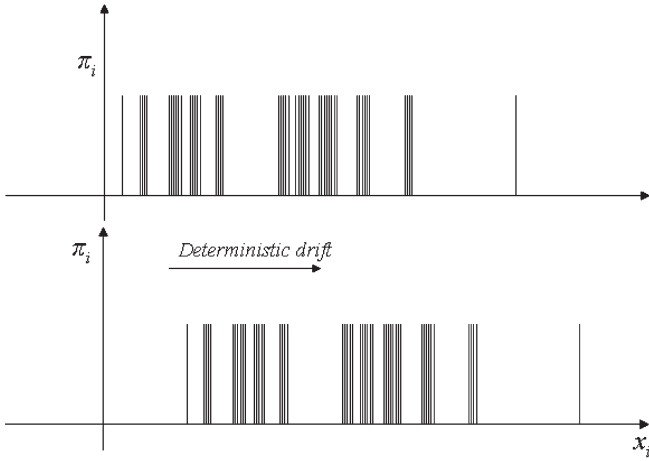


Fig. 3. Deterministic drift using weightless particles.

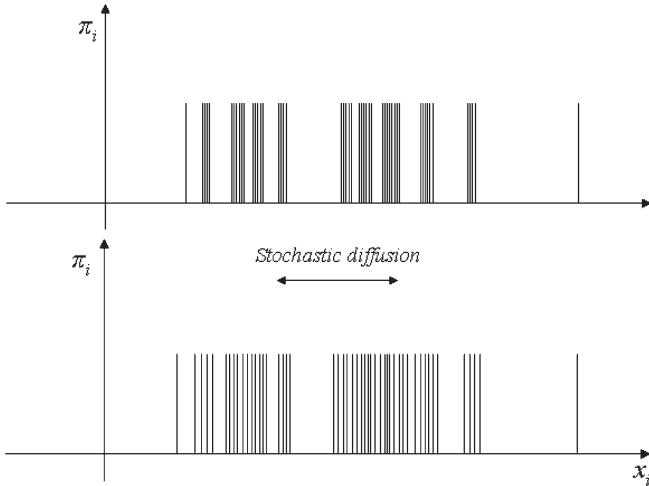


Fig. 4. Stochastic diffusion using weightless particles.

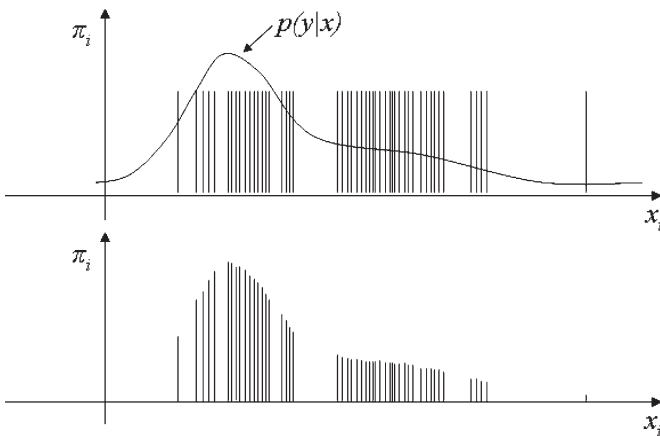


Fig. 5. Weightless particles are weighed by measurement.

IV. RELATED WORK

Lane estimation through Kalman filtering was pioneered by Dickmanns and Mysliwetz [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

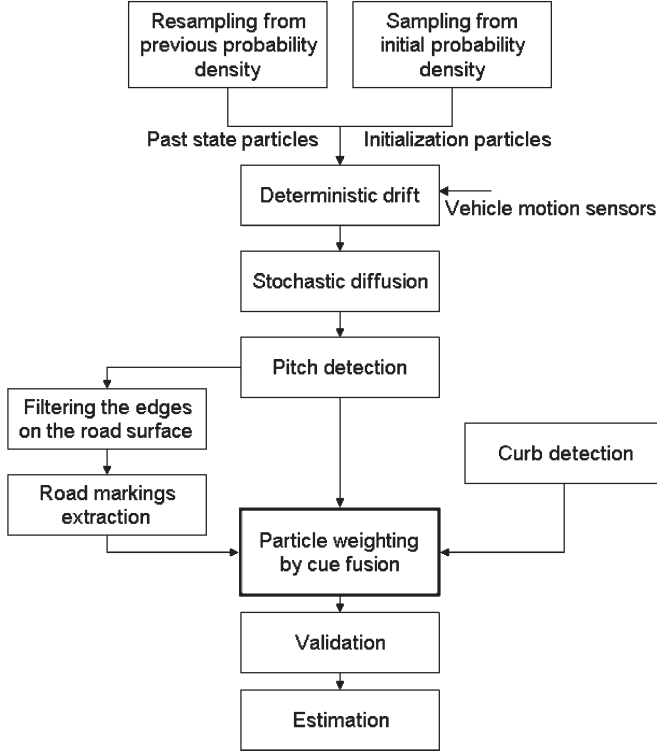


Fig. 6. Lane-detection algorithm outline.

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:

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

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

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

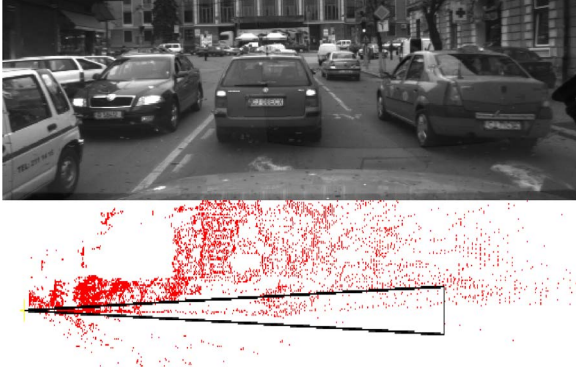


Fig. 7. Complex city scene with road, cars, and walls and a side view of the reconstructed 3-D points. The possible domain of pitch variation is highlighted.

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.

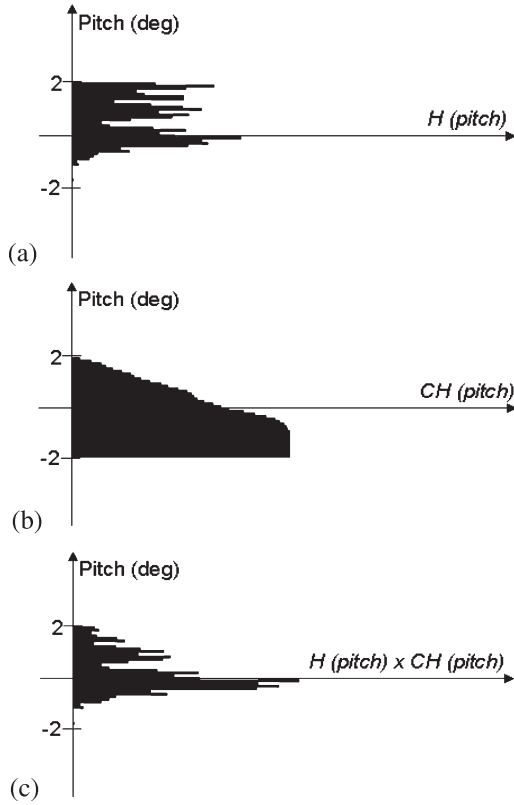


Fig. 8. Combining the cues for pitch. (a) Polar histogram. (b) Cumulative histogram. (c) Combination.

The cue corresponding to the first assumption has the likelihood directly proportional to polar histogram H , and the likelihood for the cue of the second assumption is directly proportional to a cumulative histogram derived from H , i.e., CH .

$$p(y_H | \alpha = \alpha_{\text{index}}) \propto H(\text{index}) \quad (9)$$

$$p(y_{CH} | \alpha = \alpha_{\text{index}}) \propto CH(\text{index}) \quad (10)$$

$$p(\alpha = \alpha_{\text{index}} | y_t) \propto H(\text{index}) CH(\text{index}). \quad (11)$$

The pitch candidate with the highest likelihood, which corresponds to the highest value of the histogram product, is chosen as the pitch estimate. Fig. 8 shows the effect of pitch cue fusion, leading to a clear maximum, even if the complex scene leads to multiple strong peaks in the polar histogram. Another estimation method that was taken into consideration was the weighted sum of the pitch candidates, but the maximum leads to better results.

The value vectors \mathbf{x} of the predicted particles are modified by setting their pitch field to the estimated pitch value. This pitch value is also used for selecting the road points from the available 3-D point set to perform the next stages of the measurement.

D. Mapping the Particles in the Image Space

Pitch detection is the only part of the measurement process that happens in the 3-D space, and for the next stages, the particles have to be compared to image space measurement data. To achieve the comparison, from each particle value of the form

$\bar{x}_t^i = (W, C_H, C_V, X_C, \alpha, \gamma, \psi)^T$, a measurement space vector is generated, i.e., $\bar{y}_t^i = (v_1, \dots, v_P, u_{L,1}, \dots, u_{L,P}, u_{R,1}, \dots, u_{R,P})$. Values v are coordinates of the image lines, and values u are coordinates of the image columns. The v values are common to the left and right delimiters. P is the number of points chosen to describe each lane delimiter in the image space.

To derive \bar{y}_t^i from \bar{x}_t^i , three steps have to be taken.

- 1) Generate P points in the 3-D space for each lane delimiter. The points will be equally spaced on distance axis Z , and their X and Y coordinates (lateral and height) will be given by the horizontal and vertical profiles of the lane. The nearest points will start at distance Z_B , which is the closest distance that allows the road to be visible to our system. The points will span a detection distance D . Detection distance D is variable, and its adjustment is based on the vehicle's speed. The rationale behind this decision is that a longer distance is needed if the vehicle travels at high speeds, which is usually on straight or low-curvature roads, but a shorter distance is needed at slow speeds to handle narrower curvatures. Distance D covers a second of vehicle movement at the current speed, but no shorter than 5 m and no longer than 60 m.
- 2) Project the 3-D points in the image space using the camera parameters. For each lane delimiter, a chain of unevenly spaced points will be obtained.
- 3) Intersect the segments obtained by linking the projected points for each side with a set of evenly spaced horizontal lines. The points of intersection are the points that will form the particle representation in the image space \bar{y}_t^i .

E. Visual Cues

After the pitch angle has been detected from the 3-D point set, a rough approximation of the road geometry can be made based on this angle alone. The rough approximation is used for road point selection. The image edges corresponding to these 3-D points form our first measurement cue.

The lane-marking edge points are detected using an algorithm based on the tried-and-tested dark-light-dark transition detection principle [8]. In addition to the lane markings, another high-priority lane-delimiting feature is the curb, and the curbs are detected using height variations in a dense stereovision map [9] and then converted into image edges. Due to the fact that lane markings and curbs are of similar priorities, they are inserted in a common "special edge" map, which represents the second lane measurement cue.

To allow comparison between the particles and the measurement, each cue map (road edges or special edges) undergoes distance transformation (see Fig. 9).

F. Particle Weighing by Measurement

Given the *a priori* probability density encoded in the distribution of the particle values throughout the state space, it is now time to compute the posterior probability density, which will encode all the knowledge about the lane state that we are able to extract from the sequence of measurements up to the current time t . This is achieved by assigning new weights to the

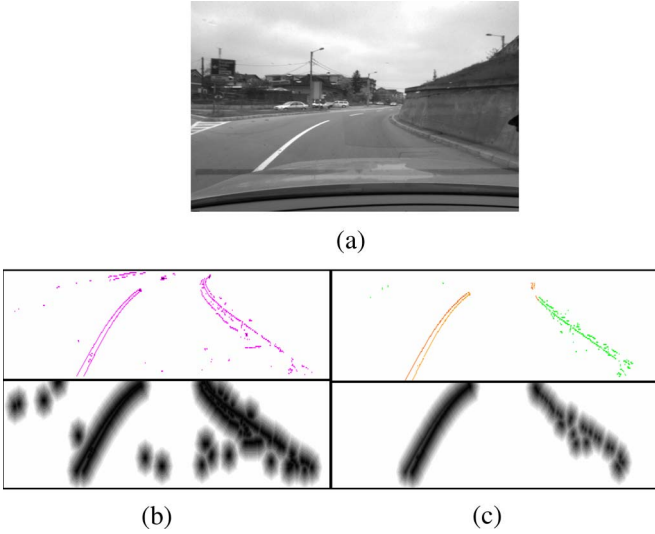


Fig. 9. Visual information. (a) Original grayscale image. (b) Edges corresponding to 3-D points contained in the road surface and the associated distance transform image. (c) Markings and curbs and their associated distance transform image.

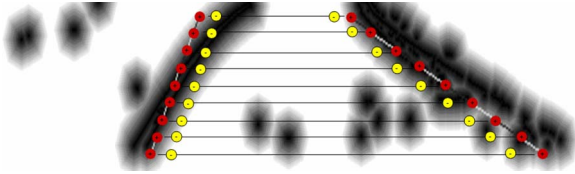


Fig. 10. Positive and negative points. The positive points are lane boundary points, and the negative points are points inside the lane area.

predicted particles, which are proportional to the measurement likelihood, given the state hypothesis, i.e.,

$$\pi_t^i = p(\mathbf{y}_t | \mathbf{x}_t = \mathbf{x}_t^i). \quad (12)$$

The measurement likelihood is obtained by multiplying the edge likelihood and the marking/curb likelihood under the measurement independence assumption, i.e.,

$$p(\mathbf{y}_t | \mathbf{x}_t = \mathbf{x}_t^i) = p(\text{road_edges} | \mathbf{x}_t = \mathbf{x}_t^i) \cdot p(\text{mark_curb} | \mathbf{x}_t = \mathbf{x}_t^i). \quad (13)$$

To compute the likelihood of the two measurement cues, a distance between the lane state hypothesis and the measurement has to be computed. The distance transformation of the two edge images now becomes very helpful.

Ideally, the lane hypothesis boundaries' projections in the image space have to exactly fit on the edges of the visual cues. In addition, the area inside the hypothetic lane projection has to be as free of edges as possible. To test these two conditions, two sets of points are used: 1) the positive points, which are points belonging to the lane delimiters' projection in the image space, and 2) the negative points, which are points that are near the borders and reside inside the projected lane area (see Fig. 10).

The positive points will generate the positive distance; this is obtained by averaging the distance-transformed pixel values at these points' coordinates. The distance corresponding to the negative points is the complement of the distance-transformed

image at these points' coordinates. The two distances are combined by weighted averaging, i.e.,

$$D^M = \frac{\alpha D^M(+) + \beta D^M(-)}{(\alpha + \beta)}. \quad (14)$$

The value of weight parameters α and β has been set to 2 and 1, respectively, through trial-and-error experiments.

Now, for each measurement M , the measurement likelihood is computed using a Gaussian distribution to relate the probability to the distance between the prediction and the visual data, i.e.,

$$p(M | \mathbf{x}_t = \mathbf{x}_t^i) = \frac{1}{\sigma_M \sqrt{2\pi}} e^{-\frac{D_M^2}{2\sigma_M^2}}. \quad (15)$$

Each particle will receive as weight the product of the two likelihoods. At this step, the particles that show a degenerate lane, such as a lane that is too narrow, too wide, or too far from the vehicle's position, will receive a null weight, preventing them for spawning new candidates in the next cycle. The final step is to normalize the new weights so that their sum is 1, and the system is ready to perform a new tracking cycle.

G. Lane Validation

Unlike a KF lane-tracking solution, the particle-filtering system does not need initialization or measurement validation before track update. The particles will freely evolve, eventually clustering around the best lane estimate, if the system is properly designed and the measurements are relevant. However, the system must know when a valid lane is being tracked if it is to be used for practical purposes.

The first attempt was to analyze the particle distribution in the state space and validate the situation when the particles were reasonably clustered. However, we have observed that particles tend to cluster, even in the presence of weak measurements, and this clustering does not guarantee the validity of the final estimate.

A much more successful solution is to compare the average weight of the predicted (from sampled) particles against the average weight of the completely random particles that are added in the sampled set. Recalling that N denotes the total number of particles and R denotes the number of totally random particles and that the random particles are inserted at the head of the particle list (without altering the probability density), a quality factor is defined as

$$q = \frac{R \sum_{i=R+1}^N \pi_t^i}{(N - R) \sum_{i=1}^R \pi_t^i}. \quad (16)$$

If q is higher than a threshold, the lane track is considered valid for output, and a lane state vector will be estimated from the particle set. A high quality factor means that the visual cues support the predicted lane to a much higher degree than some completely random lane parameters, which supports the hypothesis that the lane approximated by the particles is correct (agrees with the observation). The threshold that we found to work best is 10.

If the quality factor indicates a valid lane, the parameters of this lane are estimated by a weighted average of the particle values. Only particles that have a weight that is higher than average are considered for estimation.

VII. TEST AND RESULT

A. Comparison With a KF Solution

The stereovision-based particle-filtering lane detection system has been designed mainly to improve the handling of difficult situations, when the KF solution had significant problems. Even if the scenarios posing problems to a KF solution can be various, they can be summarized by a single term, i.e., “discontinuous road” (which is sometimes called road singularity). The most common situations that can be regarded as road discontinuities are lane appearance and disappearance, lane change maneuvers, lane forking/joining, sharp changes of direction, sharp changes of curvature, and temporary sensor failure due to internal or external conditions. (The most often problem encountered is image saturation.)

A KF solution has problems with road discontinuities due to four characteristics.

- 1) There is only one possible lane configuration that is tracked at one moment in time.
- 2) The current state is used to predict search areas for the next detection; this is a feature that drops all measurements that indicate a road discontinuity.
- 3) The system requires time to drop a track and time to initialize a new track.
- 4) Initializing a new track means running detection algorithms for the whole image, without the benefit of a reduced search region.

We have tested the particle-filtering solution in scenarios containing the specified problems, and the system has shown five behaviors.

- 1) Lane appearance and disappearance: Due to the fact that there is no detection in the classical sense, no additional time is needed to start or drop a track. The particles will cluster around the best lane visual information, and the output is validated after two to three frames.
- 2) In lane-changing maneuvers, there are two aspects of our algorithm that make the transition as smooth as possible: 1) the ability to track multiple hypotheses and 2) the use of random particles to keep an eye on new tracks. The random particles will seed a new cluster, and due to the motion of the vehicle toward the new lane, the particles of the new cluster will receive increasingly more weight until the old lane is left behind. When the lane change maneuver is completed, the new lane is already tracked.
- 3) The forking/joining situations are handled in the same way as the lane change maneuvers. The system is always ready to track a lane that has a better chance of being the right lane.
- 4) Sharp changes of curvature are handled either by generating the right hypothesis fast enough to cope with the change, similarly to the way situations 2 and 3 are handled or, if this is not possible due to the severity of

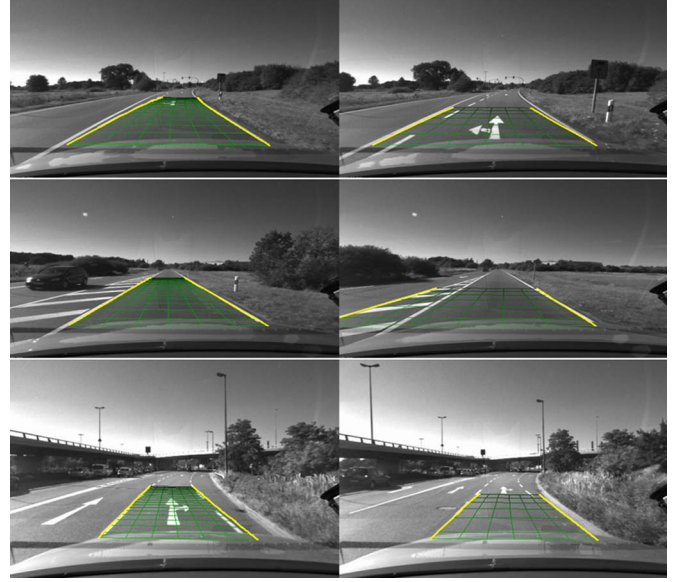


Fig. 11. Samples of side-by-side comparison. (Left) Particle-filtering solution. (Right) KF solution.

conditions, by fast recovery once the discontinuity has been passed, with the situation of false estimation being avoided in either case.

- 5) Due to the fact that there is no track reset in the particle filter system, sensor failures are uniformly treated by the tracker. The particles will begin to spread as long as there is no information to cluster them, and when the sensor goes back online, the particles will begin clustering again. If, during this time, they still describe a valid lane or not, the lane-validation system will decide, independently of the tracking process itself.

A dynamic qualitative comparison between the behavior of the method described in this paper and a KF solution is provided in the movie file *pf-kf.avi*. In the left half of the frame, the particle-filtering solution is displayed, and in the right half, one can see the results of the KF (see Fig. 11).

B. Pitch Angle Evaluation

To evaluate the results of the pitch detection method that we proposed in this paper, a sequence of images of high pitch variation was selected. The road geometry forces the vehicle to change the pitch ranging from -3° to $+4^\circ$. Because there is no ground truth for the pitch value, we have chosen to compare the pitch results with the pitch estimated by simple averaging of the heights of the points directly in front of the vehicle in a narrow 3-D window (which measures 1 m wide and 7 m long). Due to the fact that we ensured the sequence to be obstacle free in the selected window, the 3-D points located there are (mostly) in the road plane. The pitch detection that we wish to evaluate does not benefit from the fact that the road is obstacle free, because the 3-D points used in the algorithm are not restricted to a narrow window, and on the sides of the road, there are a number of obstacle features. The graph in Fig. 12 shows the two pitch values (in degrees) against the frame number. The pitch-detection system results are shown with a dotted line, and the comparison “ground truth” is shown with a continuous line. The

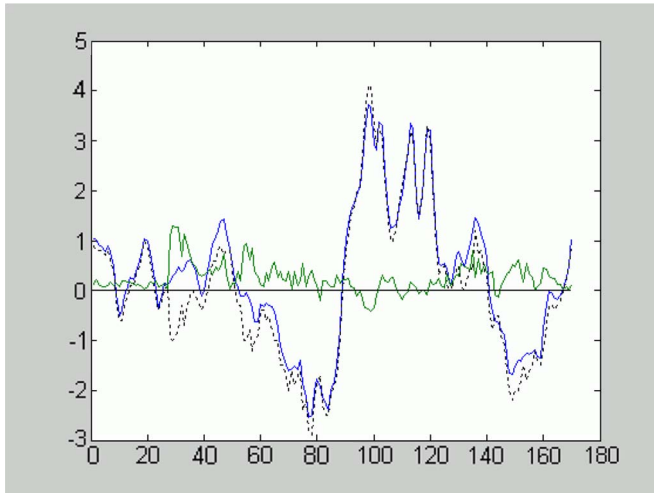


Fig. 12. Pitch angle comparison.

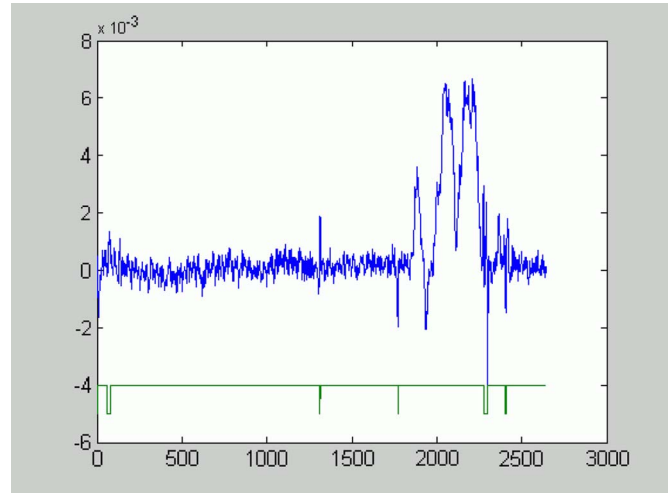
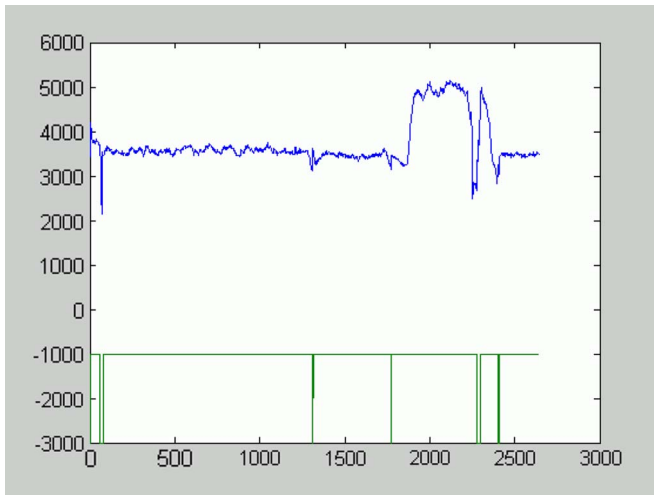
Fig. 14. Highway behavior. Horizontal curvature (in m^{-1}) versus frame number.

Fig. 13. Highway behavior. Width (in millimeters) versus frame number.

difference between the two pitch values is also shown. From the graph, it is clear that the pitch detection algorithm, which works on an unrestricted set of data, closely follows the pitch value obtained from the restricted data set. The errors are within the uncertainty given by the errors of stereo reconstruction and can affect either our pitch estimation or the “ground truth.”

The behavior of the pitch estimator can be viewed in the file *pitching.avi*, where the side projection of the scene (3-D points and lane surface) is superimposed on the perspective image. The horizontal profile fit is not perfect as the lane delimiters are poor, and the width of the lane is higher than our acceptance threshold (6 m).

C. Highway Performance Evaluation

The performance of the lane detection system on highways is evaluated using a sequence of 2644 frames acquired at about 10 frame/s (the acquisition frame rate is not constant, but each frame is timestamped, and the algorithms are able to handle the variable frame rate), which means about 4.4 min of driving. The sequence contains lane changes, highway exits and reentry, and impaired visibility due to rain and windshield wipers. Figs. 13–16 show the evolution of some of the lane parameters

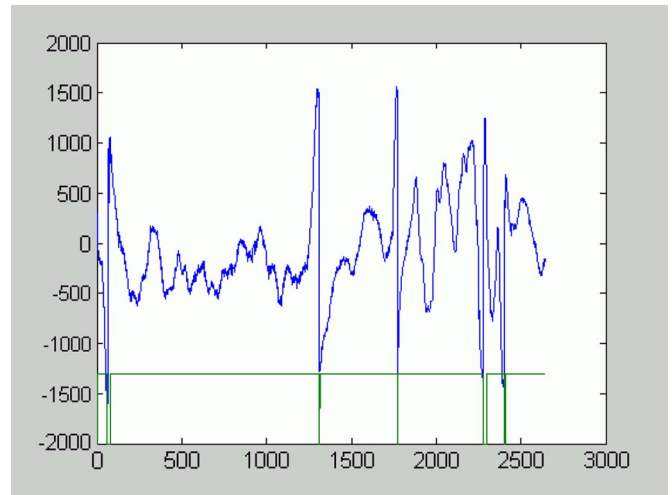


Fig. 15. Highway behavior. Lateral offset (which is the distance of the vehicle from the lane center expressed in millimeters) versus frame number.

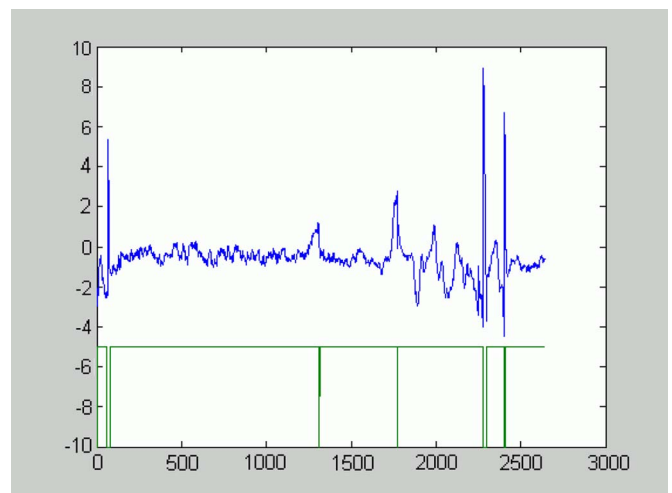


Fig. 16. Highway behavior. Yaw angle (in degrees) versus frame number.

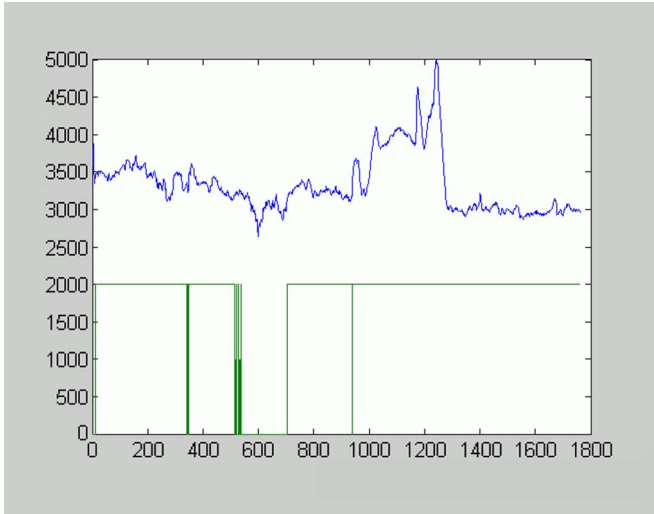


Fig. 17. Urban behavior. Width (in millimeters) versus frame number.

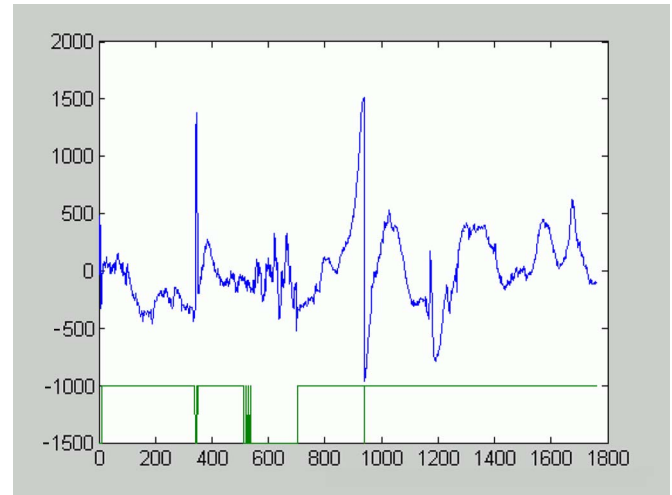


Fig. 19. Urban behavior. Lateral offset (which is the distance of the vehicle from lane center expressed in millimeters) versus frame number.

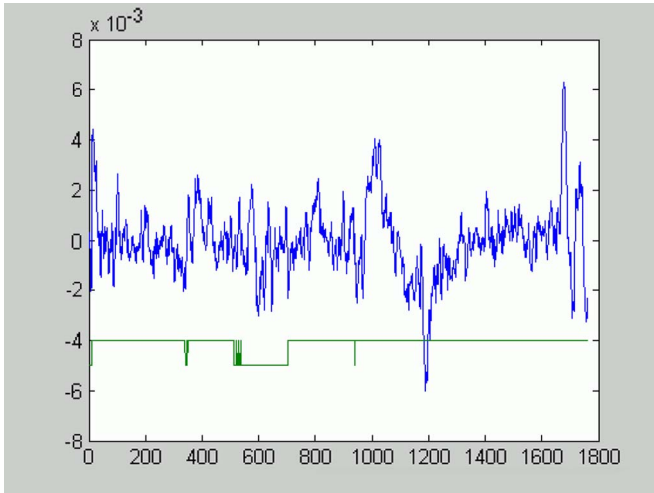


Fig. 18. Urban behavior. Horizontal curvature (in m^{-1}) versus frame number.

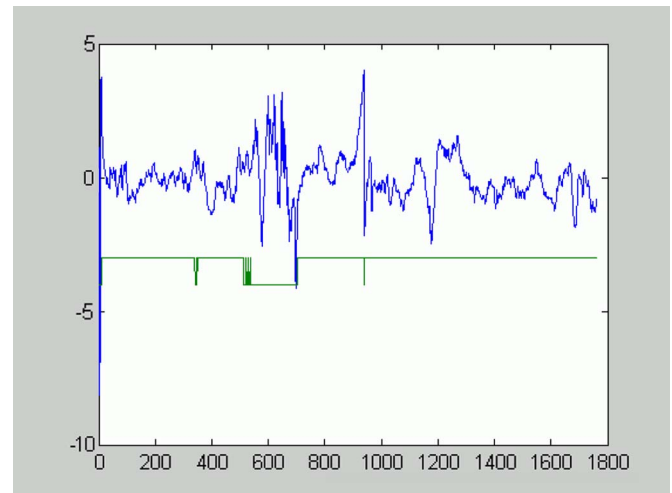


Fig. 20. Urban behavior. Yaw angle (in degrees) versus frame number.

(width, curvature, lateral offset, and yaw angle, which are the parameters needed for a top-view representation of the lane). The validation of the lane detection is shown in the graphs as a binary signal, with a high value meaning that the lane is valid. The number of “valid” frames is 2585, indicating a detection rate of 97.77.

The sequence and the detection results, in perspective projection and bird’s-eye view, can be seen in the file *highway.avi*. The purple points in the bird’s-eye view are the 3-D points provided by stereovision that are marked as road points by filtering against the detected pitch and a Canny edge detector.

D. Urban Performance Evaluation

The performance of the lane detection system in the city is evaluated using a sequence of 1763 frames acquired at about 10 frame/s (variable), which means about 3 min of driving. The sequence contains lane changes, lane forking and joining, passing through a tunnel, and passing through intersections. Figs. 17–20 show the evolution of some of the lane parameters (width, curvature, lateral offset, and yaw angle). The validation

of the lane detection is also shown in the graphs. The number of “valid” frames is 1559, indicating a detection rate of 88.43%. The presence of a low-visibility tunnel in the sequence is the main reason the detection rate is so low. The sequence and the detection results, in perspective projection and bird’s-eye view, can be seen in the file *urban.avi*.

A crowded urban sequence, with poor-quality lane delimiters, a number of obstacles, driving on the lane border, etc., is available for qualitative analysis only in the file *crowded.avi*.

E. Experimental Testing: Linear Variable Width Model

Due to the high adaptability of the particle filter framework, new lane models can easily be tried. Inserting a width variation parameter [which describes the increase or decrease of width with the longitudinal distance and is required only to change the function that computes the lateral coordinates (X) of the lane delimiters with the distance (Z)], the resulting experimental system was tested on the highway sequence, at the exit/reentry moments, when the width variation was more obvious. The results can be shown in the file *varwidth.avi*. A graph that

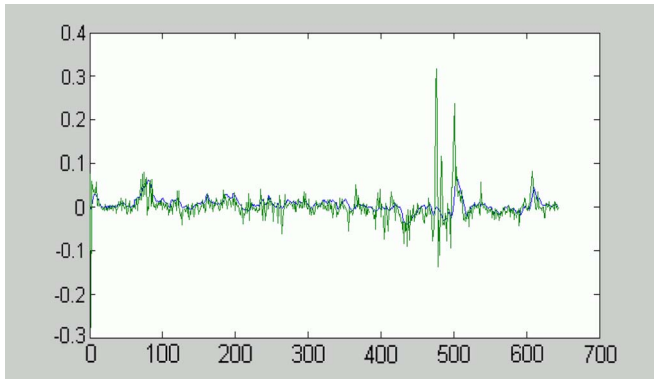


Fig. 21. Comparison between width variation estimated by lane detection and differentiation of the estimated width against the traveled space.

compares the estimated width variation with a differentiation of the estimated lane width against the traveled space is shown in Fig. 21. The modeled width variation is the smoother signal.

F. Time Performance

The time performance has been evaluated on an Intel Core 2 Duo central processing unit at 2 GHz using a single thread. The lane detection time has a fixed part that is independent of the number of particles and amounts to 9.6 ms and a time per processed particle of 0.0075 ms. Our 200-particle solution takes 11 ms to complete.

Note that the movie files showing the test results can be downloaded from http://users.utcluj.ro/~rdanescu/lane/lane_eval.htm.

VIII. CONCLUSION AND FUTURE WORK

We have presented a system that uses the advantages of stereovision and grayscale image processing through a particle-filtering framework to robustly detect lanes under difficult conditions. The system does not use detection in the classical sense. There is no track initialization or track loss; thus, the processing time is kept constant, regardless of the scenario. The system shows remarkable stability when the conditions are favorable but great capability of adaptation when conditions change.

Future work will include increasing the accuracy of the estimated parameters using more measurement cues (such as image gradient orientation) or a multiresolution approach and tracking of the side lanes. Tracking the side lanes will provide the additional benefit of reducing the detection failure time in the case of lane changes.

Due to the fact that the described method is relatively model independent, experiments with several models will be carried out to find the best compromise between generality and stability.

REFERENCES

- [1] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, Feb. 1992.
- [2] R. Aufrere, R. Chapuis, and F. Chausse, "A model-driven approach for real-time road recognition," in *Mach. Vis. Appl.*, vol. 13, Nov. 2001, pp. 95–107.
- [3] M. Isard and A. Blake, "CONDENSATION: Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.
- [4] K. Macek, B. Williams, S. Kolski, and R. Siegwart, "A lane detection vision module for driver assistance," in *Proc. IEEE/APS Conf. Mechatron. Robot.*, 2004, pp. 77–82.
- [5] U. Franke, H. Loose, and C. Knoepfel, "Lane recognition on country roads," in *Proc. IEEE Intell. Veh. Symp.*, Istanbul, Turkey, 2007, pp. 99–104.
- [6] R. Labayrade, J. Douret, and D. Aubert, "A multi-model lane detector that handles road singularities," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Toronto, ON, Canada, 2006, pp. 1143–1148.
- [7] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, and C. Pocol, "3D lane detection system based on stereovision," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Washington, DC, 2004, pp. 161–166.
- [8] R. Danescu, S. Nedevschi, M. M. Meinecke, and T. B. To, "Lane geometry estimation in urban environments using a stereovision system," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Seattle, WA, 2007, pp. 271–276.
- [9] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. B. To, "Road surface and obstacle detection based on elevation maps from dense stereo," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Seattle, WA, 2007, pp. 859–865.
- [10] B. Southall and C. J. Taylor, "Stochastic road shape estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, 2001, pp. 205–212.
- [11] P. Smuda, R. Schweiger, H. Neumann, and W. Ritter, "Multiple cue data fusion with particle filters for road course detection in vision systems," in *Proc. IEEE Intell. Veh. Symp.*, Tokyo, Japan, 2006, pp. 400–405.
- [12] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008.



Radu Danescu received the M.S. degree in computer science from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 2003, where he is currently working toward the Ph.D. degree in computer vision.

He is currently a Lecturer with the Department of Computer Science, TUCN, teaching image processing and pattern recognition. He is also a member of the Image Processing and Pattern Recognition Research Laboratory, TUCN. His research interests are stereovision- and probability-based tracking, with applications to driving assistance.



Sergiu Nedevschi (M'99) received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Cluj-Napoca (TUCN), Cluj-Napoca, Romania, in 1975 and 1993, respectively.

From 1976 to 1983, he was with the Research Institute for Computer Technologies, Cluj-Napoca, as a Researcher. In 1998, he was appointed Professor of computer science and founded the Image Processing and Pattern Recognition Research Laboratory, TUCN. From 2000 to 2004, he was the Head of the Department of Computer Science, TUCN. He is currently the Dean of the Faculty of Automation and Computer Science, TUCN. He has published nearly 150 technical papers and has edited more than ten volumes, including books and conference proceedings. His research interests include image processing, pattern recognition, computer vision, intelligent vehicles, signal processing, and computer architecture.