Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision

Radu Danescu, Sergiu Nedevschi

Abstract— **Accurate and robust lane results are of great significance in any driving assistance system. In order to achieve robustness and accuracy in difficult scenarios, probabilistic estimation techniques are needed to compensate for the errors in detection of lane delimiting features. The 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 fusion of two stereovision-based cues, a novel method for particle measurement and weighting 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 resulted solution has proven to be a reliable and fast lane detector for difficult scenarios.**

***Index terms*-Lane detection, tracking, particle filtering, cue fusion, stereovision**

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 become increasingly complex, as the targeted scenarios became increasingly 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 parameters 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, traditionally achieved by the use of the Kalman filter. The use of Kalman filter tracking has the advantage of reducing the search space, eliminating the detection outliers, and smoothing of the result.

The features that make the Kalman filter solutions smooth and efficient are the very features that cause problems when the road is not continuous. Sharp turns, lane changes, atypical road geometries pose problems to a tracker that represents the lane probability density as a Gaussian functions, and the reduction of the search space around the past results makes it difficult to handle new hypotheses, and causes detection errors to accumulate, if the search regions are drawn towards false delimiters.

Particle filtering is a novel technology for probability-based tracking, allowing multiple hypotheses 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 in order to achieve robust lane estimation results in difficult scenarios of city, highway and country roads.

R. Danescu and S. Nedevschi are with the Technical University of Cluj Napoca, Romania, e-mail: Radu.Danescu@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro, Department address: Computer Science Department, Str. C. Daicoviciu nr. 15, 400 020 Cluj Napoca, Romania, Phone: +40 264 401 457

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 $P(\mathbf{X}_i \mid \mathbf{Y}_0 = y_0,...,\mathbf{Y}_i = y_i)$, that is, to compute the conditional probability density of the state $\mathbf{X}_i$ given the sequence of measurements from the past and current frame.

Due to the fact that the tracking process must deliver result at each frame, and to the fact that a tracker should be able to function in mostly the same way for an indefinite period of time, the process of estimation of $P(\mathbf{X}_i \mid \mathbf{Y}_0 = y_0,...,\mathbf{Y}_i = y_i)$ 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. In order to achieve this, the following concepts are used:

Dynamic model: $P(\mathbf{X}_i \mid \mathbf{X}_{i-1})$, the probability of reaching some value of the random variable $\mathbf{X}_i$ given the past state $\mathbf{X}_{i-1}$, under the assumption that only the immediate past matters.

Prediction: computation of the conditional probability density of the current state given the past sequence of measurements, $P(\mathbf{X}_i \mid \mathbf{Y}_0 = y_0,...,\mathbf{Y}_{i-1} = y_{i-1})$. Given the simplification assumption that only the immediate past matters, the prediction probability values can be computed recursively, given the past results and the dynamic model:

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

Data association: At each frame i there may be several measurements available, and not all of them are equally useful. Denoting by y_i^r the r-th measurement of the frame i, the probability of this measurement being useful is expressed as $P(\mathbf{Y}_i = y_i^r \mid y_0,...y_{i-1})$. If each measurement is conditionally independent of the others (the measurement independence assumption is taken), the usefulness of each measurement can be computed as:

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

State update: the state probability density $P(\mathbf{X}_i \mid \mathbf{Y}_0 = y_0,..., \mathbf{Y}_i = y_i)$, the end result of the tracking process, is computed using Bayes' rule.

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

The equations of tracking as probabilistic inference are complex to apply in the general case. Even more, the probability densities involved impossible to represent analytically most of the time, and therefore are approximated. Approximating means either coercing them to a known probability density function, such as a Gaussian, or by maintaining a discrete numerical representation throughout the whole process. The Gaussian representation leads to the well-known Kalman filter 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, particle filtering is described in [3]. Instead of trying to approximate an unknown function analytically, 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, the sum of all weights being 1.

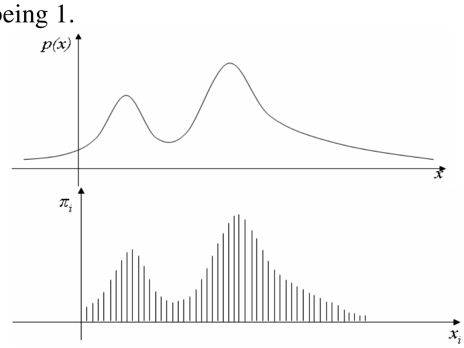

Fig. 1. Analogy between a probability density function and a set of weighted samples

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 the each particle, changing the particle representation to $\{\mathbf{x}_t^i, \pi_t^i, c_t^i, i = 1...N\}$. This parameter is defined as the sum 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 **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, while 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.

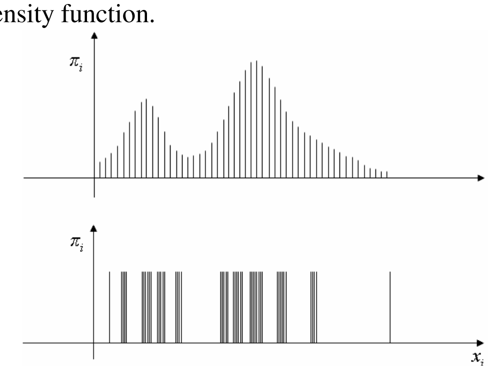

Fig. 2. Same probability density function, approximated by weighted and weightless particles

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 the 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, described by a function or a linear transformation, and partly by a random factor.

Each weightless particle 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.

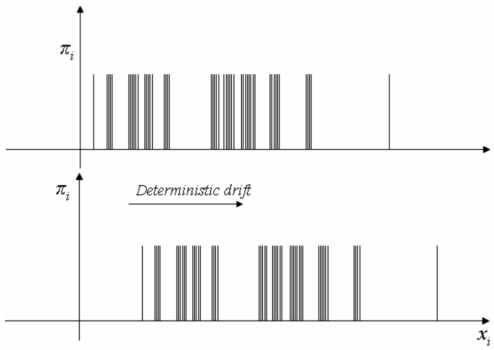

Fig. 3. Deterministic drift using weightless particles

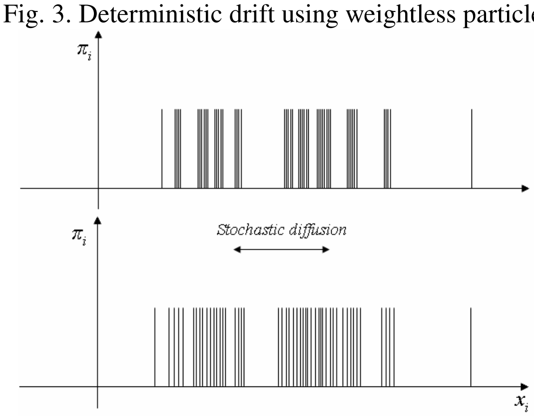

Fig. 4. Stochastic diffusion using weightless particles.

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 and the measurement. The prior state probability density is at this point completely encoded in the distribution of the weightless particles of value through the domain of possible state values. The posterior probability density function is obtained by simply weighting the particles using the likelihood of observation, $p(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}_t^i)$. Several cues can be combined in this step by multiplication, using the cue conditional independence assumption, if applicable.

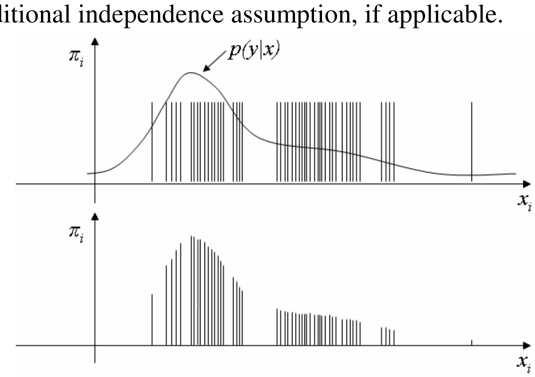

Fig. 5. Weightless particles are weighted by measurement

IV. Related Work

Lane estimation through Kalman filtering was pioneered by Dickmanns [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 Kalman filter is not the best choice for tracking discontinuous roads.

A shift towards 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 Kalman filter, because it usually consists of a comparison between the particle and the image data, from which a weight is derived, and therefore no complex detection algorithms are required. However, the measurement step is executed for each particle, so the simplicity is essential for adequate time performance. [10] presents 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, Laplacian of Gaussian). For each cue a comparison method between 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, presented in [5], where the image space is divided into road and non-road areas and each pixel in these areas contribute 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 also for the measurement. The work presented in [11] also shows the value of the particle filtering technique for heterogeneous cue fusion, when image information is fused with GPS and map information for long distance lane estimation. In [12], the authors describe a system that uses a hybrid approach, combining lane border hypotheses generated using a 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 depicted in figure 6. The cycle starts with particle resampling, which is done partially from the previous particle distribution and partly from a generic distribution that covers all lane geometries, in order to cover the possible discontinuities that may arise. The deterministic drift is applied to all particles, taking into account the ego motion parameters such as speed, yaw rate and frame timestamps, and 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 weight the particles.

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

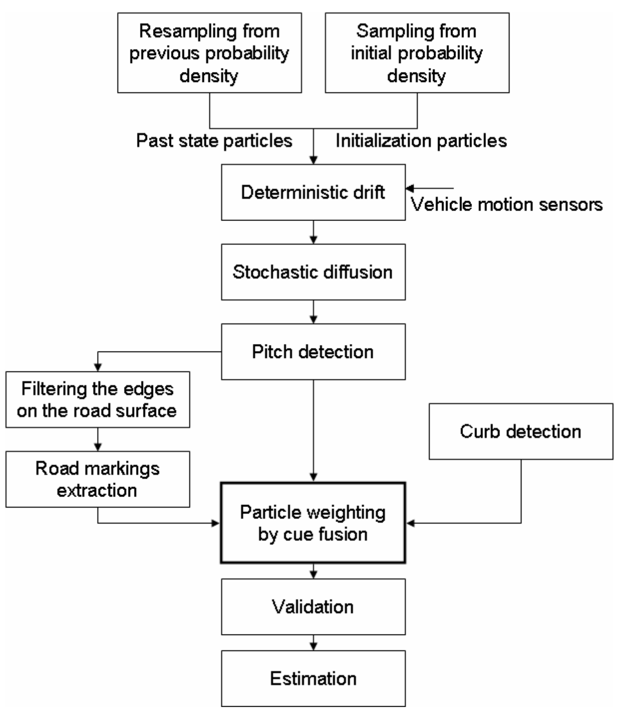

Fig. 6. Lane detection algorithm outline

VI. ALGORITHM DESCRIPTION

A. The Lane Particles

The lane state probability density is described at a given time t by a set of N weighted particles $p(\mathbf{x}) \approx \{\mathbf{x}_t^i, \pi_t^i, i = 1...N\}$. The 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, centered relatively to the width of the car. The *X* axis is positive towards the right of the vehicle, the *Y* axis is positive towards the ground, and the *Z* axis is positive along the forward direction. The lane is a surface stretching forward, 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).

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

We'll denote the above equation the horizontal profile of the lane. The lane parameters that affect the 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.

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

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

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

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 been successfully tested in highway scenarios (the results section includes the tests done with this model), but the simpler model described above 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 to the complexity of the 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=0.1 N 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, achieved by applying the following equation:

$$\bar{\mathbf{x}}_t^i = \mathbf{A}_t \hat{\mathbf{x}}_{t-1}^i + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t \qquad (4)$$

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

The 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, the curvature of the vehicle's trajectory, derived from the yaw rate. Matrices $\mathbf{A_t}$ and $\mathbf{B_t}$ depend on the space s_t traveled by the vehicle between measurements.

The 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 into a present configuration. Besides the deterministic part, each particle's position is altered by a random value $\mathbf{w_t}$, drawn from a Gaussian distribution of zero mean and covariance matrix Q_t. The covariance matrix $\mathbf{Q}_t$ is obtained by scaling a fixed matrix $\mathbf{Q}_0$, 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: pitch does not track well (is not very predictable), and pitch selection influences the measurement data, selected from the 3D set points knowing the pitch angle.

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

Assuming 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 meters, 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, spaced at 0.1 degrees, ranging from -2 to 2 degrees. The algorithm for polar histogram building is the following:

Initialize polar histogram *H (index)* to 0, for each index
For each 3D point *p*
  If *distance* (*p*)> *Limit* go to next point
  Find the angle of the line passing through *p* and the origin

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

  If $\alpha_p > 2^\circ \ or \ \alpha_p < -2^\circ$ go to next point
  Find the index of α_p in the polar histogram

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

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

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

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 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 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 the possible pitch value. This means that for each of the pitch candidates α_{index} we'll approximate the 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 due mostly to imperfections in the road surface, imperfections that are not easily 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.

$$p(\alpha \mid y_1, y_2, ..., y_t) = p(\alpha \mid y_t) \tag{7}$$

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

$$p(\alpha \mid y_t) \propto p(y_t \mid \alpha) \tag{8}$$

The measurement is composed of two cues, derived from the following assumptions about the road points 3D seen in the lateral projection:

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

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

$$p(y_H \mid \alpha = \alpha_{index}) \propto H(index) \tag{9}$$

$$p(y_{CH} \mid \alpha = \alpha_{index}) \propto CH(index) \tag{10}$$

$$p(\alpha = \alpha_{index} \mid y_t) \propto H(index)CH(index) \tag{11}$$

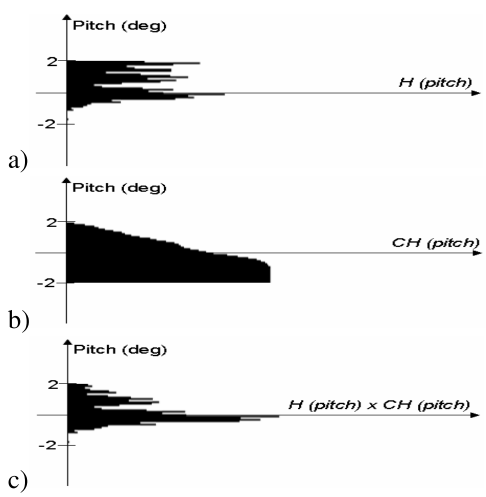

Fig. 8. Combining the cues for pitch: a) polar histogram, b) cumulative histogram, c) combination

The pitch candidate with the highest likelihood, corresponding to the highest value of the histogram product, is chosen as the pitch estimate. Figure 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 lead to better results.

The value vectors **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 3D point set, in order 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 3D space, and for the next stages, the particles have to be compared to image space measurement data. In order to achieve the comparison, from each particle value of the form $\overline{\mathbf{x}}_t^i = (W, C_H, C_V, X_C, \alpha, \gamma, \psi)^T$ a measurement space vector is generated, $\overline{\mathbf{y}}_t^i = (v_1, ...v_P, u_{L,1}, ...u_{L,P}, u_{R,1}, ..., u_{R,P})$. The values *v* are coordinates of image lines and the values *u* are coordinates of image columns. The *v* values are common to the left and right delimiter. *P* is the number of points chosen to describe each lane delimiter in the image space.

In order to derive $\overline{\mathbf{y}}_t^i$ from $\overline{\mathbf{x}}_t^i$, the following steps have to be taken:

a) Generate *P* points in the 3D space, for each lane delimiter. The points will be equally spaced on the distance axis *Z*, and their *X* and *Y* coordinates (lateral and height) will be given by the horizontal profile and vertical profile of the lane. The nearest points will start at the distance Z_B, the closest distance that allows the road to be visible to our system. The points will span a detection distance *D*. The 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, usually on straight or low curvature roads, but a shorter one is needed at slow speeds to handle narrower curvatures. The distance *D* covers a second of vehicle movement at the current speed, but no shorter than 5 m and no longer than 60 m.

b) Project the 3D points in the image space, using the camera parameters. For each lane delimiter, a chain of unevenly spaced points will be obtained.

c) 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 $\overline{\mathbf{y}}_t^i$.

E. *The visual cues*

After the pitch angle has been detected from the 3D 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 3D 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]. Besides 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 priority, they are inserted in a common “special edge” map, which represents the second lane measurement cue.

In order to allow comparison between the particles and the measurement, each cue map (road edges or special edges) undergoes a Distance Transformation.

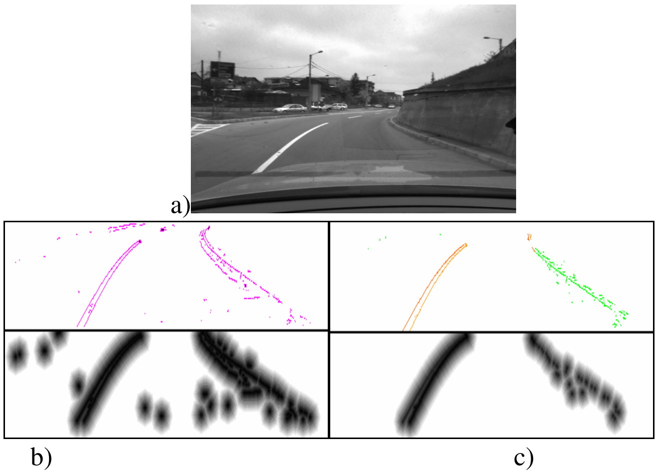

Fig. 9. Visual information: a) the original grayscale image, b) the edges corresponding to 3d points contained in the road surface and the associated distance transform image, c) markings and curbs, and their associated DT image

F. Particle Weighting 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 predicted particles, weights proportional to the measurement likelihood given the state hypothesis.

$$\pi_t^i = p(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}_t^i) \tag{12}$$

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

$$p(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}_t^i) = p(road\_edges \mid \mathbf{x}_t = \mathbf{x}_t^i) \cdot p(mark\_curb \mid \mathbf{x}_t = \mathbf{x}_t^i) \tag{13}$$

In order 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 becomes now very helpful.

Ideally, lane hypothesis boundaries’ projections in the image space have to fit exactly on the edges of the visual cues. Also, the area inside the hypothetic lane projection has to be as free of edges as possible. In order to test these two conditions, two sets of points are used: the positive points, which are points belonging to the lane delimiters’ projection in the image space, and negative points, which are points near the borders, residing inside the projected lane area (fig. 10).

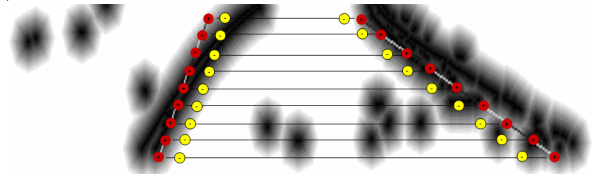

Fig. 10. Positive and negative points. Positives are lane boundary points, and negatives are points inside the lane area.

The positive points will generate the positive distance; this is obtained by averaging the distance transform pixel values at these points’ coordinates. The distance corresponding to the negative points is the complement of the distance transform image at these points’ coordinates. The two distances are combined by weighted averaging (equation 14). The value of the weight parameters α and β has been set to 2 and 1, respectively, through trial and error experiments.

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

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

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

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 Kalman filter lane tracking solution, the particle filtering system does not need initialization or measurement validation before track update. The particles will evolve freely, 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 the random particles are inserted at the head of the particle list (without altering the probability density), a quality factor is defined as:

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

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 in 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 the particles having a higher than average weight are considered for estimation.

VII. Tests and Results

1. Comparison with a Kalman filter solution

The stereovision-based particle filtering lane detection system has been designed mainly to improve the handling of difficult situations, when the Kalman filter 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, “discontinuous road” (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 is image saturation).

A Kalman filter solution has problems with road discontinuities due to the following characteristics:

- There is only one possible lane configuration that is tracked at one moment in time
- The current state is used to predict search areas for the next detection, a feature which drops all measurements that indicate a road discontinuity
- The system requires time to drop a track and time to initialize a new track
- 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 the following behavior:

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 2-3 frames.

2. In lane changing maneuvers there are two aspects of our algorithm that make the transition as smooth as possible: the ability to track multiple hypotheses and 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 towards 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 better chances of being the right one.

4. Sharp changes of curvature are either handled 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 conditions, by fast recovery once the discontinuity has been passed, in either case the situation of false estimation being avoided.

5. Due to the fact that there is no track reset in the particle filter system, sensor failures are treated uniformly 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 Kalman filter solution is provided in the movie file *pf-kf.avi*. In the left half of the frame the particle filter solution is displayed, and in the right half one can see the results of the Kalman filter.

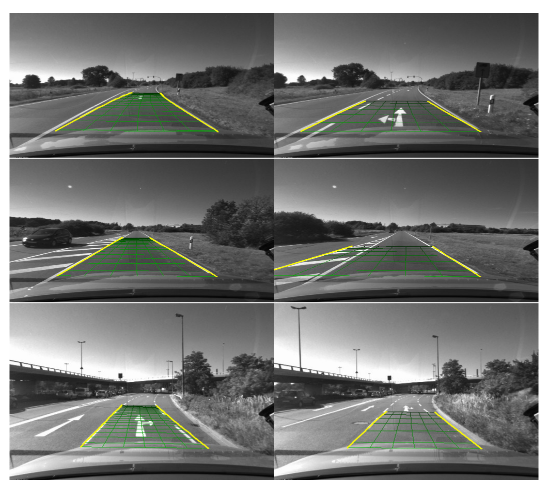

Fig. 11. Samples of side by side comparison. Left – particle filter solution, Right – Kalman filter solution.

2. Pitch angle evaluation

In order to evaluate the results of the pitch detection method that we proposed in the paper, a sequence of images of high pitch variation was selected. The road geometry forces the vehicle to change the pitch in the range of -3 to +4 degrees. 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 3D window (1 meter wide, 7 meters long). Due to the fact that we ensured the sequence to be obstacle-free in the selected window, the 3D 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 3D points used in the algorithm are not restricted to a narrow window, and on the sides of the road there are plenty of obstacle features. The graph in figure 12 shows the two pitch values (in degrees) against the frame number. The pitch detection system results are shown with dotted line, and the comparison “ground truth” is shown with continuous line. The 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, follows closely 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 (3D 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 meters).

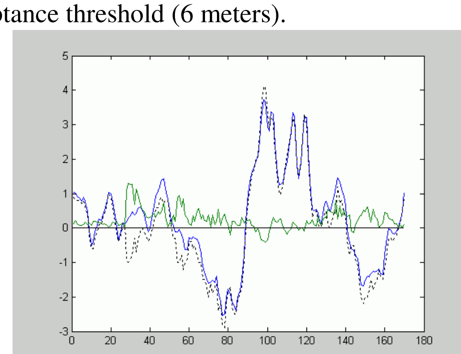

Fig. 12. Pitch angle comparison.

3. Highway performance evaluation

The performance of the lane detection system on highways is evaluated using a sequence of 2644 frames, acquired at about 10 frames per second (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 minutes of driving. The sequence contains lane changes, highway exist and reentry, and impaired visibility due to rain and windshield wipers. Figures 13 to 16 show the evolution of some of the lane parameters (width, curvature, lateral offset and yaw angle – 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, high 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-eye view, can be seen in the file *highway.avi*. The purple points in the bird-eye view are the 3D points provided by stereovision that are marked as road points by filtering against the detected pitch and a Canny edge detector.

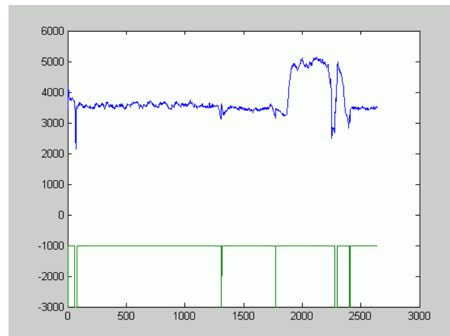

Fig. 13. Highway behavior: width (mm) versus frame number.

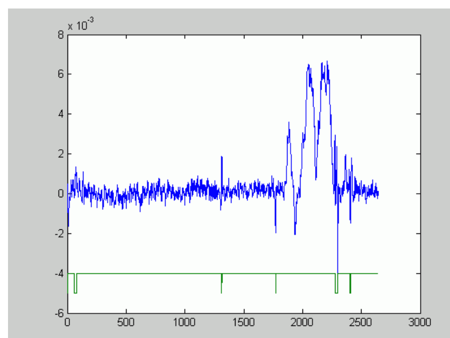

Fig. 14. Highway behavior: horizontal curvature (m^{-1}) versus frame number.

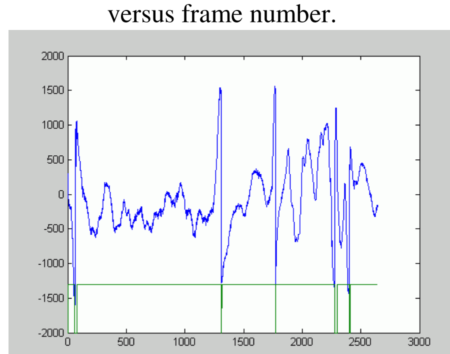

Fig. 15. Highway behavior: lateral offset (distance of the vehicle from lane center, in mm) versus frame number.

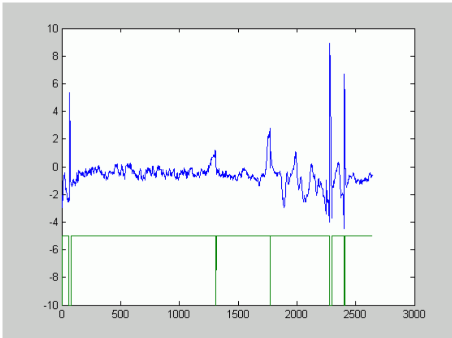

Fig. 16. Highway behavior: yaw angle (in degrees) versus frame number.

4. 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 frames per second (variable), which means about 3 minutes of driving. The sequence contains lane changes, lane forking and joining, passing through a tunnel and passing through intersections. Figures 17 to 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 why the detection rate is so low. The sequence and the detection results, in perspective projection and bird-eye view, can be seen in the file *urban.avi*.

A crowded urban sequence, with poor quality lane delimiters, plenty of obstacles, driving on the lane border and so on is available for qualitative analysis only in the file *crowded.avi*.

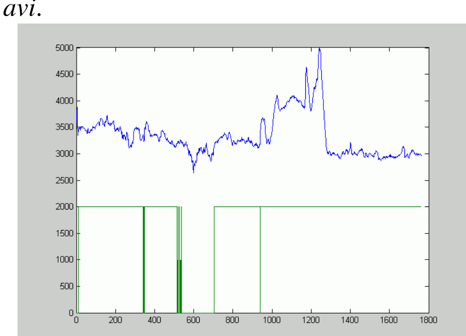

Fig. 17. Urban behavior: width (mm) versus frame number

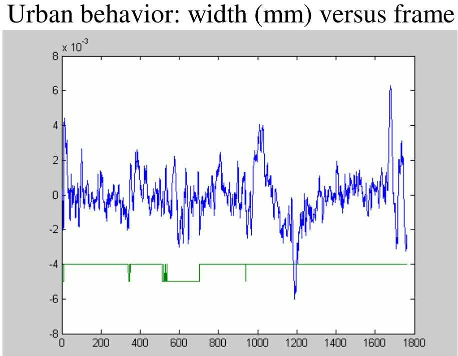

Fig. 18. Urban behavior: horizontal curvature (m^{-1}) versus frame number

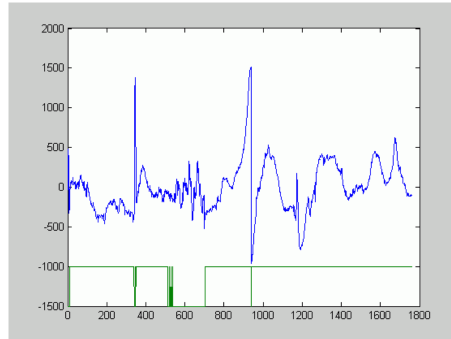

Fig. 19. Urban behavior: lateral offset (distance of the vehicle from lane center, in mm) versus frame number

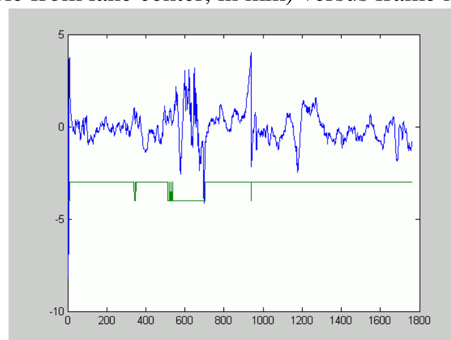

Fig. 20. Urban behavior: yaw angle (in degrees) versus frame number

5. Experimental testing – linear variable width model

Due to the high adaptability of the particle filter framework, new lane models can be easily tried. Inserting a width variation parameter (which describes the increase/decrease of width with the longitudinal distance, required only to change the function that computes the lateral coordinates (X) of the lane delimiters with the distance (Z). The resulted 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 comparing the estimated width variation with a differentiation of the estimated lane width against the traveled space is shown in figure J. The modeled width variation is the smoother signal.

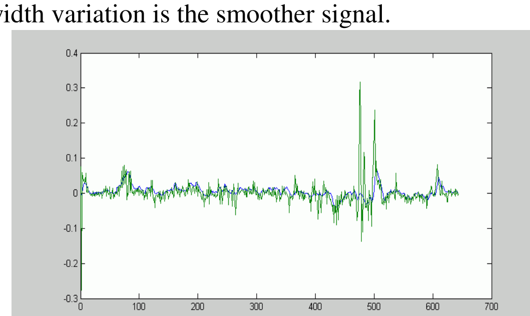

Fig. 21. Comparison between width variation estimated by lane detection, and the differentiation of estimated width against traveled space

6. Time performance

The time performance has been evaluated on an Intel Core2 Duo CPU, at 2 GHz, using a single thread. The lane detection time has a fixed part, independent on the number of particles, amounting to 9.6 ms, and a time per processed particle of 0.0075 ms. Our 200 particle solution takes a total of 11 ms to complete.

Note: The movie files showing the test results can be also 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, in order to robustly detect the lanes in difficult conditions. The system does not use detection in the classical sense, there is no track initialization or track loss, and thus the processing time is kept constant, regardless of 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 (like 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, B.D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no.2, pp. 199-213, 1992

[2] R. Aufrere, R. Chapuis, F. Chausse, "A model-driven approach for real-time road recognition", *Machine Vision and Applications*, Springer-Verlag 2001

[3] M. Isard, A. Blake, "CONDENSATION – conditional density propagation for visual tracking", *International Journal of Computer Vision*, vol. 29, nr. 1, pp. 5-28, 1998

[4] K. Macek, B. Williams, S. Kolski, R. Siegwart, "A Lane Detection Vision Module for Driver Assistance", in proc. of *IEEE/APS Conference on Mechatronics and Robotics*, 2004

[5] U. Franke, H. Loose, C. Knoeppel, „Lane Recognition on Country Roads", in proc. of *IEEE Intelligent Vehicles Symposium*, 2007, Istanbul, Turkey

[6] R. Labayrade, J. Douret, D. Aubert, "A Multi-Model Lane Detector that Handles Road Singularities", in proc. of *IEEE Intelligent Transportation Systems Conference*, 2006, Toronto, Canada

[7] S. Nedevschi, R..Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, "3D Lane Detection System Based on Stereovision", in proc. of *IEEE Intelligent Transportation Systems Conference*, 2004, Washington, USA

[8] R. Danescu, S. Nedevschi, M.M. Meinecke, T.B. To, "Lane Geometry Estimation in Urban Environments Using a Stereovision System", in proc. of *IEEE Intelligent Transportation Systems Conference*, 2007, Seattle, USA

[9] F. Oniga, S. Nedevschi, M.M. Meinecke, T.B. To, "Road Surface and Obstacle Detection Based on Elevation Maps from Dense Stereo", in proc. of *IEEE Intelligent Transportation Systems Conference*, 2007, Seattle, USA

[10] B. Southall, C.J. Taylor, "Stochastic road shape estimation", in proc. of *IEEE International Conference on Computer Vision*, 2001, Vancouver, Canada

[11] P. Smuda, R. Schweiger, H. Neumann, W. Ritter, „Multiple Cue Data Fusion with Particle Filters for Road Course Detection in Vision Systems", in proc. of *IEEE Intelligent Vehicles Symposium*, 2006, Tokyo, Japan

[12] Z. Kim,"Robust Lane Detection and Tracking in Challenging Scenarios," *IEEE Trans. on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16-26, 2008