A Stereovision-Based Probabilistic Lane Tracker for Difficult Road Scenarios

Radu Danescu, Sergiu Nedevschi, Marc-Michael Meinecke, Thanh-Binh To

Abstract— **This paper presents a lane estimation technique based on the particle filter framework, which fuses several image-based cues (edges, lane markings and curbs), and 3D cues extracted from stereovision. A partition sampling-like approach is used to decouple pitch estimation from the rest of the parameter set, allowing the use of a significantly lower number of particles, and initialization samples are used for faster handling of discontinuous roads. We also introduce a measure for detection quality, for result validation. The resulted solution has proven to be a reliable and fast lane detector for difficult scenarios.**

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 matching techniques, which use constraints to counteract possible noisy features. To further increase the stability and speed, probabilistic reasoning in the form of tracking was introduced, usually 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.

Manuscript received January 9, 2008. This work was supported by Volkswagen AG.

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

M.M. Meinecke and T.B. To are with Volkswagen AG, Driving Assistance Electronics department, e-mail: marc-michael.meinecke@volkswagen.de, thanh-binh.to@volkswagen.de.

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.

II. 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.

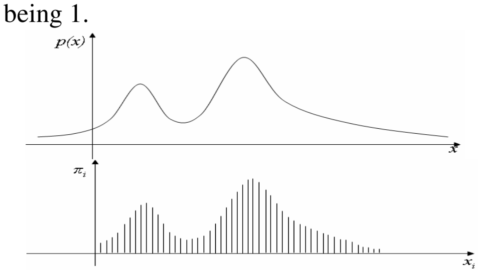

Fig. 1. Analogy between a probability density function and a set of weighted samples

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.

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.

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. Several cues can be combined in this step by multiplication, using the cue conditional independence assumption.

III. 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.

IV. 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 $p(\mathbf{x}) \approx \{\mathbf{x}_t^i, \pi_t^i, i = 1 \ldots N\}$. The particle value **x** is a lane state hypothesis, in the form of a lane description vector containing the lane width, horizontal curvature, vertical curvature, lateral offset (displacement of the ego vehicle from the center of the lane), and the angles of pitch, roll and yaw.

B. Prediction

Before prediction can be applied, the past state described by the particle set has to be resampled into particles of equal weight. Each one of these particles is subjected to the prediction process, which alters the particle value by applying the following equation:

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

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 ct, the curvature of the vehicle's trajectory, derived from the yaw rate. Matrices $\mathbf{A_t}$ and $\mathbf{B_t}$ depend on the space 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 Qt.

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.

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. 2. 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

$$\alpha_p = \tan^{-1}\frac{height(p)}{distance(p)} \tag{2}$$

  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

$$H(index_p) = H(index_p) + \frac{distance(p)^2}{K} \tag{3}$$

  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.

$$p(\alpha \mid y_1, y_2, ..., y_t) = p(\alpha \mid y_t) \tag{4}$$

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.

$$p(\alpha \mid y_t) \propto p(y_t \mid \alpha) \tag{5}$$

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*.

$$p(y_H \mid \alpha = \alpha_{index}) \propto H(index) \tag{6}$$

$$p(y_{CH} \mid \alpha = \alpha_{index}) \propto CH(index) \tag{7}$$

$$p(\alpha = \alpha_{index} \mid y_t) \propto H(index)CH(index) \tag{8}$$

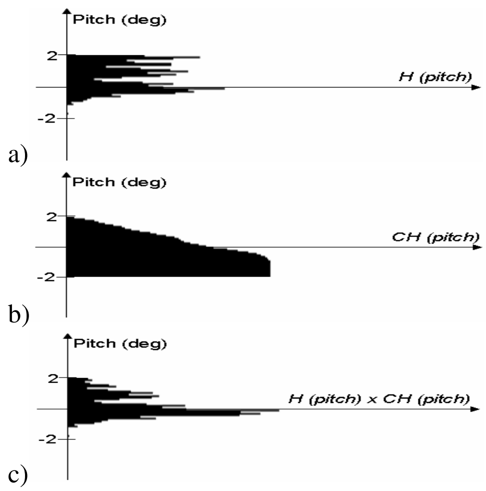

Fig. 3. Combining the cues for pitch: a) polar histogram, b) cumulative histogram, c) combination

The pitch candidate with the highest likelihood, corresponding to the highest value of the histogram product, is chosen as the pitch estimate. Figure 3 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. Image space measurement cues

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, $\overline{\mathbf{y}}_t^i = (v_1, \ldots v_P, u_{L,1}, \ldots u_{L,P}, u_{R,1}, \ldots, u_{R,P})$. 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.

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.

E. 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.

$$\pi_t^i = p(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}_t^i) \tag{9}$$

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

$$\begin{aligned} p(\mathbf{y}_t \mid \mathbf{x}_t = \mathbf{x}_t^i) &= p(road\_edges \mid \mathbf{x}_t = \mathbf{x}_t^i) \cdot \\ & p(mark\_curb \mid \mathbf{x}_t = \mathbf{x}_t^i) \end{aligned} \tag{10}$$

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. 4).

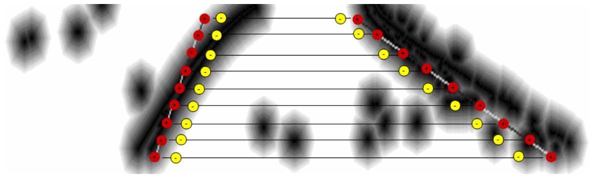

Fig. 4. Positive and negative points. Positives are lane boundary points, and negatives are points inside the lane area.

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, the positive distance being of twice the importance of the negative one.

$$D^M = \frac{2D^M(+) + D^M(-)}{3} \tag{11}$$

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.

$$p(M \mid \mathbf{x}_t = \mathbf{x}_t^i) = \frac{1}{\sigma_M \sqrt{2\pi}} e^{-\frac{D_M{}^2}{2\sigma_M{}^2}} \tag{12}$$

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.

F. 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:

$$q = \frac{R \sum_{i=R+1}^{N} \pi_t^i}{(N-R) \sum_{i=1}^{R} \pi_t^i} \tag{13}$$

If q is higher than 10, 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).

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.

V. Tests and Results

The stereovision-based particle filtering lane detection system has been designed to improve the handling of difficult scenarios, 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
- 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.

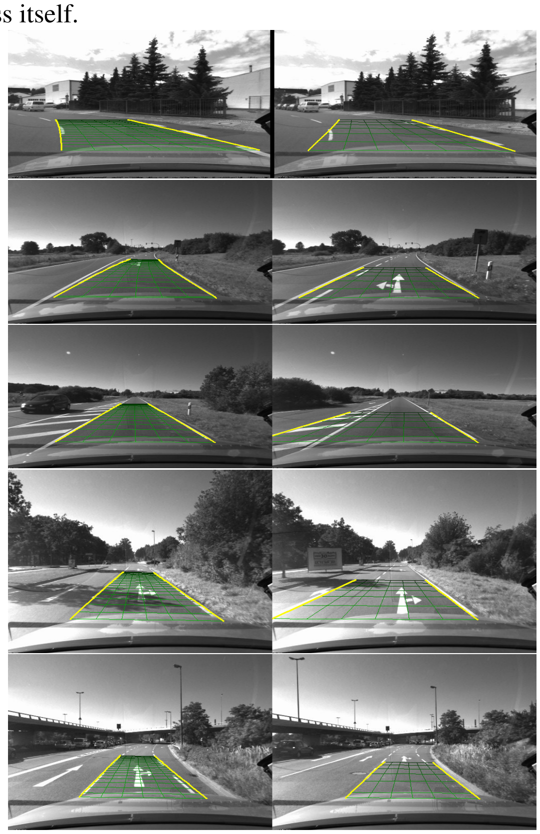

Fig. 5. Side by side comparison. Left – particle filter solution, Right – Kalman filter solution.

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.

VI. C ONCLUSION 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. Also, due to the fact that the 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