New Results in Stereovision Based Lane Tracking

Radu Danescu, Sergiu Nedevschi, *Member, IEEE*

***Abstract*—Robust and versatile lane tracking solutions can be developed using particle filter based tracking, stereo data and grayscale image processing. This paper presents multiple improvements for our previously published solutions. The lane model is extended by parameters such as width variation, vertical offset, and widths of lane markings. The increased flexibility in lane modeling is matched by a better use of measurement data, and the particle weighting process will benefit from two additional cues. The extensions in lane model and in particle weighting provide a more powerful description of the lane, but will also increase the stability and accuracy for the estimation of all lane parameters.**

I. Introduction

THE lane tracking problem can be seen as a set of three sub-problems. The first sub-problem to be solved consists of finding an appropriate lane model, which includes the static representation and the dynamics of the evolution in time; the second sub-problem is to extract the lane delimiting features by processing sensorial data, and the third problem is to find a good estimation/tracking algorithm, that will adjust the parameters of the lane model to match the real world reflected in the sensor data.

Besides the classical clothoid, researchers have defined and used more complex lane models, with the aim of increasing the power of lane description. These models, such as the dual clothoid [1], multiple circular lane segments [2] or independent 3D B-Spline models for each lane delimiter [3] are mostly aimed at representing long stretches of road, or situations where the road changes its geometrical characteristics rapidly. For long segments of road, changes in the lane model may work well with the use of long range sensors, such as radar, that can refine the base lane geometry previously estimated by the use of classical tracking techniques [4].

This paper presents a refinement of the lane model and of the lane estimation techniques, but instead of looking far ahead, we'll try to improve the results in the close range. The work presented in this paper is based on our previous results in particle filter-based lane tracking using stereo information, results which are presented in [8]. The vertical road geometry is extended by the use of a height offset, which allows the system to better cope with the car's oscillations. The pitch angle is inserted in the tracking process, and not evaluated at each frame independently. The variation model is assumed to be quasi-stationary, but a further extension to an oscillatory model, such as the one presented in [5], is possible. The horizontal lane model is extended with the parameters for the width of the lane markings, and the matching process is augmented with additional cues in order to be able to use these parameters. The width of the lane markings has been used before in the lane detection/tracking field, but usually as a means of selecting the right lane delimiters, as in [6], or for selecting the primary lane delimiting curves [7]. In this paper, the width of the delimiters is assumed to be unknown and variable, and the lane tracking process will estimate these values along with the rest of the parameter set.

Manuscript received January 21, 2011. This work was supported in part by CNCSIS –UEFISCSU, project number PNII – IDEI 1522/2008, and by the POSDRU program, financing contract POSDRU/89/1.5/S/62557.

Radu Danescu and Sergiu Nedevschi are with the Technical University of Cluj-Napoca (e-mail: radu.danescu@cs.utcluj.ro). Department address: Computer Science Department, Str. Memorandumului nr. 28, Cluj-Napoca, Romania. Phone: +40 264 401457.

II. Modeling the Lane

The lane model is an extension of the model presented in [8], which has the following parameters: W_0 – *Lane width,* C_H – *Horizontal curvature,* C_V – *Vertical curvature,* D – *Detection distance,* X_{CW} – *Lateral offset,* α – *Pitch angle,* γ – *Roll angle,* ψ – *Yaw angle.*

In this paper, we'll extend the lane model by the adding several additional parameters to the set:

W_1 – width variation. The width becomes linearly variable with the distance Z from origin, allowing the lane to widen or narrow. By width we'll understand the distance between the inner edges of the lane delimiters (the width of the asphalted surface between the markings).

$$W(Z) = W_0 + W_1 Z \tag{1}$$

Y_0 – the vertical offset. So far, we have assumed that the vertical coordinate of the road surface was completely described by the pitch angle, the vertical curvature, and the roll angle. This was equivalent to the assumption that for X=0 and Z=0, the Y coordinate is always zero. We have found that sometimes this assumption is not true, and a free term (offset) may appear. This free term, while not a high value in itself, can affect the precision of the other components of the vertical profile, leading to false results. By modeling and tracking this parameter, we will improve the quality of the match between the model and the reality.

W_L and W_R – the width of the left and right lane markings. If the lane is delimited by painted markings, the width of these markings may be of interest for the driving assistance applications. If the lane is delimited by something else than markings, the value of these parameters will be zero. An

additional benefit of using these parameters is that the overall quality of the lane geometry estimation will be improved. This is caused by the capability of the system to better lock on the lane marking, following both its edges, and not only the inner edge (the edge closer to the center of the lane). We'll see how this is achieved in the following sections.

The new lane state vector has 11 parameters: W_0 – lane width at Z=0, C_H – horizontal curvature, W_1 – lane width variation with the distance, C_V – vertical curvature, Y_0 – vertical offset, X_{CW} – lateral offset, α – pitch angle, γ – roll angle, ψ – yaw angle, W_L – left marking width, W_R – right marking width.

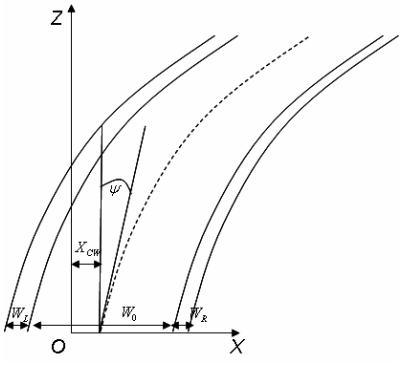

Fig. 1. Horizontal profile parameters.

The horizontal profile equations now describe the X coordinates of the inner and outer lane delimiters.

$$\begin{cases} X_{LIn}(Z) = X_C(Z) - W(Z)/2 \\ X_{RIn}(Z) = X_C(Z) + W(Z)/2 \end{cases} \quad (2)$$

The equations for the outer delimiters use the marking width parameters.

$$\begin{cases} X_{LOut}(Z) = X_C(Z) - W(Z)/2 - W_L \\ X_{ROut}(Z) = X_C(Z) + W(Z)/2 + W_R \end{cases} \quad (3)$$

For the vertical profile, the equation of the lane height when X=0 is now changed to accommodate the height offset.

$$Y_C(Z) = Y_0 + \alpha Z + C_V Z^2/2 \quad (4)$$

III. Particle Filter Based Lane Tracking

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. 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, in order to cover the possible discontinuities that may arise (initialization particles).

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.

The *preliminary pitch detection* is done independently of the particle system, using a probabilistic approach. This pitch value is used to select the road features, which are then used to weight the particles in the measurement step.

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

A more detailed description of the particle filter solution applied to lane detection is presented in [8].

IV. Preliminary Pitch Detection and Selection of Road Features

The pitch angle's behavior in time is less predictable than the behavior of the rest of the lane parameters. This angle is influenced by sudden acceleration and braking maneuvers, or by imperfections in the road surface. Also, the pitch angle is the key parameter in selecting the road features based on their stereovision-provided coordinates. Relying solely on a prediction/update mechanism may lead us to selecting false road data, cascading the tracking errors.

Initially, due to the unpredictability of the pitch angle evolution in time, we have determined this parameter in each frame independently, using the algorithm presented in [8]. This technique does not rely on past information, and does not take into consideration the existence of the vertical offset Y_0 in the lane vertical profile.

Detecting the pitch angle directly has the advantage of coping with the fast changes in this angle, but it is not the perfect solution for normal traffic conditions, where one can benefit from some degree of filtering. For this reason, the solution presented in this paper uses the pitch value extracted here only as a preliminary discriminator of road points, while the actual value of the pitch angle and vertical offset are tracked using the particle filter mechanism, along with the rest of the parameters.

V. Mapping the Particles into the Measurement Space

The measurement process will fuse several different cues, obtained using different ways of comparing the lane hypothesis of the particle with the reality described by sensorial data. Some of these comparisons are done in the image space, and some are done in a discrete 3D space which approximates the 3D point distribution in the YOZ plane (the plane of the vertical profile). For this reason, each particle will be mapped into the image space, and into the discrete lateral 3D space.

A. Mapping the particles into the image space

Mapping into the image space is done by projecting the inner and outer border points of each lane hypothesis, using the camera parameters. The process is described in detail in [8]. The only change this paper brings is the addition of the

outer lane border to the projection.

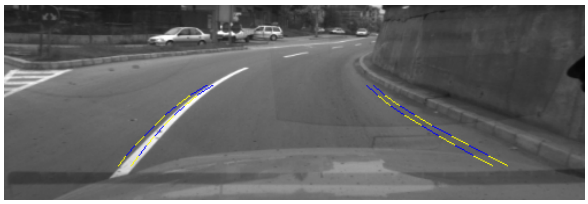

Fig. 2. Mapping a particle into the image space.

B. Mapping the particle into the discrete 3D space

The discrete 3D space is represented as a matrix of h by w cells, each cell corresponding to a rectangle in the YOZ plane (the lateral projection of the 3D space) of dZ = 10 cm in length and dY = 1 cm in height. The measurement process in this space is used to help the particles converge faster to the pitch and vertical offset values, as the convergence rate would be quite slow if we would rely only on 2D data for measurement.

The particle representation in the discrete 3D space is the vector $\overline{\mathbf{y}}^i_{3D,t} = (r_1,...r_P, c_1,...c_P)$, r_i being the rows and c_i being the columns of P points. The process of obtaining this vector from $\overline{\mathbf{x}}^i_t$ is the following:

a) Generate P distance values Z_k, k=1 to P, as:

$$Z_k = Z_B + (D_{3D} - Z_B)k / P \tag{5}$$

Z_B is the minimum distance where the road becomes visible (the blind spot), and D_{3D} is the range we have chosen for this mapping. D_{3D} is not the same as the distance we have chosen for the 2D space mapping, but shorter, and fixed. Currently, this distance is 15 meters.

b) Get the Y coordinates of these points, using the vertical profile equation of the lane:

$$Y_K = Y_0 + \alpha Z_k + C_V Z_k^{\,2} / 2 \tag{6}$$

c) Get the discrete values r_k and c_k, corresponding to Y_k and Z_k.

$$c_k = Z_k / dZ \tag{7}$$

$$r_k = Y_k / dY + h/2 \tag{8}$$

The h/2 offset in equation 8 is required because the Y values can be either positive or negative (pitching forward or backward, etc). No such offset is needed for Z.

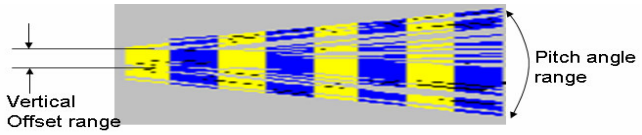

Fig. 3. Particles projected into the discrete 3D space.

VI. The Measurement Cues

The measurement process compares the particle state, projected into the measurement space, with several visual cues. For each comparison with a cue, a weight is computed. Finally, all the weights are combined by multiplication. First, we will analyze the cues themselves, and then we'll see how the comparison is made.

A. Measurement cues in the image space

Three types of cues are specific to the image space: 1-road edges, 2-markings and curbs, and 3-the horizontal gradient computed by a variable width kernel.

a) Road edges – these are the most basic pieces of information about the lane. They are the projection of the 3D points, selected by the pitch angle as belonging to the road. The road edge image is subjected to a distance transformation, which creates a distance image DT^E. This distance image will be processed in the particle weighting algorithm.

b) Lane markings and curbs – most lanes are delimited by markings or by curbs. The lane markings are extracted using the algorithm described in [9], and the curbs are extracted by elevation map processing using the method described in [10]. They have the same importance in our algorithm, and they will be processed together. A special edge image is created, combining the lane marking edges and the curb edges, and this edge image is subjected to the distance transform. A distance image, DT^S (Distance Transform – Special Delimiters) is obtained, and will be used in the particle weighting process.

c) The horizontal gradient computed using the variable width kernel – this gradient is an intermediate result in the lane marking extraction process, and the method for its computation is shown in [9]. The idea for the variable width kernel differentiation is to smoothen out the near road texture (which can produce noisy edges), while preserving the detail of the far road delimiters. This gradient image (fig. 4) is used as it is in the particle weighting process. This cue is the most valuable one for the fitting of both inner and outer lane borders to the image data.

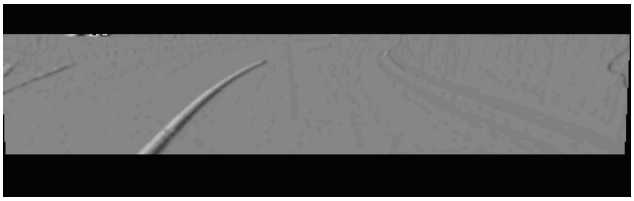

Fig. 4. The variable kernel width horizontal gradient image.

B. Measurement cues in the discrete 3D space

The road points in the continuous 3D space will be transformed into features in a 200x300 pixels image which corresponds to discrete cells of 1cm x 10 cm size. We considered that the density of 3D points in each cell is a reasonable initial value for the discrete image value. Instead of using a point per cell count as an approximation of point density, approximation that would lead to discontinuities, we have used a triangular filter [12], which will produce much more continuous results.

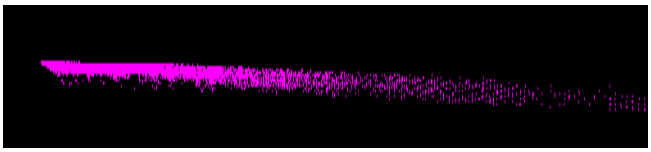

Fig. 5. Lateral view of the road 3D points.

The density image is initialized with null values. Then, each 3D point will increment its corresponding cells with a maximum value, and the neighboring cells with values that decrease linearly with the distance from the center. The size of the filter is 11x11, and the linear decrement is 1. This means that the center cell is incremented by a value of 6, and the outer cells by a value of 1. The result of density computation for the point set displayed in figure 5 is shown in the figure below.

Fig. 6. Density in the discrete 3D space (density image).

The next step is taken under the assumption that the road surface consists of column local maxima, that is, points that are of higher density than its neighbors above or below its position. Therefore, a non-maxima suppression routine for columns is applied, and all points having a density value lower than its top or bottom neighbors are eliminated. This step is followed by a thresholding with a low and fixed threshold, to remove the very low density cells. What results is the density edge image, shown in figure 7.

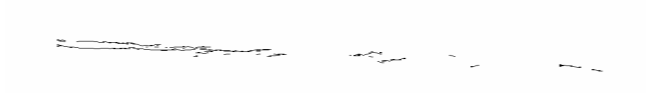

Fig. 7. Edge obtained by non-maximum suppression and thresholding of the density image.

A distance transform routine is applied to the density edge image, and the result is the final cue that will be used in the particle weighting process. We'll denote this distance image as DT^D.

VII. Using the Cues for Particle Weighting

A. Computing the road edge and the marking/curb likelihood

The road edge likelihood and the marking/curb likelihood are computed in the same way, as the same algorithm is applied to the distance transform images DT^E and DT^S. For the computation of these likelihoods, only the inner edges of the lane delimiters (or, more exactly, their projections in the image space) are used.

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 *inner* lane delimiters' projection in the image space, and negative points, which are points near the borders, residing inside the projected lane area, the asphalt area between delimiters.

It is important that we do not confuse the *negative/positive pairs of point chains* with the *inner and outer point chains* described previously. The positive points are the inner point chains, and the negative points are inside the area described by the inner point chains. The outer point chains are not used in computing the edge or marking likelihoods.

The positive points will generate the positive distance; this is obtained by averaging the distance transform pixel values at these points' coordinates.

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

B. Computing the horizontal gradient likelihood

The horizontal gradient image contains both positive values, which stand for dark to light transitions, and negative values, which stand for light to dark transitions. A horizontal line which intersects a white lane marking in the grayscale image will intersect first a positive then a negative area of the gradient image.

The horizontal gradient image computed with a variable width kernel has continuous transitions of the gradient values, both positives and negatives, towards the local maxima which correspond to the marking edges. For this reason, this gradient image can be used directly, without the need for additional smoothing through Distance Transform, as we have done with the previous cues.

We shall exploit this cue using the full representation of the lane particle in the image space, meaning that we shall use both the inner and the outer delimiter points. The matching function will aim to fit the sides of a lane delimiter to the corresponding intensity transition – the left side of the delimiter to a dark-light transition, and the right side of the delimiter to a light-dark transition. For our notations, this translates in the fact that the outer points of the left delimiter and the inner points of the right delimiter must fall on positive values of the gradient, and the inner points of the left side along with the outer points of the right side must fall on negative gradient values.

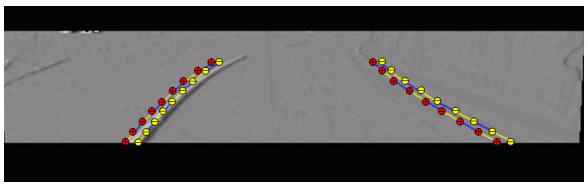

Fig. 8. Comparing the lane particle projection with the horizontal gradient image. Positive points (red) should match on positive gradient values, and negative points (yellow) should match on negative gradient values.

We shall denote the points that must fall on positive values as + points, and the points that must fall on negative values as – points. Figure 8 shows both sets of points.

For the positive points, *2P* of them, we shall compute the distance D^G. Taking into account that the pixel value of the gradient image is centered in 128, this means that the maximum positive gradient value is 255 and the minimum negative gradient is zero. That is, the positive points should fit (ideally) on 255 and the negative points on zero. Denoting by *HG(u,v)* the horizontal gradient image at coordinates *u* and *v*, we have the average distance for positive and for negative points:

$$D^G(+) = \sum_{+} (255 - HG(u^+, v^+)) \Big/ 2P \tag{9}$$

$$D^G(-) = \sum_{-} HG(u^-, v^-) \Big/ 2P \tag{10}$$

Due to the fact that both the positive and the negative points have now the same importance, the final distance is computed as a simple average.

$$D^G = (D^G(+) + D^G(-))/2 \tag{11}$$

This distance is transformed into a weight by the Gaussian function.

$$p(gradient \mid \mathbf{x}_t = \mathbf{x}_t^i) = \frac{1}{\sigma_G \sqrt{2\pi}} \exp\left(-\frac{(D^G)^2}{2\sigma_G{}^2} \right) \tag{12}$$

C. Computing the 3D point density likelihood

The comparison between the discrete vertical profile, which was computed as chain of points in the discrete coordinates of the discrete space image, and the measurement data, which is available in the form of a distance image DT^D, is a straightforward process. There are no negative points this time, and, ideally, all *P* points of the projection should match on zero values of the distance image. The comparison distance is a simple average:

$$D^D = \sum DT^D(u,v) \Big/ P \tag{13}$$

This distance translates easily into likelihood.

$$p(density \mid \mathbf{x}_t = \mathbf{x}_t^i) = \frac{1}{\sigma_D \sqrt{2\pi}} \exp\left(-\frac{(D^D)^2}{2\sigma_G{}^2} \right) \tag{14}$$

D. Combining the likelihoods, generating the final weight

The likelihoods will be combined by multiplication, and the result is assigned as weight to each particle. A validation is performed for each particle, and if the lane hypothesis represented by the particle is degenerate (the lane is too narrow, too wide, too far from the vehicle's position), the weight is set to zero. This validation will prevent a degenerate lane particle to spawn new hypotheses in the next frame.

VIII. Lane Estimation

Due to the fact that the particle filter may track multiple hypotheses at the same time, at least for a short period of time, it is not always recommended to consider all the particles in the set for the lane state estimation. For this reason, a clustering algorithm is applied after the measurement step is completed.

The clustering algorithm is the MBSAS, Modified Basic Sequential Algorithm Scheme [11]. The particle set is partitioned into at most *S* clusters based on a distance between particles. A satisfactory metric, which is also very simple and easy to compute, is the absolute value of the difference between the lateral positions of each particle, X_{CW}.

The cluster C_{max} that has the highest number of particles will generate the lane estimation. Formally, we can write:

$$\varepsilon(\mathbf{x}_t) = \sum_{i=R}^{N} \mathbf{x}_t^i \pi_t^i \delta_t^i \Big/ \sum_{i=R}^{N} \pi_t^i \delta_t^i \tag{15}$$

Not all non-initialization particles (resampled particles) participate in the lane estimation process, a fact indicated by the binary condition δ_t^i. This condition states that only the chosen particles are considered for estimation. The particles chosen for estimation must belong to the most populous cluster, and they must have a higher than average weight.

IX. Tests and Results

The extensions to the lane model and to the measurement process brought increase stability and accuracy to the overall lane parameter estimation process. The addition of lane markings width to the parameter set allowed us to extract the most out of the new measurement source, the horizontal gradient image, because of the direct dark-light-dark match. Not only are we able to extract more information about the lane, such as the lane marking width, but the overall lane matching quality is improved. In figure 9 we have a controlled test situation, where we know the lane marking width (100 mm) and the lane width (3000 mm).

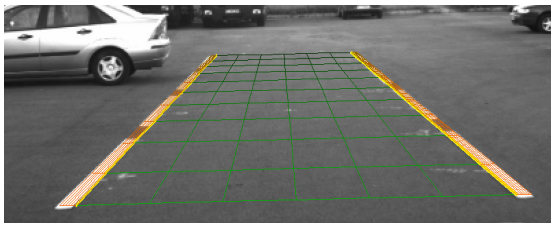

Fig. 9. Controlled test scene, with known lane and lane marking width.

Figure 10 shows a comparison between the quality of lane width estimation from the previous lane tracking approach (no lane marking width model, no horizontal gradient image as measurement) and the new approach. The previous approach has a mean absolute error of 65.69 mm (2.1 %), while the new approach has a mean absolute error of 42.33 mm (1.4%).

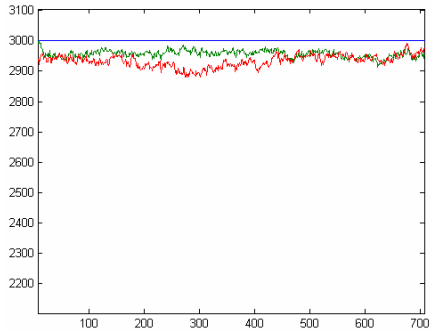

Fig. 10. Lane width estimation, mm versus frame number. Blue – ground truth, Red – estimation without the use of marking width model, Green – estimation using the lane marking width model.

Comparing the estimation of lane marking widths against the ground truth, we have obtained a mean absolute error of 13.2 mm (13.2 %) for the left marking and 11.82 mm (11.82 %) for the right marking. The errors are quite large, but the average image space width of a lane marking is 7 pixels, which means that one pixel of error is 14.2 % of the total amount.

The width variation parameter extends the applicability of the lane tracking algorithm, as the lanes are not always of constant width. A situation when the width is constantly changing is at entrances and exits from the highway. Therefore, we have tested our system on such a situation, to see how well the width variation is detected.

As we do not have a ground truth (this is not a controlled environment), we compare the estimated width variation with the variation of the detected width using a constant width lane model. Consecutive width values are differentiated against the traveled space between frames, to provide us with a reference variation.

The results are shown in the video file http://users.utcluj.ro/~rdanescu/lane/varwidth.avi . A graph comparing the estimated width variation with a differentiation of the estimated lane width (using a fixed width model) against the traveled space is shown in figure 11. As expected, the variations match, but the width variation that is tracked as a model state parameter is a much smoother signal.

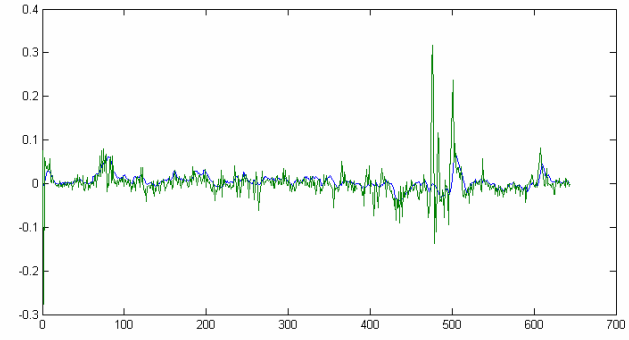

Fig. 11. Comparison between width variation estimated by lane detection (blue), and the differentiation of estimated width against traveled space (green).

X. Conclusions

This paper presents multiple ways in which our stereovision-based lane tracking solution presented in [8] was extended and improved. The improvements consist of extending the lane model with several new parameters, height offset, width variation, and lane marking widths, and the extraction and use of two more cues in the process of lane estimation. In this way, we were not only able to provide additional information about the lane geometry, but also to improve the quality of estimation for the existing parameters.

References

[1] C. Corridori, M. Zanin – "High Curvature Two-Clothoid Road Model Estimation", in proc of *IEEE Intelligent Transportation Systems Conference 2004 (IEEE-ITSC 2004),* pp. 630- 635.

[2] H. Cramer, U. Scheunert, G. Wanielik, "A New Approach for Tracking Lanes by Fusing Image Measurements with Map Data", in proc of *IEEE Intelligent Vehicles Symposium 2004 (IEEE-IV 2004)*, pp. 607- 612.

[3] H. Loose, U. Franke, "B-Spline-Based Road Model for 3D Lane Recognition", in proc of *IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 91-98.

[4] U. Meis, W. Klein, C. Wiedemann, "New Method for Robust Far-Distance Road Course Estimation in Advanced Driver Assistance Systems", in proc of *IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1357-1362.

[5] M. Chech. W. Niem, S. Abraham, C. Stiller, "Dynamic Ego-Pose Estimation for Driver Assistance in Urban Environments", in proc of *IEEE Intelligent Vehicles Symposium 2004 (IEEE-IV 2004),* pp. 43-48.

[6] R. Labayrade, S. S. Ieng, D. Aubert, "A Reliable Road Lane Detector Approach Combining Two Vision-Based Algorithms", in proc of *IEEE Intelligent Transportation Systems Conference 2004 (IEEE-ITSC 2004),* pp. 149-154.

[7] 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, pp. 77-82.

[8] R. Danescu, S. Nedevschi, "Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision", *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, No. 2, June 2009, pp. 272-282.

[9] 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 (ITSC 2007)*, pp. 271 – 276.

[10] 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 Conference on Intelligent Transportation Systems*, 2007, pp. 859 – 865.

[11] S. Theodoridis, K. Koutroumbas, "Pattern Recognition", *Academic Press*, 2006, pp. 527-528.

[12] D. W. Scott, "Multivariate density estimation: theory, practice, and visualization", *John Wiley & sons inc*, 1992.