Proceedings of the IEEE ITSC 2006
2006 IEEE Intelligent Transportation Systems Conference
Toronto, Canada, September 17-20, 2006

Robust Vanishing Point Estimation for Driver Assistance

Thorsten Suttorp
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum, Germany
Email: thorsten.suttorp@neuroinformatik.rub.de

Thomas Bücher
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum, Germany
Email: thomas.buecher@neuroinformatik.rub.de

Abstract— **This paper presents an architecture for real-time vanishing point estimation for driver assistance applications. It consists of a data-driven estimation and a model-based filtering module. The data-driven estimation algorithm is based on line-segments that are assumed to be calculated in an independent preprocessing stage. Model-based filtering is achieved by a Kalman filter that operates on the results of the data-driven processing step. The robustness of the overall estimation is significantly increased by online adaptation of the parameters of both, the data-driven as well as the model-driven processing units. The design of the feedback loop assures that no instable system states occur. The resulting architecture provides robust vanishing point estimation in a wide variety of environmental conditions.**

I. Introduction

In general, the vanishing point is defined by the intersection of the perspective projection of a set of lines that are parallel in world, but not parallel to the image plane, and is specified in image coordinates. In the context of vision-based driver assistance systems the vanishing point is approximated by the linear parts of the lane borders in the vicinity of the vehicle. In this case the vanishing point provides important information about the vehicle's orientation to the ground-plane (pitch angle) and to the lane borders (yaw angle). Assuming the road to be planar it can serve for calculating the transformation from image to world coordinates and vice versa. Therefore, vanishing point estimation plays an important role in driver assistance applications like lane detection (e.g. [1], [2]).

A large number of approaches for vanishing point detection dealing with architectural environments have been proposed (e.g. [3], [4], [5], [6]). Different sets of parallels are used to determine one or several vanishing points, normally given by the parallels of the house walls and the windows. Strategies for vanishing point estimation for vehicle guidance are presented in [7], [8], [9]. In contrast to these approaches we mainly focus on robust real-time vanishing point estimation for driver assistance applications.

In Section II the architecture and the algorithms for vanishing point estimation are described. Section III demonstrates the performance of the presented approach.

II. Architecture for Vanishing Point Estimation

In this section we present an architecture for vanishing point estimation that focuses on robustness and real-time capability. It is based on line segments, which provide a sparse and high level representation of the actual image contours, thereby allowing for fast and efficient algorithms. Further, they can be used not only for vanishing point estimation but for several other modules like lane detection and object recognition (see e.g. [10]). The line segments are assumed to be calculated in an independent preprocessing stage. In our system we use an efficient algorithm that determines line segments by clustering local orientations of contour pixel obtained by a Canny-Filter [11].

The vanishing point estimation is composed of two different modules as depicted in Fig. 1. The first module performs the data-driven vanishing point estimation including a simple temporal filtering. It receives the line segments, which have been extracted from the image as input, and selects those that are most likely to point to the vanishing point. These lines are partitioned into the two sets I^l and I^r: the lines lying on the left and those lying on the right hand of the ego-vehicle. Then, for each pairwise combination of line segments from I^l and I^r the intersection points are calculated, providing single estimates $\hat{\boldsymbol{v}}_{i_l,i_r}$. Their weighted mean results in the data-driven vanishing point estimation $\hat{\boldsymbol{v}}_D(t)$, which is passed to a recursive filter. Furthermore, this unfiltered vanishing point estimation is fed into the second module, together with an associated confidence. The second module realizes the model-

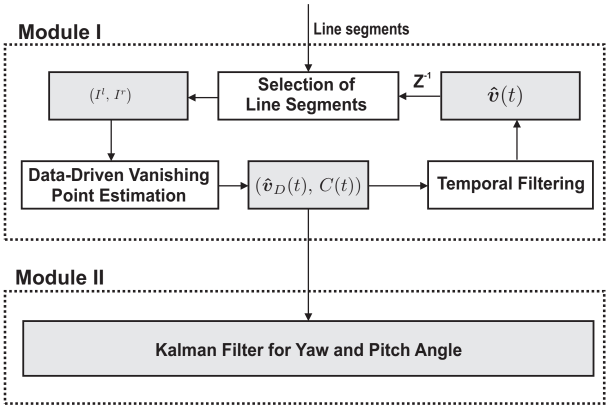
Module I
Line segments
(I^l, I^r)
Selection of Line Segments
Z^{-1}
$\hat{\boldsymbol{v}}(t)$
Data-Driven Vanishing Point Estimation
$(\hat{\boldsymbol{v}}_D(t), C(t))$
Temporal Filtering
Module II
Kalman Filter for Yaw and Pitch Angle

Fig. 1. Architecture for vanishing point estimation.

1-4244-0094-5/06/$20.00 ©2006 IEEE

based filtering of the vanishing point estimation of the first module. It consists of a Kalman filter, whose measurement noise is a function of the confidence. It is important to note that there is no feedback into the first module.

Before we describe the different submodules of our architecture in detail, the notation that will be used in the remainder of this section is shown in Table I.

TABLE I

NOTATION FOR THE DESCRIPTION OF THE VANISHING POINT ESTIMATION

Notation	**Description**
$\hat{\boldsymbol{v}}(t)$	Filtered vanishing point at time t
$\hat{\boldsymbol{v}}_D(t)$	data-driven vanishing point estimation at time t
$S = \{S_i\}$	set of all line segments
g_{S_i}	by S_i defined straight line
l_i	length of line segment S_i
α_i	slope of line segment S_i
$\boldsymbol{m}_i$	midpoint of line segment S_i
$\boldsymbol{n}_i$	normal vector of g_{S_i}
$d(S_i, \boldsymbol{v})$	distance of point $\boldsymbol{v}$ to line g_{S_i}

A. Selection of Line Segments

For estimating the vanishing point only those line segments, which are parallel to the lane borders in world coordinates should be used. Therefore, the first processing step is the selection of line segments that are subsequently used in the estimation algorithm. The following criteria ensure that only suitable lines are selected:

- The line segments must have a minimal length. Thereby, the number of line intersections that have to be calculated is limited (for n selected line segments $O(n^2)$ time is required). Further, the probability for selecting a line segment that is not due to the lane borders of the road is reduced, because lane borders in the region close to the ego-vehicle usually provide the longest lines in the image.
- The orientation of the line segments must not be horizontal or approximately horizontal. In this case the calculated intersection points from this line segment have a large variance. Additionally, this condition prevents that line segments are selected that have a large distance from the ego-vehicle in world.
- Lines to be selected must point towards the vanishing point. This property directly emerges from the definition of the vanishing point. Because the true vanishing point is unknown to the estimation process it is replaced by the estimation from the preceding time step. This is a reasonable approximation because the vanishing point is a function of the vehicle's orientation, and it's displacement between two time steps is limited (Newton's Law of Mechanics).
- The endpoints of the line segments must lie in a certain distance below the horizon. Thereby, it is assured that only lines in the region close to the ego-vehicle are selected.

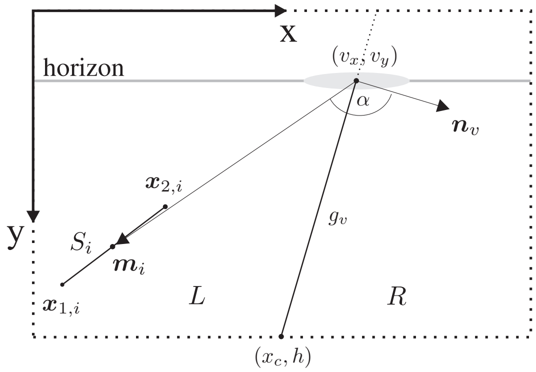

Fig. 2. Determination of the sets I^l and I^r.

On the basis of these criteria the set I is determined, which defines those line segments that are used for the estimation of the vanishing point. Precisely, using the properties of the line segments defined in Table I, the set I is specified by

$$
\begin{aligned}
I = \; \{ S_i \mid \; & l_i > l_{min} \, \wedge \, |\alpha_i| > \alpha_{min} \, \wedge \\
& d(S_i, \hat{\boldsymbol{v}}(t-1)) < d_{max}(t) \, \wedge \\
& \min(y_{1,i}, y_{2,i}) > \hat{\boldsymbol{v}}(t-1) + \Delta y_v \} \quad ,
\end{aligned}
$$

where l_{min}, α_{min}, $d_{max}(t)$ and Δy_v are parameters of the selection criterion, and the line properties are defined by

$$
\begin{aligned}
l_i &= \sqrt{(x_{2,i} - x_{1,i})^2 + (y_{2,i} - y_{1,i})^2} \\
\alpha_i &= \frac{y_{2,i} - y_{1,i}}{x_{2,i} - x_{1,i}} \\
\boldsymbol{m}_i &= \frac{1}{2}(x_{1,i} + x_{2,i}, y_{1,i} + y_{2,i})^T \\
\boldsymbol{n}_i &= (y_{1,i} - y_{2,i}, x_{2,i} - x_{1,i})^T \quad ,
\end{aligned}
$$

where $(x_{1,i}, y_{1,i})$ and $(x_{1,i}, y_{1,i})$ are the endpoints of the line segment S_i.

These lines are partitioned into the two sets I^l and I^r, such that $I = I^l \cup I^r$ and $I^l \cap I^r = \emptyset$. In Fig. 2 all relevant variables are depicted. The vector $\boldsymbol{n}_v$ denotes the normal vector of the straight line g_v and is assumed to be oriented such that $\langle \boldsymbol{n}_v, \hat{\boldsymbol{v}} \rangle > 0$. The line g_v determines two half planes and by using the sign of $\langle \boldsymbol{n}_v, \boldsymbol{m}_i - \hat{\boldsymbol{v}} \rangle$ it can efficiently be determined, whether the midpoint of a line segment $\boldsymbol{m_i}$ lies in the left (< 0) or in the right (> 0) of these planes. This directly follows from Fig. 2 and the relationship $|\boldsymbol{n}_v| \cdot |\boldsymbol{m}_i - \hat{\boldsymbol{v}}| \cos\alpha = \langle \boldsymbol{n}_v, \boldsymbol{m}_i - \hat{\boldsymbol{v}} \rangle$. Therefore, the determination of the sets is defined by

$$
\begin{aligned}
I^l &= \{ S_{i_l} \mid S_{i_l} \in I \wedge \langle \boldsymbol{n}_v(t-1), \boldsymbol{m}_{i_l} - \hat{\boldsymbol{v}}(t-1) \rangle < 0 \} \\
I^r &= \{ S_{i_r} \mid S_{i_r} \in I \wedge \langle \boldsymbol{n}_v(t-1), \boldsymbol{m}_{i_r} - \hat{\boldsymbol{v}}(t-1) \rangle \geq 0 \}.
\end{aligned}
$$

In Fig. 3 a typical example for line segment selection and partitioning in I^l and I^r is illustrated. Further, the single estimates $\hat{\boldsymbol{v}}_{i_l,i_r}$ are shown.

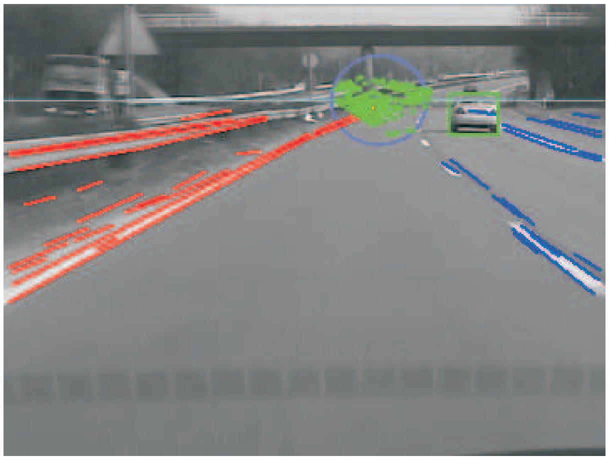

Fig. 3. Example of a result of line segment selection. All lines pointing to the vanishing point are depicted. The blue lines are on the left, the red ones on the right hand of the ego-vehicle. The green points are the single estimates $\hat{\boldsymbol{v}}_{i_l,i_r}$ from the pairwise combinations.

B. Weighting of Line Segments

Each line segment that is selected for vanishing point estimation is weighted on the basis of some criteria. The criteria based upon which the weighting factors for the line segments are calculated, aim to take into account the following aspects:

- The longer a line segment, the higher is the probability that this segment corresponds to a lane border. Further, because of the pixel grid, the accuracy of the intersection point of two lines increases with the line length.
- The greater the distance between the vanishing point and the more distant endpoint of the line segment S_i, the closer is the line segment to the ego-vehicle.

Taking these observations into account and defining $d_i = \max\left(\|(x_{1,i}, y_{1,i})^T - \hat{\boldsymbol{v}}(t-1)\|, \|(x_{2,i}, y_{2,i})^T - \hat{\boldsymbol{v}}(t-1)\|\right)$ the weight of a line S_i is determined by

$$w_i = \frac{l_i}{n_l} \cdot \frac{d_i}{n_d} \quad ,$$

where n_l and n_d are parameters for the normalization of the line length and the distance to the vanishing point, finally characterized by only one effective parameter.

C. Data-Driven Vanishing Point Estimation

The data-driven vanishing point estimation is defined by the weighted sum of the single estimates $\hat{\boldsymbol{v}}_{i_l,i_r}$ of the line segment pairs $(S_{i_l}, S_{i_r}) \in (I^l \times I^r)$:

$$\hat{\boldsymbol{v}}_D(t) = w_T^{-1} \sum_{(i_l,i_r)\in I^l \times I^r} (w_{i_l} \cdot w_{i_r}) \hat{\boldsymbol{v}}_{i_l,i_r} \ ,$$

where $w_T = \sum_{(i_l,i_r)} w_{i_l} \cdot w_{i_r}$ and the $\hat{\boldsymbol{v}}_{i_l,i_r}$ are the intersection points of the lines defined by S_{i_l} and S_{i_r}. Using $\beta_{i_l,i_r} = 1/(\alpha_{i_l} - \alpha_{i_r})$ these intersection points can be calculated according to the equation

$$\hat{\boldsymbol{v}}_{i_l,i_r} = \beta_{i_l,i_r} \begin{pmatrix} y_{1,i_r} - y_{1,i_l} + \alpha_{i_l} x_{1,i_l} - \alpha_{i_r} x_{1,i_r} \\ \alpha_{i_l}\alpha_{i_r}(x_{1,i_l} - x_{1,i_r}) + \alpha_{i_l} y_{1,i_r} - \alpha_{i_r} y_{1,i_l} \end{pmatrix} .$$

D. Confidence Estimation

In practical applications knowledge about the accuracy of the actual estimation decides whether this estimation is trusted to a greater or lesser extent. This information can be expressed by a confidence measure that enables the system to adapt it's data processing in a suitable way.

In the context of the vanishing point estimation approach presented in this paper, it is important to control the criteria for line segment selection (Section II-F) as well as for temporal filtering (Sections II-E and II-G). This controlling should be done dependent on the availability of data needed for the estimation, which is usually linked to the current environmental condition. The confidence estimation aims to represent this information in a single quantity. It is composed of three different factors, which reflect the accuracy of the vanishing point estimation:

- On both sides of the ego-vehicle there should exist enough line segments.
- Most of the line segments should approximately pass through the estimated vanishing point.
- The estimated vanishing point must lie in a physical meaningful range.

Because the confidence measure controls the selection of line segments, which in turn affects the calculation of the confidence measure, it is extremely important to assure that no instabilities can occur within this feedback loop. For this reason, for calculating the confidence measure we only take into account line segments S_i for which $d(S_i, \hat{\boldsymbol{v}}_D(t)) < r_C$ hold. Here, r_C is a constant and must be chosen, such that $r_C \leq r_{min}$. This requirement becomes obvious in Section II-F, where the online adaptation of the line segment selection parameter $d_{max}(t)$ is discussed.

The first factor used in the confidence calculation is related to the overall weight of all line segments, and is calculated by

$$W_{\hat{\boldsymbol{v}}_D(t)} = \min\left(1, \frac{1}{n_N}\sum_{i_l} w_{i_l}\right) \cdot \min\left(1, \frac{1}{n_N}\sum_{i_r} w_{i_r}\right) \quad ,$$

where n_N is a constant that is linked to the average sum of line segment weights.

High confidence implies that meaningful line segments on both sides of the ego-vehicle were available for estimation.

The second factor is based on the weighted distances of the line segments to the estimated vanishing point:

$$\bar{D}_{\hat{\boldsymbol{v}}_D(t)} = \frac{1}{\sum_i w_i} \sum_i w_i \cdot d(S_i, \hat{\boldsymbol{v}}_D(t)) \quad .$$

The third factor for the confidence measure is based on physical limitations for yaw and pitch angle, and thereby incorporates prior knowledge of the world. Using $e_x = |(\hat{\boldsymbol{v}}_D)_x(t) - v_x^0|$ and $e_y = |(\hat{\boldsymbol{v}}_D)_y(t) - v_y^0|$, this factor is defined by

$$\begin{aligned} S_{\hat{\boldsymbol{v}}_D(t)} &= \frac{1}{4}\Big(1 - \tanh\big(\gamma_x(e_x - c_x)\big)\Big) \\ &\quad \cdot \Big(1 - \tanh\big(\gamma_y(e_y - c_y)\big)\Big) \quad . \end{aligned}$$

This function assigns estimates close to the resting vanishing point (v_x^0, v_y^0) a high and more distant ones a low confidence. The parameters $c_x, c_y, \gamma_x, \gamma_y$ define the range of valid vanishing points.

For the calculation of the overall confidence measure, the three terms described before are combined:

$$C(t) = W_{\hat{\boldsymbol{v}}_D(t)} \cdot e^{-\frac{\bar{D}^2_{\hat{\boldsymbol{v}}_D(t)}}{2\sigma^2}} \cdot S_{\hat{\boldsymbol{v}}_D(t)} \quad ,$$

where σ is a normalization constant. The resulting confidence measure $C(t)$ lies in the range $0 \leq C(t) < 1$ by construction.

E. Temporal Filtering

For the temporal filtering of the data-driven vanishing point estimation a recursive low pass-filter is used:

$$\hat{\boldsymbol{v}}(t) = \hat{\boldsymbol{v}}(t-1) + \begin{pmatrix} \alpha_x(t) & 0 \\ 0 & \alpha_y(t) \end{pmatrix} (\hat{\boldsymbol{v}}_D(t) - \hat{\boldsymbol{v}}(t-1)) \ .$$

The gains for the temporal filtering are calculated using $\begin{pmatrix}\alpha_x(t) & \alpha_y(t)\end{pmatrix}^T = C(t) \cdot \Delta t \cdot \begin{pmatrix}\omega_x & \omega_y\end{pmatrix}^T$, where $C(t)$ denotes the confidence and $\begin{pmatrix}\omega_x & \omega_y\end{pmatrix}^T$ the cut-off frequency, which determines the frequency where attenuation starts in the case of optimal confidence. If the confidence is low the estimated vanishing point $\hat{\boldsymbol{v}}(t)$ is adapted only slowly.

F. Online Adaptation of Selection Criterion

The selection of the right line segments is a crucial step within the data-driven estimation process. In addition to some general unspecific parameters, this selection process is dependent on the radius parameter $d_{max}(t)$ and the filtered estimate $\hat{\boldsymbol{v}}(t-1)$.

If the confidence is low, the radius should be increased. Thereby, an possibly wrong estimated vanishing point can be recovered, and thus a reset of the vanishing point estimation is realized. If the confidence is high, it is assumed that the actual data-driven vanishing point estimation is good. To increase the accuracy and reduce the runtime of the algorithm in this case the search area is reduced by decreasing the radius $d_{max}(t)$.

The radius $d_{max}(t)$ is adapted in a linear dependence of the confidence $C(t)$:

$$d_{max}(t) = r_{min} + (1 - C(t)) \cdot (r_{max} - r_{min}) \quad .$$

The parameters are the minimum and the maximum radius r_{min} and r_{max}.

G. External Kalman Filtering

Using an additional module a model-based filtering for the vanishing point estimation is achieved. This module consists of a Kalman filter and is almost completely decoupled from module I. The communication consists of receiving the actual data-driven (unfiltered) vanishing point estimate $\hat{\boldsymbol{v}}_D(t)$ with the associated confidence $C(t)$. The data-driven vanishing point estimate serves as observation for the Kalman filter. Here, again (see Section II-E) the confidence $C(t)$ is used for controlling to which extent the data-driven estimate is tracked. This is achieved by making the measurement noise of the Kalman filter a function of the estimated confidence.

For modeling yaw and pitch angle the decay function $\dot{x} + \gamma \cdot x = 0$ is used, where x denotes the zero-centered yaw angle and the centered pitch angle, respectively. This function provides the prediction step $x_{k+1} = (1 - \Delta t \cdot \gamma) \cdot x_k$ for the Kalman filter with the model parameter γ.

One may wonder, why not to use a different model, e.g. a damped oscillation for the pitch angle: $\ddot{x} + 2\gamma \cdot \dot{x} + \omega_0^2 \cdot x = 0$, $\gamma \leq \omega_0$. But this model has shown to provide non-realistic behavior within the proposed architecture. Single estimates with high confidence can result in oscillations as defined by the model if successive estimates have low confidence.

The noise variance of the Kalman-Filter for a single measurement is determined using the confidence measure. It is adapted dependent on the confidence $C(t)$ of the actual data-driven vanishing point estimation $r : [0, 1] \rightarrow [10^{\text{minExp}}, 10^{\text{maxExp}}]$

$$r(C(t)) = 10^{\text{maxExp} + C(t) \cdot (\text{minExp} - \text{maxExp})} \quad .$$

The parameters minExp and maxExp are tuned such that for high confidence the measurement highly influences the filtering, whereas for small confidence it has almost no effect.

III. Experiments

The performance of the framework, which has been presented in Section II is evaluated using real-world grayscale images. For giving a complete picture of the performance of the estimation approach, we do different kind of analyses. In order to allow for statements of robustness, we qualitatively analyze the estimation results in a wide variety of scenarios, e.g. we investigate the behavior in the absence of meaningful line segments. We additionally compare the estimation results with the gyro-sensor data recorded in a test vehicle. To prove the real-time capabilities of the proposed architecture, we furthermore investigate the distribution of processing time.

A. Qualitative Evaluation

The evaluation of the vanishing point estimation was performed on a wide variety of scenarios reaching from autobahn to urban scenes. These scenarios contain different problem cases like rain, low sun angles, missing lane markings, occlusions from other vehicles, etc. The key observations are:

- If proper line segments exist on both sides of the ego-vehicle the vanishing point is correctly and robustly estimated. This is true for a wide variety of scenes, because these lines must not necessarily correspond to lane borders, but can also belong to curbstones or safety barriers, for example.
- If lines are completely missing on one side of the ego-vehicle, e.g. in narrow curves or if other vehicles occlude the lines, vanishing point estimation is not possible. These situations are reliably detected by the confidence measure. In these cases the estimate relaxes to the resting level.
- If the distribution of $\hat{\boldsymbol{v}}_{i_l,i_r}$ becomes multi-modal what e.g. might occur at exits, the confidence declines and

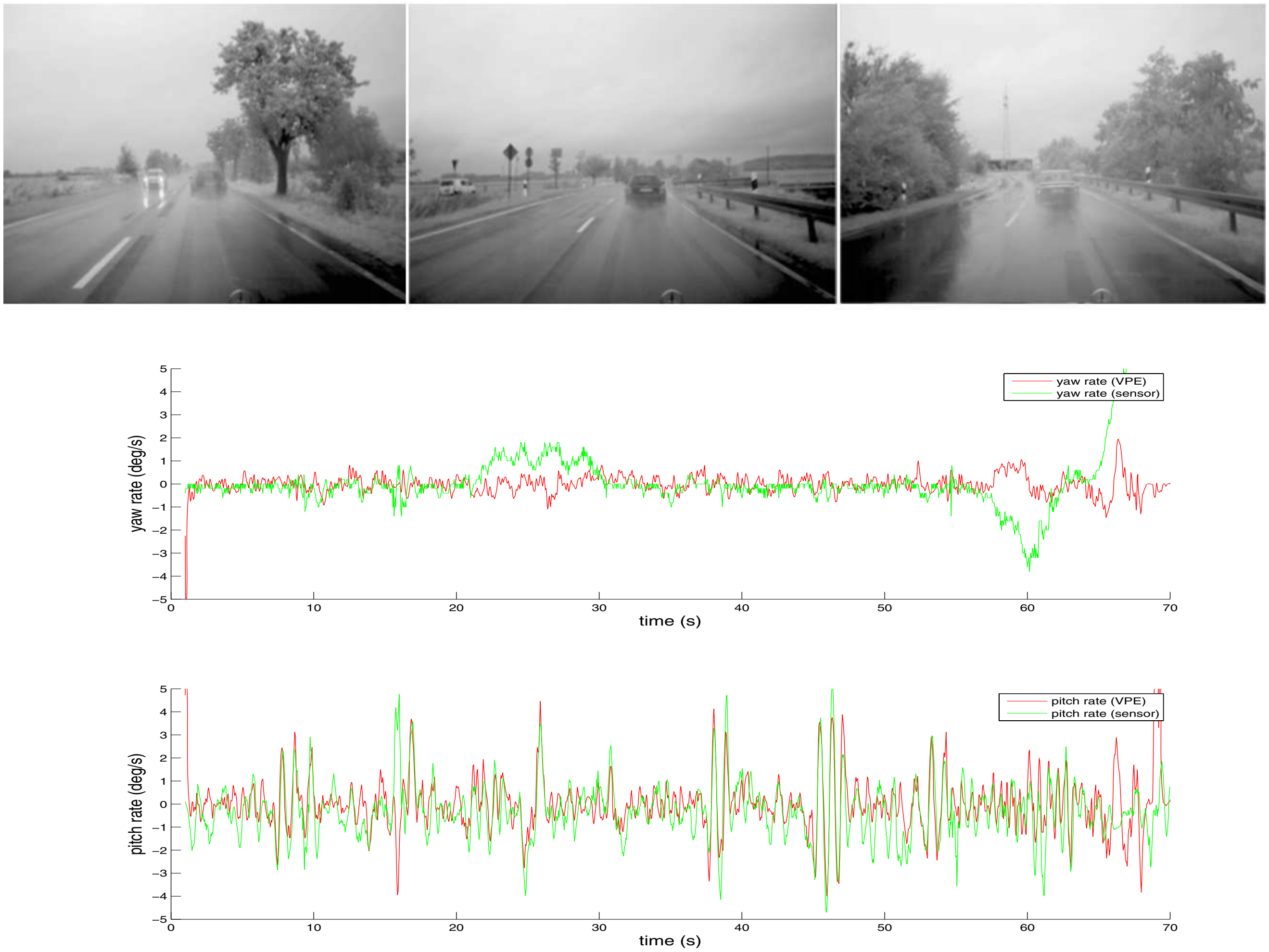

Fig. 4. Evaluation of the vanishing point estimation approach using gyro-sensor data. The sequence, evaluated in this example shows a German Bundesstrasse with rainy weather. The first peak of the yaw rate from the sensor data corresponds to a curve to the right, the second peak to a curve to the left. The different behavior of the results from the vanishing point estimation is due to the discussed different frames of reference. The pitch rate estimations from the sensor and from the vanishing point estimation are very similar.

no real vanishing estimation takes place (relaxation of the estimation to the resting level). As soon as only one vanishing point remains relevant to the ego-vehicle, the vanishing point estimation is continued.

The problems mentioned are within the nature of vanishing point estimation. Missing lines on one side of the ego-vehicle remain a major critical problem. The attempt to estimate the vanishing point in such cases will most likely lead to an inaccurate estimation. Besides, the lines on one side of the ego-vehicle can be used to estimate one coordinate of the vanishing point, when the other coordinate is kept constant.

If two vanishing points occur, which is not conform with our model implemented, the proposed architecture can be extended. For example, we have shown that exits can explicitly be detected, resulting in a more accurate vanishing point estimation.

B. Evaluation with Gyro-Sensor Data

One substantial requirement for the quantitative analysis of the algorithm is the availability of ground-truth data. Because precise labeling of recorded sequences is not possible, especially for large data sets, we use sensors for measuring yaw and pitch rate, which are installed inside a test vehicle. The problems of comparing the vanishing point estimation with the (noisy) sensor outputs will become apparent in the discussion below.

In order to allow for a comparison of the results of the vanishing point estimation with the gyro-sensor data, we use the former to calculate yaw and pitch rates. This is done by first transforming the vanishing point estimation to yaw and pitch angle using the camera parameters, and then determining the yaw and pitch rates.

We briefly discuss the effect of the different reference frames, which are not negligible, exemplary for the yaw angle.

The yaw angle of the ego-vehicle with regard to an absolute coordinate system $\Psi_V(t)$ is composed of the angle between the street at height of the ego-vehicle and the absolute coordinate system $\Psi_L(t)$ and of the angle between the ego-vehicle and the street $\Psi_{WB}(t)$: $\Psi_V(t) = \Psi_L(t) + \Psi_{WB}(t)$. It directly follows by differentiating that

$$\dot{\Psi}_V(t) = \dot{\Psi}_L(t) + \dot{\Psi}_{WB}(t) = c(t) \cdot v(t) + \dot{\Psi}_{WB}(t) \quad ,$$

where $c(t)$ is the curvature of the road and $v(t)$ the velocity of the ego-vehicle.

The vanishing point estimation provides $\Psi_{WB}(t)$, whereas the sensor for the yaw rate provides $\dot{\Psi}_V(t)$. Thus, for a comparison between the vanishing point estimation and the sensor data the curvature of the road and the velocity of the ego-vehicle are needed (see Fig. 4 for the effects).

The yaw and the pitch rate are calculated from the vanishing point estimation using the camera parameters and applying the filter $\frac{1}{8} \cdot [-1 \ -2 \ 0 \ 2 \ 1]$. The yaw rate from the sensor is not filtered at all, whereas for the pitch rate the floating mean over a window width of 10 is calculated.

We show one exemplary result in Fig. 4. This sequence contains relative strong pitch motion and partly missing lane markings.

C. Runtime of Vanishing Point Estimation

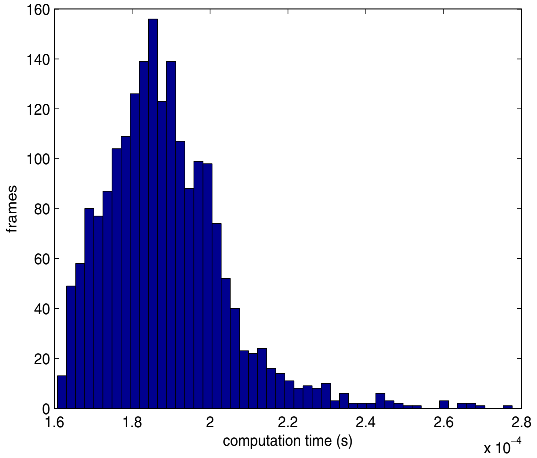

Fig. 5. Distribution of computation times of the vanishing point estimation algorithm evaluated on an AMD 1.6 GHz CPU.

Besides robustness, the real-time capability of the estimation approach is an important matter. Because in practical applications the mean computation time, needed to process one frame is not primary relevant, we determined the distribution of computing time. For this purpose, we used four video sequences (1000-2000 frames each), all of them containing entirely different scenarios.

On an AMD 1.6 GHz processor the mean computation time turned out to be below 0.2 ms, the maximum time per frame below 0.4 ms. One of the resulting histograms of the analyses is depicted in Fig. 5.

The time used for vanishing point detection scales with n^2 in the number of the selected line segments (calculation of intersection points). So, the computation time can always be restricted by selecting only a predefined maximal number of line segments.

IV. Conclusion

We presented a vision-based architecture for vanishing point estimation that focuses on robustness and real-time capability. Vanishing point estimation is based on the calculation of intersection points of line segments from the left and the right side of the ego-vehicle. Model-based behavior of the estimation is achieved by a Kalman filter. We introduced a confidence measure that reliably indicates the accuracy of the vanishing point estimation and thereby provides important information for the estimation process. This information is used for the online adaptation of the line selection criterion as well as the adaptation of the measurement noise of the Kalman filter.

The performance of the estimation approach was successfully demonstrated in a wide variety of real road conditions. The estimation has proved to be robust as long as line segments exist on both sides of the ego-vehicle. Situations in which no reasonable estimation is possible are reliably detected by the confidence measure.

References

[1] E. Dickmanns and B. 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] A. Guiducci, “Parametric model of the perspective projection of a road with applications to lane keeping and 3D road reconstruction,” *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 414–427, 1999.

[3] E. Lutton, H. Maître, and J. Lopez-Krahe, “Contribution to the determination of vanishing points using hough transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, pp. 430–438, 1994.

[4] G. McLean and D. Kotturi, “Vanishing point detection by line clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 11, pp. 1090–1095, 1995.

[5] J. Košecká and W. Zhang, “Efficient computation of vanishing points,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA’02)*, vol. 1, 2002, pp. 223–228.

[6] C. Rother, “A new approach for vanishing point detection in architectural environments,” *Image and Vision Computing*, vol. 20, no. 9-10, pp. 647–656, 2002.

[7] N. Simond and P. Rives, “Homography from a vanishing point in urban scenes,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 1, 2003, pp. 1005–1010.

[8] C. Rasmussen, “Grouping dominant orientations for ill-structured road following,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 1, 2004, pp. 470–477.

[9] Y. Wang, E. Teoh, and D. Shen, “Lane detection and tracking using B-Snake,” *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.

[10] T. Bücher, C. Curio, J. Edelbrunner, C. Igel, D. Kastrup, I. Leefken, G. Lorenz, A. Steinhage, and W. von Seelen, “Image processing and behaviour planning for intelligent vehicles,” *IEEE Transactions on Industrial Electronics*, vol. 90, no. 1, pp. 62–75, 2003.

[11] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.