A Generic Temporal Integration Approach for Enhancing Feature-based Road-Detection Systems

Thomas Michalke, Robert Kastner, Jannik Fritsch, Christian Goerick

2008

Preprint:

This is an accepted article published in Intelligent Transportation Systems Conference. The final authenticated version is available online at: https://doi.org/[DOI not available]

A Generic Temporal Integration Approach for Enhancing Feature-based Road-detection Systems

Thomas Michalke*, Robert Kastner*, Jannik Fritsch◇, Christian Goerick◇

*Darmstadt University of Technology
Institute for Automatic Control
D-64283 Darmstadt, Germany
thomas.michalke,robert.kastner
@rtr.tu-darmstadt.de

◇Honda Research Institute Europe GmbH
D-63073 Offenbach, Germany
{jannik.fritsch,
christian.goerick}
@honda-ri.de

***Abstract*— Visual feature-based approaches for detecting the drivable area on unmarked streets and roads were introduced in recent years. Although the accumulated results are promising, the detected street segments often contain holes and show a detection performance that strongly varies in time depending on environmental conditions. This paper presents a real-time capable approach for improving the road detection results for this type of state-of-the-art systems by adding a generic postprocessing step. Our proposed architecture removes the drawbacks of said systems using a temporal integration approach based on the bird's eye view. In order to test the proposed approach, one typical visual feature-based road detection system was implemented. Still, the used road detection system can be exchanged with any other state-of-the-art system. Evaluation results computed on inner-city data show that this approach is an important enhancement for all visual feature-based road-detection systems. One of the used streams and corresponding ground truth data is accessible on the internet for benchmark testing. The proposed approach is a crucial step toward robust road detection in complex scenarios that allows building high-level applications, as, e.g., active collision avoidance or trajectory planning, based on vision as the major cue.**

Keywords: driver assistance, robust path identification, lane detection

I. INTRODUCTION

The importance of driver assistance systems for further decreasing the number of traffic accidents is a widely acknowledged fact. The growing complexity of tasks, handled by Advanced Driver Assistance Systems, lead to complex systems that use information fusion from many sensory devices and incorporate processing results of other modules. One important field of interest for said systems are applications that are based on a robust detection of the drivable path, as e.g., the "Honda Intelligent Driver Support System" [1] supporting the driver to stay in the lane and to maintain a safe distance from the car in front. Other systems focus on collision avoidance based on autonomous steering and braking (see, e.g., [2]) as well as path planning even in unstructured environments (see, e.g., [3]). The more safety-relevant applications become, the more the required quality of the detected drivable path must be improved.

Initial approaches for lane detection on marked roads date back to the 1990s (see [4] for an overview of the early approaches). These to date commercially available systems are restricted to marked roads with a predictable course based on a clothoid lane model that is also used for road construction of motor-ways. In recent years, the field of operation for path detection has shifted to unmarked country roads and inner-city streets. To date, such systems rely on statistical evaluation of different image features and primarily use a training region in front of the car to determine the current existing street characteristics (see, e.g., [5] and [6]).

The street segments that are detected by these approaches often contain false negative pixels (regions that belong to the street but are not detected as such, i.e., holes in the detected street segment) and the detection performance often changes dynamically in time. The varying detection performance is due to the changing content of the training region in front of the car. Thereby, the system possibly adapts to local characteristics present in the current training region that might differ from the global road characteristics. Furthermore, local illumination changes that depend on the current view angle and lighting conditions influence the detection performance. See Fig. 1 for a visualization of both effects. Future safety-relevant applications (such as, e.g., autonomous steering of a car based on long-term trajectory planning) require both, a dense and stable detected road segment as input data.

In this paper, we present a fast and robust method for improving the quality of existing state-of-the-art road-detection systems by including information of the street segments detected in the past - a concept termed temporal integration. The proposed approach is an important step towards robust driving path detection on unmarked roads in complex environments, as for example in inner city.

II. RELATED WORK

The concept of temporal integration is used in various applications in the field of computer vision for driver assistance. For example, [7] uses spatiotemporal integration to improve the classifier performance when detecting signal boards and cars. Other applications for improving the classifier performance rely on (temporal integration based) voting mechanisms, which are widely used in numerous domains (see [8] for an overview). Also the well-known Kalman filter approach [9] stabilizes its state estimate by temporal integration (fusion of measured and predicted data). In [10] temporal integration is used to determine the camera

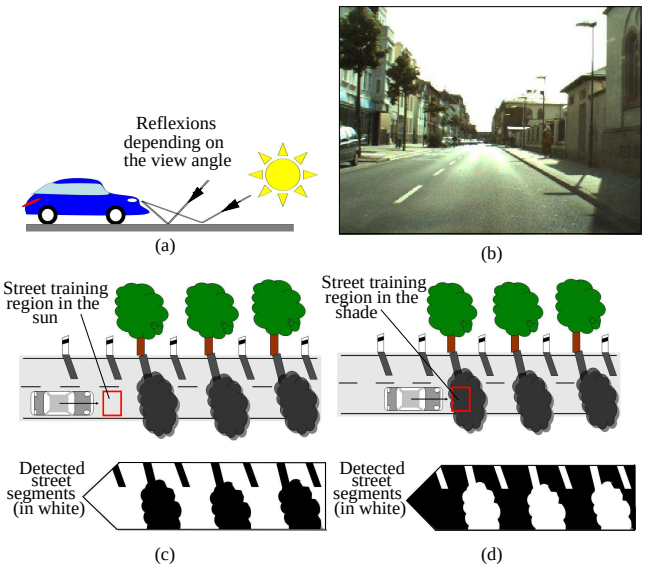


Fig. 1. Causes for varying road-detection performance: (a) Illumination change with dependance on the view angle, (b) Sample image showing typical illumination gradient, (c) Schematic example: Training region in the sun and resulting detected street segment (in white), (d) Schematic example: Training region in the shade and resulting detected street segment (in white).

parameters, thereby stabilizing the input image of a marked lane detection system running online in a car.

Also for clothoid model based lane detection on highways and country roads (see, e.g., [11] and [12]) temporal integration was found to improve the detection performance. Still, the usage of such model based approaches for road detection in complex inner-city scenes is heavily restricted, due to the unpredictable and abruptly changing course of the road and various occlusions of road parts. Figure 2a shows the complexity of a hand labeled ground truth road segment for an inner-city frame that can hardly be modeled using, e.g., a clothoid model. Therefore, also a model based temporal integration is not possible and will not show the desired results in such complex scenarios.

Newer road-detection approaches that rely on the statistical evaluation of different image features (see, e.g. [5] and [6]) can handle such scenarios but have the drawbacks discussed in the introduction. Nevertheless, also for these systems temporal integration can and should be used for making the road segment detection more robust. To this end, the most direct approach would be to use the optical flow that reflects the magnitude and direction of the motion of image regions, as shown in Fig. 2b. Based on that, the current position of a street segment detected in the past can be determined and used for a fusion with the current road-detection results. However, the optical flow has certain drawbacks. Firstly, it's to date high computational costs make it scarcely applicable in domains with hard real time constrains, as the car domain. Secondly, the optical flow cannot be calculated at the borders of an image and is error prone due to ambiguities resulting from the aperture problem, illumination change, and camera noise [13]. Instead of detecting the motion of all image regions based on the

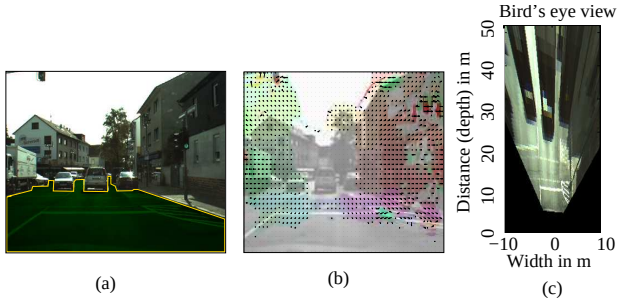


Fig. 2. Frame 105 of inner-city stream: (a) Hand-labeled ground truth street segment, (b) Optical flow (colors code the direction of the motion), (c) Bird's eye view.

optical flow, the approach proposed here concentrates on the drivable street plane alone, relying on the bird's eye view (see Fig. 2c and Fig. 4a).

III. SYSTEM DESCRIPTION - TEMPORAL ROAD SEGMENT INTEGRATION

In the following, a rough overview of our approach of bird's eye view based temporal road integration is given (see Fig. 3). Thereafter, all processing steps and their theoretical background are described in more detail.

As input data our system uses 400x300 monocular gray value images and a binary map of the currently detected street segment. The images are used for calculating the bird's eye view, which is a representation of the scene as viewed from above (see Fig. 4a). In the following step, the bird's eye view is used for detecting the motion of the static vehicle environment based on Normalized Cross Correlation (NCC). Based on these correlation results the current and past street segments are fused by temporal integration on the bird's eye view. The fused street segments are then mapped back to the perspective view corresponding to the input image.

The system takes optional input data that improves the quality and makes the temporal integration more robust. As such optional input data stereo images as well as the longitudinal ego velocity and yaw rate of the CAN bus of our prototype vehicle are processed. The depth map that is calculated from stereo images (using the commercial "Small Vision System" [14]) is the basis for correcting the changes in the pitch and roll angle. These changes induce the bird's eye view to be unstable in case the car brakes or the street profile is not flat. The CAN data is used for predicting the car motion based on a single track model. The predicted motion is used for determining the anchor for the correlation on the bird's eye view. The usage of CAN data makes the system faster. Still, without CAN data the detection quality is not reduced.

In the following, the processing steps (as depicted in Fig. 3) are described in more detail. Firstly, the camera lens distortion is corrected. In Eq. (1) and Eq. (2) the undistorted vertical and horizontal pixels v and u are computed on the initial (distorted) vertical and horizontal pixels v_d and u_d. The undistortion is based on a lens distortion model (described in [15]) that uses radial (k_1 and k_2) and tangential

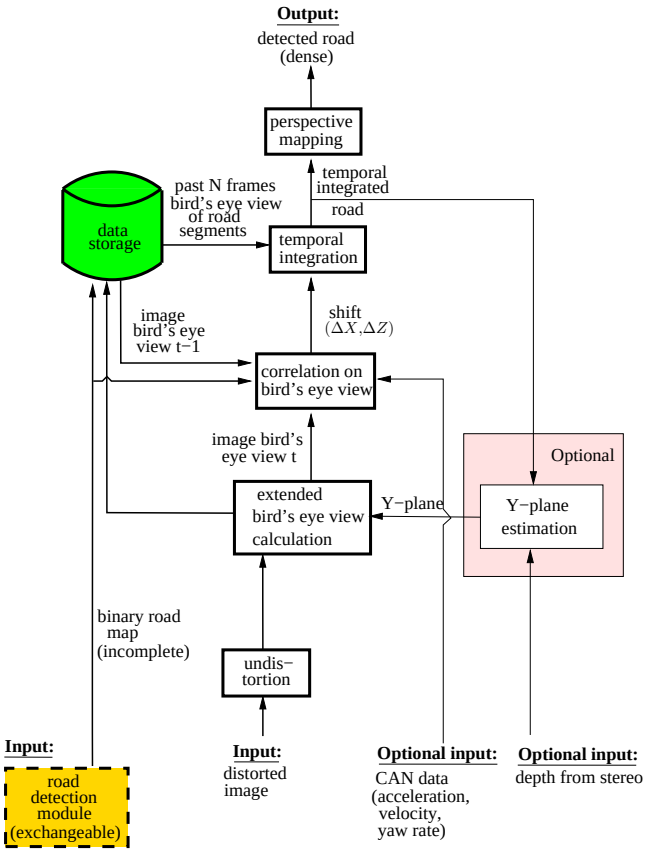


Fig. 3. System structure: Temporal road segment integration (the dashed module can be exchanged with the road-detection algorithm preferred by the user, optional module is highlighted in red).

distortion coefficients (k_3 and k_4). The undistortion step is essential in order to allow a correct mapping of the image pixels to the bird's eye view. It is important to note, that for the bird's eye view as a metric representation, the undistorition step makes sure that the proportions in the bird's eye view match the real world.

$$u = (1 + k_1\beta^2 + k_2\beta^4)u_d + 2k_3u_dv_d + k_4(\beta^2 + 2u_d^2) \quad (1)$$

$$v = (1 + k_1\beta^2 + k_2\beta^4)v_d + k_3(\beta^2 + 2u_dv_d) + 2k_4u_dv_d \quad (2)$$

$$\text{with } \beta = \sqrt{u_d^2 + v_d^2}$$

Then the bird's eye view is calculated on the undistorted pixels v and u based on Eq. (3) and Eq. (4) by inverse perspective mapping of the 3D world points X, Y, and Z (see Fig. 4b for the notation in our coordinate system) to the 2D (u,v) image plane. The equations describe how to map a 3D position of the world to the 2D image plane (refer to [4]). More specifically, only the image pixels (u,v) that are needed to get a dense metric bird's eye view plane are mapped into the XZ-plane. The usage of inverse perspective mapping makes the inversion of Eq. (3) and (4) for calculating the bird's eye view obsolete. Eq. (3) and (4) use the 3 camera angles θ_X, θ_Y, and θ_Z, the 3 translational camera offsets t_1, t_2, t_3 (see Fig. 4b), the horizontal and vertical principal point c_u and c_v as well as the horizontal and vertical focal lengths f_u and f_v. The intrinsic (i.e., internal camera properties, like the focal length and the principal point) and extrinsic (i.e., external camera properties, like camera angles and offsets) camera parameters were determined using the freely available calibration toolbox [16] and a calibration scene similar to the one described in [17].

$$u = -f_u\frac{r_{11}(X-t_1) + r_{12}(Y-t_2) + r_{13}(Z-t_3)}{r_{31}(X-t_1) + r_{32}(Y-t_2) + r_{33}(Z-t_3)} + c_u \quad (3)$$

$$v = -f_v\frac{r_{21}(X-t_1) + r_{22}(Y-t_2) + r_{23}(Z-t_3)}{r_{31}(X-t_1) + r_{32}(Y-t_2) + r_{33}(Z-t_3)} + c_v \quad (4)$$

with $Y = 0$,

$$R = R_XR_YR_Z = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

and

$$\begin{aligned}
r_{11} &= cos(\theta_Z)cos(\theta_Y) \\
r_{12} &= -sin(\theta_Z)cos(\theta_X) + cos(\theta_Z)sin(\theta_Y)sin(\theta_X) \\
r_{13} &= sin(\theta_Z)sin(\theta_X) + cos(\theta_Z)sin(\theta_Y)cos(\theta_X) \\
r_{21} &= sin(\theta_Z)cos(\theta_Y) \\
r_{22} &= cos(\theta_Z)cos(\theta_X) + sin(\theta_Z)sin(\theta_Y)sin(\theta_X) \\
r_{23} &= -cos(\theta_Z)sin(\theta_X) + sin(\theta_Z)sin(\theta_Y)cos(\theta_X) \\
r_{31} &= -sin(\theta_Y) \\
r_{32} &= cos(\theta_Y)sin(\theta_X) \\
r_{33} &= cos(\theta_Y)cos(\theta_X)
\end{aligned}$$

As can be seen in Eq. (3) and (4) the 3D world position coordinates X, Y, and Z of all image pixels (u,v) are needed. By using a monocular system, one dimension (the depth Z) is lost. A solution to this dilemma is the so called flat plane assumption. Here, for all pixels in the image, the height Y is set to 0. Based on this, only objects in the image with $Y = 0$ (especially, the street we are interested in) are mapped correctly to the bird's eye view, while all the other regions are stretched to infinity in the bird's eye view (for example the car in Fig. 7b).

In case this assumption is not fulfilled (i.e., the street surface is not flat) the bird's eye view is inaccurate, which leads to decreasing quality of the temporal integration. To allow a stable bird's eye view even in case of non-flat street surfaces and pitching of the vehicle, stereo data from our stereo camera setup is used. In order to enhance the robustness of the correction, only pixels that belong to the currently detected street segment are used for surface estimation. More specifically, the differences between the orientation and position of the coordinate axes and the street surface in terms of the pitch $\Delta\theta_X$ and roll angle $\Delta\theta_Z$, as well as the height of the camera over the ground Δt_2 are detected (see Eq. (6), (7), and (8)). This is done based on the 3D position for all image pixels derived from the stereo disparity (see Fig. 5 for 3D data of a sample image). The flat plane assumption $Y = 0$ is then replaced by $Y = f(X, Z)$

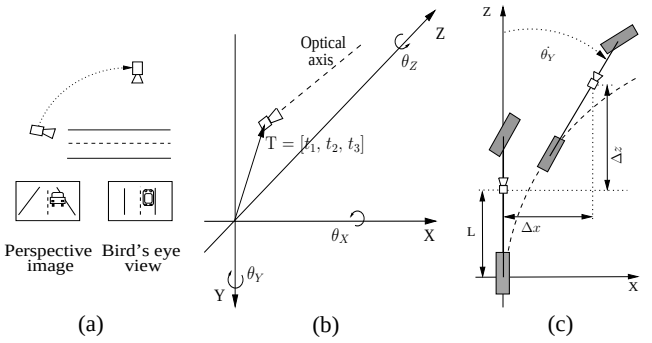


Fig. 4. (a) Visualization of the bird's eye view, (b) Coordinate system and position of the camera (car is heading in Z-direction), (c) Single track vehicle model.

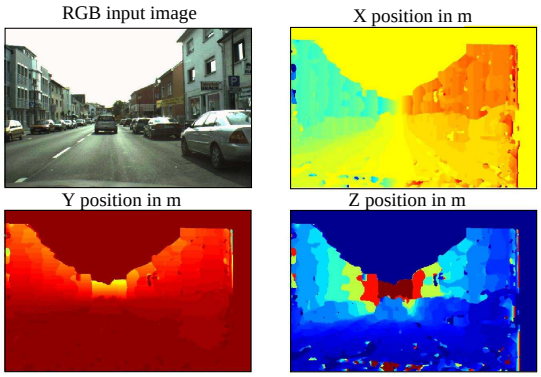


Fig. 5. Dense 3D world positions for all image pixels based on stereo.

leading to an extended bird's eye view. In our implementation a first order model for the street surface (linear hyperplane) is used as shown in Eq. (5) (see [18] for more details). Results have shown that higher order models lead to inferior performance. The reason for this is the restraint number of 3D measurement points at the borders of the image, since only reliable pixels belonging to the detected street are used for the surface estimation.

$$Y = Y_0 + aZ + bX \quad (5)$$

$$\Delta\theta_Z = atan(b) \quad (6)$$

$$\Delta\theta_X = atan(a) \quad (7)$$

$$\Delta t_2 = Y_0 \quad (8)$$

Since the estimated surface is noisy (stereo data is calculated based on error prone correlation between the left and the right image), a linear Kalman filter is used on the parameters Y_0, a, and b that raise the performance considerably. A possible improvement would be to use a model of the vehicle kinetics (containing damper and spring characteristics, distribution of the vehicle mass) for the Kalman prediction (as proposed in [19]) instead of the linear prediction model used here.

By NCC based correlation between the current and the stored previous bird's eye view the vehicle motion (ΔX, ΔZ) since the previous time step is detected. A single track vehicle model, as depicted in Fig. 4c, predicts the starting point $x_t = x_{t-1} + \Delta x$ and $z_t = z_{t-1} + \Delta z$ of the NCC correlation patch of time step t-1 in the current bird's eye view map. The values Δx and Δz are calculated based on the sample time T, the distance of the camera from the rear wheel L, as well as the yaw rate $\dot{\theta}_Y$, and lateral velocity $\dot{Z}$ from the CAN bus (see single track model Equations (9) and (10)). The derived longitudinal and lateral motions as well as rotational change (i.e., yaw angle) between the current and the previous bird's eye view are stored along with the incremental motion between the previous $N = 40$ frames (equivalent to 4 seconds of processing by our prototype vehicles' vision system).

$$\Delta x = \frac{\dot{Z}}{\dot{\theta}_Y}(1 - cos(\dot{\theta}_Y T)) + sin(\dot{\theta}_Y T)L \quad (9)$$

$$\Delta z = \frac{\dot{Z}}{\dot{\theta}_Y} sin(\dot{\theta}_Y T) + cos(\dot{\theta}_Y T)L - L \quad (10)$$

The NCC correlation patch on the bird's eye view is selected to contain enough structure (using the entropy-based measure described in [20]), which improves the accuracy of the NCC. Furthermore, it is assured that the patch belongs to the detected street and that it is not too far away from the ego vehicle, since the resolution of the bird's eye view decreases with growing distance to the vehicle.

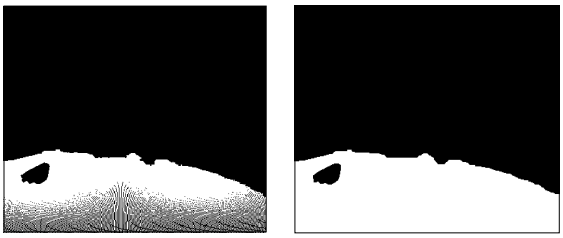
The bird's eye view maps of the detected street segments of the previous $N = 40$ frames are calculated and stored. The stored incremental motion during the past 4 seconds is integrated and used to shift all stored bird's eye view street segments correspondingly. Then the shifted previous 40 bird's eye view street segments are weighted (weights α_t) and summed up by Eq. (11). Thereafter, in Eq. (12) the sum of the street segments $S_{\text{integ}}(X, Z)$ is related to the maximum possible number of overlaid street segments $S_{\text{max}}(X, Z)$, which results in an Integrated Road Probability Map (IRPM). Please note that $S_{\text{max}}(X, Z)$ changes depending on the position in the bird's eye view map. The final threshold operation (Eq. (13)) determines the final temporal integrated street segment S_{final} in the bird's eye view representation.

$$S_{\text{integ}} = \sum_{t=1}^{N} \alpha_t S_t \quad \text{with} \quad \sum_{t=1}^{N} \alpha_t = N \quad (11)$$

$$IRPM = \frac{S_{\text{integ}}(X, Z)}{S_{\text{max}}(X, Z)} \quad (12)$$

$$S_{\text{final}} = \begin{cases} 1 & \forall\ \text{IRPM}(X, Z) \geq \beta \\ 0 & \forall\ \text{IRPM}(X, Z) < \beta \end{cases} \quad (13)$$

The weight α_1 in Eq. (11) is set high to ensure that the pixels in the current detected street segment are with a high probability also present in the final temporal integrated street segment. The other weights α_t could be set dynamically dependent on a quality measure of the bird's eye view based NCC or the road-detection system as well as the capturing



(a) (b)

Fig. 6. Final morphological fill operation for closing spaces in the street segment that are due to perspective mapping (justified holes are preserved). (a) Raw perspectively mapped street segment, (b) After morphological closing.

time t. The threshold β in Eq. (13) is currently set to 0.7. This means that a pixel is classified as street if at least 70% of the overlaid past street segments have voted for street.

Next, the final temporal integrated street segment S_{final} is mapped back to the image using Eq. (3) and (4). For this operation the resolution of the street segment in the bird's eye view representation needs to be high (which is done by upsampling the size by factor of 4) in order to allow a lossless perspective mapping of the street segment. The perspective mapping step produces equidistant, periodic spaces in the street segment directly in front of the car (see Fig. 6a). These spaces are filled using a morphological close operation with a small morphological structuring element to prevent adding too many false positive street pixels (see Fig. 6b). In other words, holes in the bird's eye street segment (that, e.g., correspond to objects on the street) are retained in the final perspectively mapped street segment.

The following section shows that the steps proposed above result in an enhanced detected street segmentation. The final detected street segment has fewer holes and is dynamically more stable than that of other approaches, which allows complex path-related applications.

IV. RESULTS

In this section, we evaluate the performance of our system by applying it to the results of a specific state-of-the-art road-detection algorithm. Additionally, the reached computation time for the proposed temporal integration approach is given.

The implemented road-detection approach processes data of a training region in front of the car as well as HSI color, structure, and stereo disparity as input features that are combined based on iconic processing (with probability values for all pixels) and region growing. The implementation is roughly related to the system described in [5]. However, the proposed temporal integration approach can work on top of all road-detection algorithms for unmarked roads and is therefore interchangeable.

Figure 7 shows results of the various system modules of our system. The depicted snapshot is part of a result stream showing our system running on 160 consecutive frames of an inner-city course. The input images and stereo data used for the evaluation as well as the ground truth data and results are accessible on the internet [21] for open benchmark testing. Additionally, Fig. 7e shows a kind of 360° representation of the environment that is derived from the combination of all stored bird's eye view maps of the past 4 seconds. This representation builds up gradually, after the algorithm starts. It could be used for higher-level trajectory planning algorithms.

The white rectangle in Fig. 7b and d-f represents the position of our prototype vehicle, while the black regions are outside the field of vision of our vehicle cameras.

In order to evaluate our algorithm with respect to its impact on the road-detection performance, we adopt the Equations (14), (15), and (16). The equations define different ground truth based measures, which were taken from [22].

$$\text{Completeness} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{14}$$

$$\text{Correctness} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{Quality} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{16}$$

with

TP ... True positive pixels

FN ... False negative pixels

FP ... False positive pixels

On a descriptive level the Completeness states, based on given ground truth data, how much of the present street was actually detected. The Correctness states how much of the detected street is actually street. The Quality combines both measures. Its computation is appropriate, since a trade-off between the Completeness and Correctness is possible. Based on this, the Quality measure should be used for a comparison, since it weights the FP and FN pixels equally. For a more detailed analysis the Completeness and Correctness state what exatly caused a difference in Quality. The necessary ground truth data was produced by accurate manual annotation of the 440 test images (see Fig. 2a).

The three measures were then calculated on the detected street segments of 440 image frames of two inner-city streams. The gathered results are depicted in Tab. I. There, the standard street detection algorithm without temporal integration is compared to our approach. Furthermore, our approach is compared to one that uses the optical flow for temporal integration (based on the state-of-the-art optical flow algorithm described in [23]), and finally to our approach using only the mandatory input data (without the usage of stereo data). In all 4 cases the same algorithm for detecting the street was used, in order to allow a comparison. As the results in Tab. I show, the highest Quality (89.9% enhancing the 60.5% of the initial street detection algorithm) is reached with temporal integration based on our algorithm. Without stereo data our algorithm still reaches a Quality of 81.7%. Optical flow based temporal integration reaches a Quality of merely 68.1%, which is due to the well-known apperture problem (see, e.g., [23]) and present illumination changes. The initial road detection approach without temporal integra-

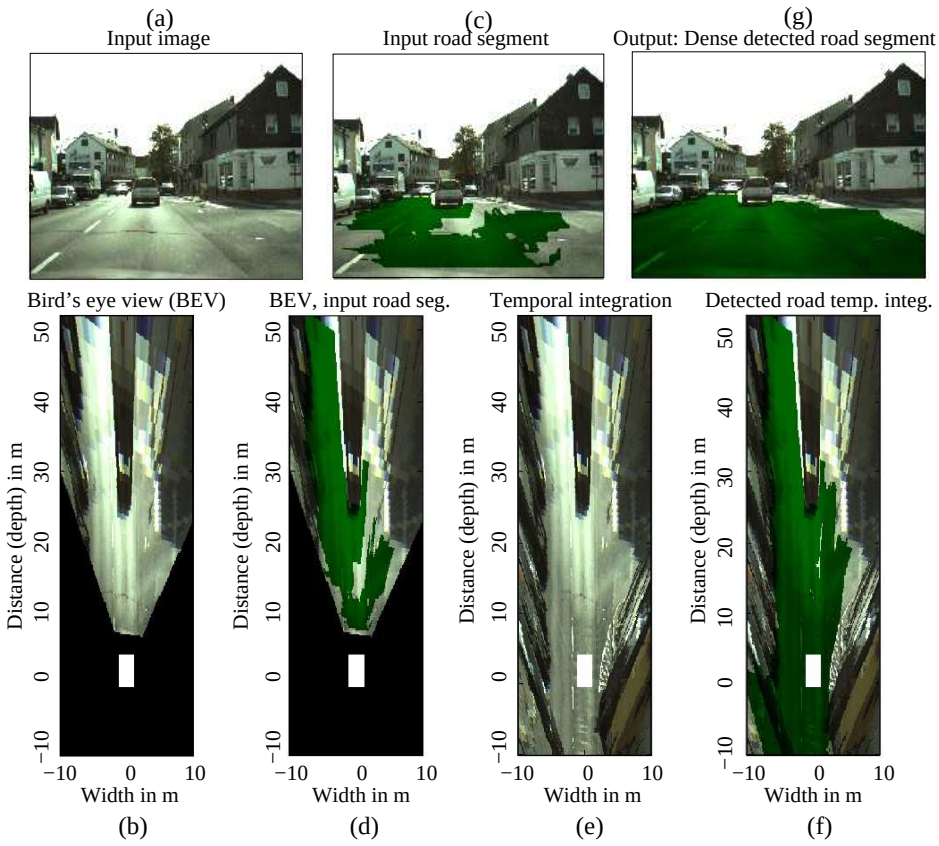


Fig. 7. (a) System input image, (b) Input image in bird's eye view, (c) System input: Detected road segment of road-detection module, (d) Detected road segment in bird's eye view, (e) Temporal integration of bird's eye view images of past 4 seconds, (f) Temporal integration of detected road segments, (g) System output: Integrated road segment mapped back to the perspective image.

Road detect. approaches (BEV: bird's eye view)	**# test images**	**Correct-ness**	**Comple-teness**	**Quality**
No temp. integration	440	98.1%	61.5%	60.5%
Temp. integration, BEV	440	**95.2%**	**94.1%**	**89.9%**
Temp. integration, optical flow	440	92.6%	72.4%	68.1%
Temp. integration, BEV, without stereo	440	96.9%	84.0%	81.7%

TABLE I

COMPARISON OF DIFFERENT METHODS FOR TEMPORAL INTEGRATION.

Module / submodule	**Comp. time [in ms] (frame rate [in Hz])**
Temp. integration, BEV	**Σ 49.8 ($\approx$ 20)**
Bird's eye view	6.9
Correlation submodule	14.7
Temp. integration	20.0
Perspective mapping to image plane	8.2
Temp. integration, optical flow	**$\Sigma >$537.0 ($\approx$ 2)**

TABLE II

COMPARISION OF COMPUTATIONAL DEMANDS.

tion has the highest Correctness with 98.1%, but this comes to the cost of reduced Completeness of merely 61.5%. Our approach applying temporal integration decrease this value from 98.1% to 95.2%, but it increases the Completeness disproportionately (from 61.5% to 94.1%).

For further evaluation Fig. 8 shows typical results of a standard street detection algorithm compared to results gathered with the proposed temporal integration approach based on 4 sample images of the inner-city stream.

For the experiments we use a Honda Legend prototype car equipped with a mvBlueFox CCD color camera from Matrix Vision delivering images of 800x600 pixels at 10Hz, which is hence the processing rate our road detection module must at least reach. The image data as well as the laser and vehicle state data from the CAN bus is transmitted via LAN to several Toshiba Tecra A7 (2 GHz Core Duo) running our RTBOS integration middleware [24] on top of Linux. The road detection component together with other driver assistance components (see, e.g., [25]) are implemented in C using an optimized image processing library based on the Intel IPP [26]. The road detection component is set to run on a single core.

Table II shows the computational demands of different submodules of the presented approach and compares these to the qualitatively inferiour approach based on the optical flow (as was shown in Tab. I). The reasonable parametrized state of the art optical flow implementation (based on [23]) needs 537.0 ms ($\approx$ 2 Hz), without taking further system modules into account, which are additionally required by this approach. The overall computation time of our temporal integration system amounts to 49.8 ms ($\approx$ 20 Hz), which allows real-time processing on our prototype vehicle.

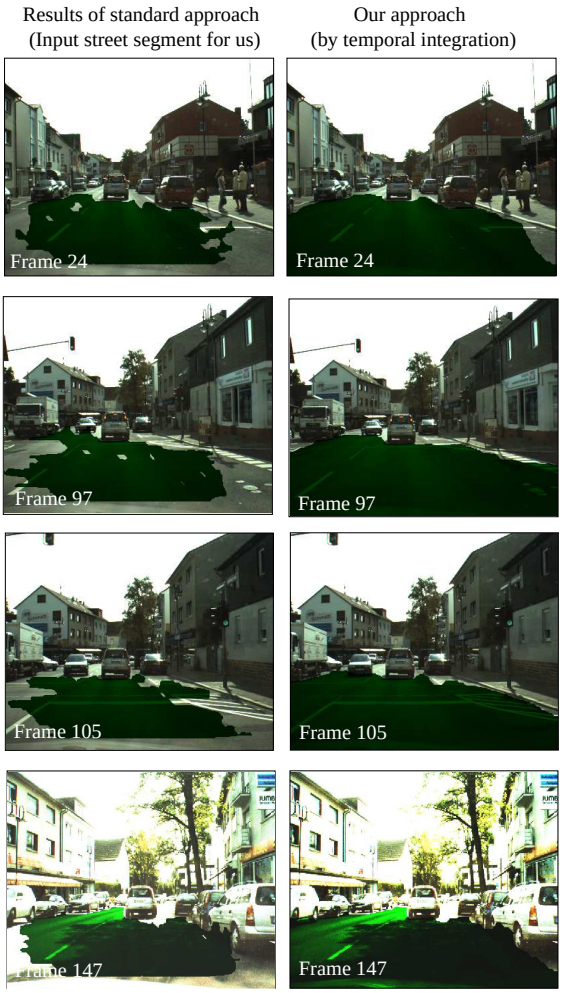


Fig. 8. Example images of the used inner-city stream (left: Standard approach (input street segment for our approach), right: Our approach (by temporal integration), the last image is visually enhanced to improve its legibility when printed

V. SUMMARY

This paper describes a generic and fast method for enhancing the quality of all state-of-the-art visual feature-based road detection systems. The proposed temporal integration approach improves the road detection performance, which allows building safety-relevant algorithms as trajectory planning and active collision avoidance based on vision as the main cue even in cluttered inner city scenarios.

We currently incorporate the proposed real-time capable approach to our biologically motivated driver assistance system (described in [25]) to get both conjointly running online and in real-time on our prototype vehicle in inner city.

REFERENCES

[1] M. Ikegaya, N. Asanuma, S. Ishida, and S. Kondo, "Development of a lane following assistance system," in *Int. Symp. on Advanced Vehicle Control*, Nagoya, 1998.

[2] M. Schorn, U. Stahlin, A. Khanafer, and R. Isermann, "Nonlinear trajectory following control for automatic steering of a collision avoiding vehicle," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2006.

[3] T. Dang, S. Kammel, C. Duchow, B. Hummel, and C. Stiller, "Path planning for autonomous driving based on stereoscopic and monoscopic vision cues," in *IEEE Proceedings of the 2006 American Control Conference*, 2006, pp. 191–196.

[4] A. Broggi, "Robust real-time lane and road detection in critical shadow conditions," in *IEEE Proc. Int. Symp. on Computer Vision*, 1995.

[5] C. Rotaru, T. Graf, and J. Zhang, "Extracting road features from color images using a cognitive approach," in *IEEE Intelligent Vehicles Sysposium*, 2004.

[6] N. Soquet, D. Aubert, and N. Hautiere, "Road segmentation supervised by an extended vdisparity algorithm for autonomous navigation," in *IEEE Intelligent Vehicles Sysposium*, 2007.

[7] A. Gepperth, B. Mersch, C. Goerick, and J. Fritsch, "Color object recognition in real-world scenes," in *J. Marques de Sa et al. (Eds.): Artificial Neural Networks, 17th International Conference ICANN, Part II*, ser. Lecture Notes in computer science, J. de Sa, Ed. Springer Verlag Berlin Heidelberg New York, 2007, pp. 583–592.

[8] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *IEEE Intelligent Vehicles Sysposium*, 6 2007.

[11] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, 1992.

[12] U. Franke, H. Loose, and C. Knoeppel, "Lane recognition on country roads," *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 99–104, 13-15 June 2007.

[13] V. Willert, M. Toussaint, J. Eggert, and E. Körner, "Uncertainty optimization for robust dynamic optical flow estimation," in *Proceedings of the 2007 International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2007.

[14] K. Konolige, "Small vision system: Hardware and implementation," in *Eighth International Symposium on Robotics Research*, 1997.

[15] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," pp. 1106–1112, 1997.

[16] J.Y.Bouguet, "Camera calibration toolbox for matlab," 2007, http://www.vision.caltech.edu/bouguetj.

[17] T. Marita, F. Oniga, S. Nedevschi, T. Graf, and R. Schmidt, "Camera calibration method for far range stereovision sensors used in vehicles," in *IEEE Intelligent Vehicles Sysposium*, 6 2007, pp. 356–363.

[18] X. Li, X. Yao, Y. Murphey, R. Karlsen, and G. Gerhart, "A real-time vehicle detection and tracking system in outdoor traffic scenes," in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[19] M. Cech, W. Niem, S. Abraham, and C. Stiller, "Dynamic ego-pose estimation for driver assistance in urban environments," in *IEEE Intelligent Vehicles Symposium*, 6 2004, pp. 43–48.

[20] T. Michalke, J. Fritsch, and C. Goerick, "Enhancing robustness of a saliency-based attention system for driver assistance," in *The 6th International Conference on Computer Vision Systems (ICVS), Santorini, Greece, 2008. Lecture Notes in Computer Science, Springer*, no. 5008, 2008, pp. 43–55.

[21] BenchmarkData, 2008, http://www.rtr.tu-darmstadt.de/~tmichalk/ITSC_TempIntegration/.

[22] P. Lombardi, M. Zanin, and S. Messelodi, "Unified stereovision for ground, road and obstacle detection," in *IEEE Intelligent Vehicles Sysposium*, 2005.

[23] V. Willert, J. Eggert, J. Adamy, and E. Koerner, "Non-gaussian velocity distributions integrated over space, time and scales," *IEEE Transactions on Systems, Man and Cybernetics B*, vol. 36, no. 3, pp. 482–493, 2006.

[24] A. Ceravola, F. Joublin, M. Dunn, J. Eggert, and C. Goerick, "Integrated research and development environment for real-time distributed embodied intelligent systems," in *Proc. Int. Conf. on Robots and Intelligent Systems*, 2006, pp. 1631–1637.

[25] T. Michalke, A. Gepperth, M. Schneider, J. Fritsch, and C. Goerick, "Towards a human-like vision system for resource-constrained intelligent cars," in *Int. Conf. on Computer Vision Systems*, Bielefeld, 2007.

[26] Intel, "Integrated Performance Primitives," 2006, http://www.intel.com/cd/software/products/asmo-na/eng/perflib/ipp/302910.htm.