Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Monocular Road Segmentation using Slow Feature Analysis

Tobias Kühnl, Franz Kummert, Jannik Fritsch

2011

Preprint:

This is an accepted article published in IEEE Intelligent Vehicles Symposium (IV). The final authenticated version is available online at: https://doi.org/[DOI not available]

Monocular Road Segmentation using Slow Feature Analysis

Tobias Kühnl, Franz Kummert and Jannik Fritsch

Abstract— **In this paper a novel approach for road detection with a monocular camera is introduced. We propose a two step approach, combining a patch-based segmentation with additional boundary detection. We use Slow Feature Analysis (SFA) which leads to improved appearance descriptors for road and non-road parts on patch level. From the slow features a low order feature set is formed which is used together with color and Walsh Hadamard texture features to train a patch-based GentleBoost classifier. This allows extracting areas from the image that correspond to the road with a certain confidence. Typically the border regions between road and non-road have the highest classification error rates, because the appearance is hard to distinguish on the patch level. Therefore we propose a post-processing step with a specialized classifier applied to the boundary region of the image to improve the segmentation results. In order to evaluate the quality of road segmentation we propose an application-based quality measurement applying an inverse perspective mapping on the image to obtain a Birds Eye View (BEV). The advantage of this approach is that the important distant parts and boundaries of the road in the real world, which are only a low fraction in the perspective image, can be assessed in this metric measure significantly better than on the pixel level. In addition, we estimate the driving corridor width and boundary error, because for Advanced Driver Assistant Systems (ADAS) metric information is needed. For all evaluations in different road and weather conditions, our system shows an improved performance of the two step approach compared to the basic segmentation.**

I. Introduction

In order to decrease the number of traffic accidents accompanied by an increase of driving comfort for future cars, the topic of road detection is of high interest for ADAS. Due to lack of generality, commercial ADAS are often limited to specific scenarios. For example, Lane Keeping Assistant Systems (LKAS) are restricted to highway situations with certain conditions, e.g. a low curvature of the lane. However, the robust recognition of arbitrary road in front of the ego-vehicle will be needed for future ADAS operating in more complex traffic situations, especially in inner-city. If there are no explicit road boundaries (e.g. curbstones / lane makers) detectable, e.g., because of parking cars on the side occluding them, current LKAS are not working.

Road detection is beneficial for, e.g., path planning and all kinds of object detection, because it creates knowledge about where the ego-vehicle will probably move to and where other road users, e.g. cars and pedestrians, will potentially appear. ADAS require high recognition rates which is especially demanding for vision-based detection because coping with changing visual appearance and illumination of the road surface is very challenging.

The novel segmentation approach introduced in this paper detects the road using a monocular camera. It is suitable for any type of road because we do not apply an explicit model for the road shape or its boundaries. Our offline-trained system is split into two major parts, the basic segmentation and additional boundary region detection. The basic segmentation uses texture and appearance features to represent road and non-road regions on the patch-level. We found out that the use of Slow Feature Analysis [1] is very efficient in obtaining class specific appearance descriptors for the task of patch-based road classification. The benefit of the proposed features for road detection is shown by training a classifier and evaluating the results on real-world video data. Typically the border regions between road and non-road have the highest classification error rates, because on the patch level they are hard to distinguish. Therefore we extend our system with a post-processing step, the boundary region detection. We train a specialized classifier in the extracted boundary region, taking additional features (compared to the basic segmentation) and increasing the complexity of the classifier, which significantly improves the segmentation results for distant road sections and the border between road and non-road. Afterwards we fuse the results of both system parts to obtain the final segmentation result.

We evaluate the system in two ways, on the pixel-level in a perspective image, to allow the comparison to other approaches, and additionally on an application-oriented metric representation based on the Birds Eye View (BEV) transformation. In contrast to a perspective image, in BEV the size of a road section does not depend on the distance from the ego-vehicle which allows far better evaluation. Therefore, image-based evaluations as pixel-based accuracy or quality measures do not significantly reflect the requirements for ADAS because the bigger part of the image is covered by nearby regions. Especially in the metric evaluation the proposed boundary region detection improves the quality significantly. Additionally, we perform a metric driving corridor estimation in order to assess the feasibility of detecting narrow road sections with a single camera.

II. Related Work

Vision-based road segmentation has been addressed in many papers in the last decade. Therefore only some are cited here.

T. Kühnl is with the Research Institute for Cognition and Robotics, Bielefeld University, Bielefeld, Germany TKuehnl@cor-lab.uni-bielefeld.de

F. Kummert is with the Faculty of Technology, Bielefeld University, Bielefeld, Germany Franz@techfak.uni-bielefeld.de

J. Fritsch and T. Kühnl are with the Honda Research Institute Europe GmbH, Offenbach am Main, Germany Jannik.Fritsch@honda-ri.de

A group of authors propose to use road boundary models for representing the road. Features for these models are extracted from longitudinal road structures like lane markings or road boundary obstacles (like curbstones or barriers) by visual processing. This is mainly based on color and edge appearance (see e.g. [2]) or 3D information from stereo processing (see e.g. [3]) or Structure From Motion (see e.g. [4]). From the extracted features the model parameters can be tracked using different road shape models (see e.g. [5]). It was shown that the range of these methods can be extended by fusing visual information with digital map data [2], [6]. However, especially for inner-city the applicability of these approaches is limited because of violated model assumptions (intersections, parked cars occluding curbstones).

Pixel based classifications, using Conditional Random Fields (CRF), can be used to identify multiple scene elements in the field of view, including the road surface [7], [8]. These currently popular approaches from the computer vision community are powerful, but the comparison of these rather holistic methods with dedicated road segmentation methods (e.g. [9]) shows that the concentration on the relevant road surface results in a better classification performance. The results from [10] show that the use of boosting for monocular image classification achieves a high pixel-based accuracy with a very low system complexity. This approach is similar to the basic segmentation of our system. However, we use different features and extend the system with an additional boundary detection to improve the application-based metric quality.

While our approach has to be trained, adaptive approaches aim at a higher robustness when encountering unseen situations like, e.g., different weather and lighting conditions [11]. This, however, requires adaptation criteria suitable for arbitrary traffic situations which are often hard to define. Alternatively, adaptation of the geometrical setup can be performed [9]. They propose a parameter optimization for a probabilistic model using a homography of stereo images. Assuming a planar world, road regions are classified using the offset of corresponding corner points mapped from the stereo images into the homography.

III. Slow Feature Analysis of Temporal Signals

Slow Feature Analysis (SFA) is a learning technique which enables to find useful and invariant representations by using unsupervised learning [1]. During the training the algorithm performs an optimization in order to obtain a static transformation from a highly varying multidimensional temporal input signal to a slowly-varying output signal. This concept is illustrated in Fig. 1. For vision-based tasks the rapidly changing sensory inputs, namely the pixel values, encode the behaviorally relevant visual information like class membership only indirectly. In our patch-based classification system the temporal signal corresponds to the change of pixel values x_i. The temporal change is generated by spatially shifting a patch over image-areas, belonging to one class and sampling the function value for each pixel. While therefore the pixel values change, they all belong to the same class.

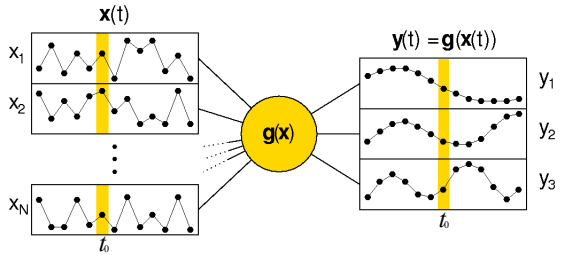

Fig. 1. Schematics of the optimization problem solved by slow feature analysis. Timestep t_0 marked in yellow, illustrating the instantaneous transformation.

In order to easily separate road and non-road input signals in feature space, we need a transformation that creates output signals with low variance from arbitrary input signals belonging to one class. This can be achieved with SFA because it creates a class specific representation for our type of input signals. Additionally it can be used for order reduction, because in general a specified number of slow features that are able to distinguish inputs from different classes can be found [12]. Mathematically spoken we search the quantity of functions $g_j(x)$ that is generating the slowest varying output functions $y_j(t)$ from a multidimensional input signal $x(t)$ (see Eq. (1)).

$$y_j(t) = g_j\left(x(t)\right) \tag{1}$$

Given Eq. (1) we can formulate an optimization problem: Finding the transfer function $g_j(x)$ that minimizes the temporal variance of the output signals $\Delta(y_j)$ (see Eq. (2)).

$$\Delta(y_j) = \left\langle \dot{y}_j^2 \right\rangle_t \tag{2}$$

We require uncorrelated output signals, having an equal variance and zero mean, which leads to the constraints in Eq. (3)-(5). Eq. (3) forces the output signals to be decorrelated, Eq. (4)-(5) exclude trivial solutions.

$$\forall i < j : \langle y_i \cdot y_j \rangle_t = 0 \tag{3}$$

$$\left\langle y_j^2 \right\rangle_t = 1 \tag{4}$$

$$\langle y_j \rangle_t = 0 \tag{5}$$

In Eq. (2)-(5) $\langle f \rangle_t := \int_{t_0}^{t_1} \frac{1}{t_1 - t_0} f(t)\,\mathrm{d}t$ means averaging the function f over time and with the temporal derivative of f being $\dot{f}$. A solution for the optimization problem can be found in [1].

IV. System

The system (see Fig. 2) consists of three parts: basic segmentation (I), boundary detection (II) and fusion. Input are RGB images, output is a confidence map that can be thresholded to extract a binary road segmentation.

A. Basic segmentation

The processing of the basic segmentation (blue marked part of Fig. 2) can be split in four major parts: The patch extraction, feature computation, classification and mapping to the image plane. The module computing the SFA-features and the classifier have to be trained offline once, afterwards the system can process input images with the learned parameters. Output of the basic segmentation is a confidence

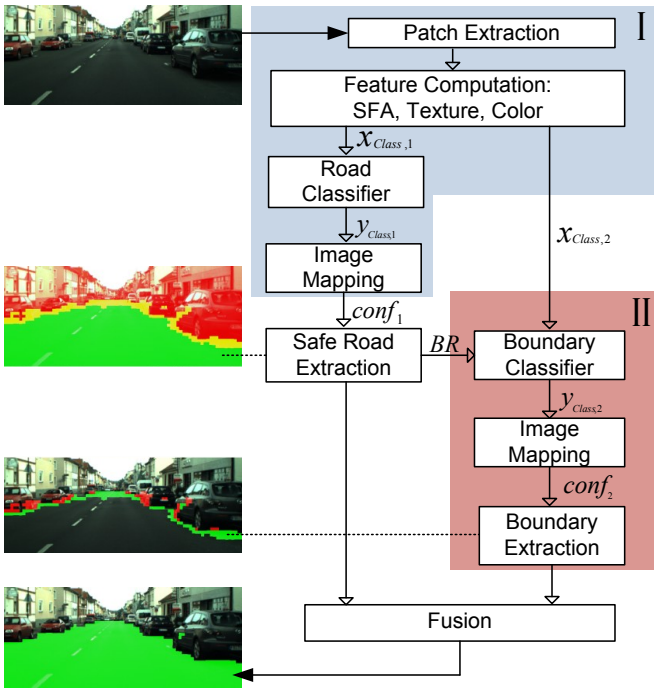

Fig. 2. System block diagram: Part I is the basic segmentation, part II is the boundary detection. The fusion module combines the safe road from part I and the expert decision from part II (binary).

map indicating for every pixel position whether it is likely to belong to the road.

Training of SFA: As mentioned in Sec. III, we extract patches and serialize the pixel values into signals needed for SFA training. We realize this spatial image sampling by using predefined constant paths which define how the point of patch-extraction moves over the image plane. There are two paths, one horizontal p_{hor} and one vertical path p_{ver}, as illustrated in Fig. 3. The advantage of using two paths is that the higher variability of the spatial input signal increases the likeliness of finding a useful transformation.

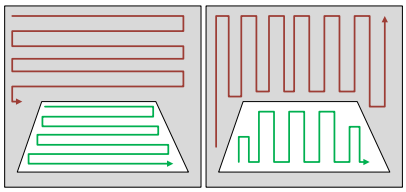

Fig. 3. Path for spatial patch sequence extraction for SFA training: on the left the horizontal and on the right the vertical path is illustrated. The paths are partitioned into road (green) and non-road (red) sections.

Given a patch $P_i = f(c_i, a_P)$, defined by its center $c_i = [c_{i,u}\,, c_{i,v}]$ and size $a_P = [a_{P,u}\,, a_{P,v}]$, we can sequentially extract patches by shifting the center c_i along a path $p_{\text{hor|ver}}$ with a constant step size s_p which results in a spatial signal $x_{\text{SFA}}(k_t)$. For the patch size a_P we used 21×21 pixels and a step size of $s_p = 10$, the spatial index k_t corresponds to t from Eq. (1). A signal $x_{\text{SFA}}(k_t)$ is a $d_k \times d_x$ matrix, where d_k describes the number of samples and $d_x = 21^2 \cdot 3$ the input dimension of an image patch. In order to minimize the temporal variance for each class, temporal signals corresponding to road $x_{\text{SFA,R}}(k_t)$ and non-road $x_{\text{SFA,NR}}(k_t)$ are extracted, as it is illustrated in Fig. 3. With ground truth information, given by a binary matrix (road = true), the assignment for every patch along the path can be found by thresholding the number of patch-pixels belonging to the road class. Every patch containing more than 50% of true pixels in the ground truth is interpreted as belonging to the road. Applying this for every training image we are able to train a model (linear SFA), defined by the transfer function $g(x(k_t))$ (cf. Sec. III), by presenting the system a certain number of signals $x_{\text{SFA,R}}(k_t)$ and $x_{\text{SFA,NR}}(k_t)$, using the SFA-TK Toolbox [13].

With the trained transformation, we are now able to extract a slowly varying output signal $y_{\text{SFA}}(k_t)$ for every input image patch P_i. The signal $y_{\text{SFA}}(k_t)$ has the dimension n_{slow} ($n_{\text{slow}} \leq d_x$) which is the number of slow features. Here we used $n_{\text{slow}} = 3$ which is a huge reduction of the feature space, compared to the input dimension d_x.

In principle it should be sufficient to use the first slowest feature to separate the slowly varying road from the rapidly varying non-road (cf. [12]), but due to noise and additional influences like changes in illumination and appearance (road markings, different surface colors), the classification results improve for multiple slow features.

Training of GentleBoost: We use boosting for patch-classification, as it has been shown to be very successful in feature selection and classification [10]. Here the GentleBoost classification method [14] is used, taking a 27×1 dimensional feature vector $x_{\text{Class},1}$, containing 3 slow features $y_{\text{SFA},3}(k_t)$, retinal position $[u, v]$, a set of 16 Walsh Hadamard texture features [15] and 6 simple color features (mean and variance of the RGB values in a patch). The algorithm generates a sequentially weighted set of weak classifiers that build a strong classifier in combination. In every iteration of the procedure, according to the current distribution of weights on the input signal, the method attempts to find an optimal classifier. We set up the weak classifiers with decision stumps (1 tree split) and a maximum of 100 boosting iterations to get a classifier with low complexity. After training is finished, the classifier generates a confidence value $y_{\text{Class},1}$ for a given feature vector, indicating whether the corresponding patch center position c_i is likely belonging to the road class or not.

Processing phase: In the processing phase of the system, patches are extracted along a horizontal path over the complete image with a step size of $s_p = 7$ px. We reduced the step size, compared to the training, in order to achieve finer graduation in the results. Similar to the training, we compute a feature vector $x_{\text{Class},1}$ for every patch and obtain patch-based confidences $y_{\text{Class},1}$ with the trained road classifier. Based on patch center position c_i we map the confidences $y_{\text{Class},1}$ onto the perspective image plane and obtain the image-based confidence map $conf_1(u, v)$ by applying linear interpolation. In $conf_1(u, v)$ a threshold th_1 that maximizes the average quality (cf. Sec. V) over all frames can be found. A visual example of this basic segmentation result can be seen in Fig. 4. On the system level $conf_1(u, v)$ is used for safe road estimation and boundary region extraction in order to use it for the boundary detection and the fusion process.

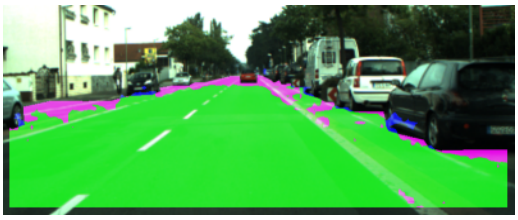

Fig. 4. Segmentation result in inner-city / overcast (frame 678): Reaching a quality $Q_I = 92.1\%$ ($th_1 = 0$), illustrated is detected (green), missed road (magenta) and blue the wrongly detected road using ground-truth information.

B. Boundary detection

As we will see in the evaluation, a high percentage of the road can be detected with the basic segmentation (Sec. IV-A). Especially untextured gray regions and lane-markings have been learned to belong to the road and can be easily classified. However, the real challenge is to recognize important parts in the image like distant parts and regions close to the road-boundary. These cover only a low percentage of the overall image, therefore we propose to use a second processing step that is focusing on the low confidential regions from the basic segmentation to improve the classification results. The task is more challenging compared to classification on the whole image, because on the patch level the samples for road and non-road have a similar appearance. Therefore we increase the number of input features and the classifier complexity for this postprocessing step.

Preprocessing: Before starting with the actual boundary detection a preprocessing step is applied on the confidence map $conf_1(u,v)$ (see Sec. (IV-A)), in order to extract the boundary region BR of the image. A pixel position $(u,v)^T$ belongs to BR if its confidence value $conf_1(u,v)$ satisfies $conf_{\text{low}} \leq conf_1(u,v) \leq conf_{\text{high}}$. The value $conf_{\text{low}}$ is obtained during training by finding the region in the confidence map $conf_1(u,v)$ that implies a false-positive rate FPR we want to tolerate. Because we allow only a very low rate of non-road pixels to be falsely classified ($FPR < 1\%$), we name the resulting binary segment safe road R_{SR}. The R_{SR} is anyway part of the final road segment because it is used in the binary fusion. In the same way $conf_{\text{high}}$ is obtained, finding the threshold for the safe non-road region R_{SN} with a false-negative rate $FNR < 1\%$. The amount of the high confident classification area size $S_{I,C}$ is described by the aggregated area of R_{SR} and R_{SN} divided by the overall image area. For example, on urban roads (overcast) the mean of $S_{I,C}$ is 85%. This illustrates that only a low fraction of the perspective image is hard to classify.

Training and processing: We take the same patch parameter setup (patch, step size) as in Sec. IV-A and skipped retraining the SFA-module. However, we build a new feature vector $x_{\text{Class},2}$ with 20 slow features, 256 Walsh Hadamard features and 6 color features. For training the boundary classifier we proceed like in Sec. IV-A. As mentioned we increase the complexity of the classifier by taking 4 tree splits and 400 boosting iterations. Applying the image mapping (same like in Sec. IV-A) to the classification result $y_{\text{Class},2}$, we obtain the confidence map of the boundary detection $conf_2(u,v)$.

Fusion: With a simple binary fusion method the results from basic segmentation and boundary detection are combined. The intention is to combine the safe road R_{SR} with the expert decision of the boundary detection. The binary fusion to obtain the confidence map $conf_{\text{fus}}(u,v)$ is given in Eq. 6.

$$conf_{\text{fus}}(u,v) = \begin{cases} 1, & \text{if } (u,v) \subseteq R_{SR} \\ -1, & \text{if } (u,v) \subseteq R_{SN} \\ conf_2(u,v), & \text{else} \end{cases} \quad (6)$$

A decision if a pixel (u,v) belongs to the road can be made by applying a threshold th_{fus} (determined during training) on $conf_{\text{fus}}(u,v)$. An exemplary result of the system performance is shown in Fig. 5.

Fig. 5. Segmentation result in inner-city / overcast (frame 678). The upper image shows the distinct parts of the boundary region extraction: safe road (green), safe non-road (red) and the boundary region (yellow). In the lower image the segmentation result of the boundary detection is illustrated. The fused result gives a quality $Q_I = 94,4\%$ ($th_{fus} = 0$).

V. Evaluation

For the proposed system we use RGB images with a resolution of 800×330 pixels from our video streams[1] manually annotated with 1Hz (recorded with 20 Hz) and a total stream length of about 25.5 minutes (1531 annotated frames). Corresponding to road category (highway, rural road, urban) and weather condition, these streams can be separated into 7 datasets (see Tab. I). We split each dataset into training and testing part by using blocking of 4 seconds (blocks for test / train alternate). As ADAS need to handle varying conditions the generalization is an important issue for our trained system. Therefore dedicated and general training sets are used. Dedicated means training and testing dataset of one specific condition while for general training multiple datasets were merged and training included different conditions. Two general training sets are used: first on different road types (highway, rural road and urban) under a specific weather condition (overcast) and second on one specific road type (urban) under multiple day weather conditions (overcast, sunny, rain, snow).

For evaluation (see next subsections) several criteria from related research [16] are used. The number of negative

[1]The used videos and annotations can be obtained by sending an e-mail request to *hri-road-traffic@honda-ri.de*. Annotations include road, vehicles, traffic signs and generic obstacles.

TABLE I

DATASET INFO

Name	Weather	Short	# Frames
highway	overcast	H/O	94
rural road	overcast	RR/O	351
urban	overcast	U/O	200
urban	sunny	U/SU	210
urban	rain	U/R	260
urban	snow	U/SN	220
urban	night	U/N	196

N (non-road) and positive P (road) pixels and the false positives FP and false negatives FN are obtained for every single frame i. We use the accuracy A, given in Eq. 7, in order to allow comparison of the system performance with other approaches (cf. [9], [10]). In addition we use the quality measure (cf. [11]), given in Eq. 8, as it is a criteria weighting errors much harder, compared to the accuracy.

$$A_{I,M} = 1 - \frac{\sum_{i=1}^{n}(FP+FN)}{\sum_{i=1}^{n}(P+N)} \cdot 100\% \quad (7)$$

$$Q_{I,M} = \frac{\sum_{i=1}^{n} TP}{\sum_{i=1}^{n}(TP+FN+FP)} \cdot 100\% \quad (8)$$

We apply this measurement for the perspective image (index = $_I$) and the metric (index = $_M$) representation, which is a more relevant application-oriented performance measurement. The problem of the right evaluation criteria for road detection methods, as it has been also discussed in [16], becomes visible if we have a look at a metric representation of the visual scene. Therefore we use inverse perspective mapping, to obtain the so called Birds-Eye-View (BEV) [17]. Under the presumption of a flat world ($y = 0$) and known extrinsic camera parameters, we can map every pixel at image position $[u, v]$ in a cell with the coordinates $[x, z]$ in the BEV (resolution is $10cm \times 10cm$). Applying this to the confidence maps of the basic segmentation and the fused results, we can evaluate the metric accuracy A_M and quality Q_M of each system part and the quality gain ΔQ_M (used for evaluating the performance gain of the boundary detection).

In general the results of the metric evaluation are always lower than those of the perspective image (see Sec. V-B). The reason can be seen by comparing Fig. 4 and Fig. 6 (left): the distant regions only cover a very low percentage of the whole image, but a large area in the BEV, additionally the borders have a higher impact in the BEV. This is a very important issue for application oriented systems, because for an ADAS warning a driver about oncoming narrow street sections, the width of the road needs to be measured in a distance of about 30-50m.

A. Basic segmentation evaluation

To asses the performance of the basic segmentation an evaluation on the confident image region, described by its size $S_{I,C}$ (cf. Sec. IV-B), is carried out (see Tab. II). The confident region size $S_{I,C}$ can be seen as a degree, how much the basic segmentation contributes to the final segmentation result. Note, that for this evaluation the accuracies $A_{\#,C}$ and qualities $Q_{\#,C}$ are only measured on the confident part (appending a subindex $_C$) and not on the whole image / metric space. In Tab. II we measure a quality $Q_{I,C}$ of at least 92.5%, where the confident region $S_{I,C}$ covers at least 75% of the image, for all datasets (dedicated and general training).

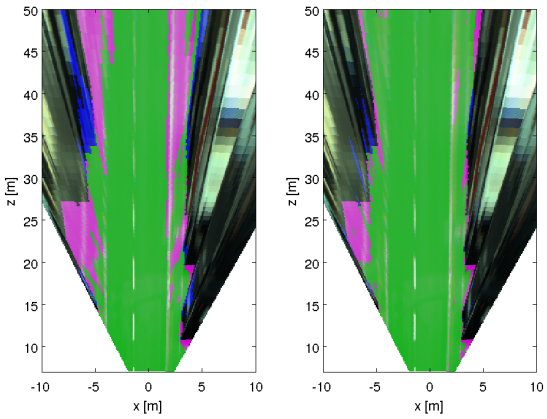

Fig. 6. BEV (frame 678) for the basic segmentation result (left) and fused result (right). Green is the detected, magenta the missed and blue the wrongly detected road.

TABLE II

BASIC SEGMENTATION

Test	$A_{I,C}$ [%]	$Q_{I,C}$ [%]	$S_{I,C}$ [%]	$A_{M,C}$ [%]	$Q_{M,C}$ [%]	$S_{M,C}$ [%]
Dedicated training						
H/O	99.52	99.07	97.84	98.62	97.74	91.78
RR/O	98.72	97.32	90.50	95.26	91.58	69.84
U/O	98.05	95.68	85.61	96.05	89.71	56.01
U/SU	98.37	96.47	88.41	97.72	94.51	62.38
U/R	97.81	94.39	77.10	94.42	71.45	49.91
U/SN	97.70	94.62	80.06	94.32	82.79	43.43
U/N	97.48	93.63	77.04	92.42	72.31	39.42
General training (overcast, different road types)						
H/O	99.75	99.50	92.66	99.45	99.04	65.50
RR/O	98.81	97.42	87.65	95.24	90.13	60.42
U/O	97.42	94.76	85.47	94.44	90.63	52.19
General training (urban, different day weather conditions)						
U/O	98.14	95.83	81.74	96.77	90.49	42.60
U/SU	98.52	96.52	81.94	96.00	85.59	50.60
U/R	97.01	92.64	76.04	94.13	76.64	36.30
U/SN	97.40	93.93	78.50	95.70	86.32	36.95

Performance differences on the particular datasets are visible from the variations in the individual region size $S_{M,C}$ and quality $Q_{M,C}$. For the urban datasets one can trace this back to not uniformly shaped roads due to higher traffic, parked cars and entry sections, compared to highway and rural roads. The lower performance for rain, snow, and night is caused by the challenging appearance and texture of the road in these datasets. If we compare the general training with dedicated training results, a decrease of $S_{I,C}$ and $S_{M,C}$ becomes visible. This results in the boundary detection to operating on a larger image area.

B. Complete system evaluation

To assess the performance gain of the system extension with the boundary detection, we evaluate the quality gain ΔQ. This quality difference can be computed by subtracting the quality of the basic segmentation, computed on the complete image, from the combined system quality. Although we measure only a minor increase of ΔQ_{I} on the pixel level, this results in a significant increase of ΔQ_{M} in the metric representation (especially on rural roads and urban datasets). The offset is rather small for highway (0.45 %) because the boundary detection is only applied on 2.2 % of the perspective image (cf. Tab. II: $S_{I,C}$ = 97,8%), while for all other test conditions (dedicated and general training) the system shows a larger performance increase for the extended system due to the larger boundary regions.

This offset is also visible in Fig. 6 (right), in the distant sectors and the border regions the segmentation is improved. For example, when using the system on the urban / overcast dataset, with training on different weather conditions, the boundary detection improves the quality with 12 %. As we see that the system performance only slightly decreases for general training, we infer that the system can cope with different appearance and lighting conditions in the training data and learns a representation that robustly separates road and non-road in different scenarios.

TABLE III
COMPLETE SYSTEM EVALUATION

Test	A_I [%]	Q_I [%]	ΔQ_I [%]	A_M [%]	Q_M [%]	ΔQ_M [%]
Dedicated training						
H/O	98.95	98.04	0.29	96.09	93.66	0.45
RR/O	97.46	94.85	2.83	91.69	85.44	6.71
U/O	95.85	91.42	4.64	89.90	78.44	9.89
U/SU	96.10	91.85	2.22	91.13	80.73	4.18
U/R	93.34	85.44	4.99	85.81	63.99	14.51
U/SN	93.09	85.20	2.37	86.64	67.38	7.72
U/N	93.27	85.51	4.43	82.54	62.95	12.53
General training (overcast, different road types)						
H/O	98.94	98.00	1.60	96.52	94.30	2.59
RR/O	97.28	94.44	3.49	90.79	83.62	7.98
U/O	94.82	89.61	3.19	88.25	76.27	7.77
General training (urban, different day weather conditions)						
U/O	95.29	90.34	4.18	89.05	76.85	11.90
U/SU	95.35	90.32	3.58	89.01	76.30	9.18
U/R	91.79	82.60	2.75	80.73	56.15	7.60
U/SN	91.91	83.04	0.63	84.24	63.23	3.47

The direct comparison to related approaches is not possible, because the image datasets and the mounting positions of the cameras are different. However, we see that the obtained results are in a similar range of values like, e.g., those of Guo et al. [9], with the advantage of our approach using only a single camera.

C. Driving Corridor Estimation on Urban Roads

In order to assess the feasibility to detect narrow road sections, the driving corridor width $w(z)$ and the position of the right driving corridor boundary $b(z)$ from the detected road is estimated. Starting at 8m from the rear axis of the ego-vehicle the corridor is sampled at discrete distances with Δ_z = 2m. A basic outline is given in Fig. 8. To assess the system performance on driving corridor estimation we

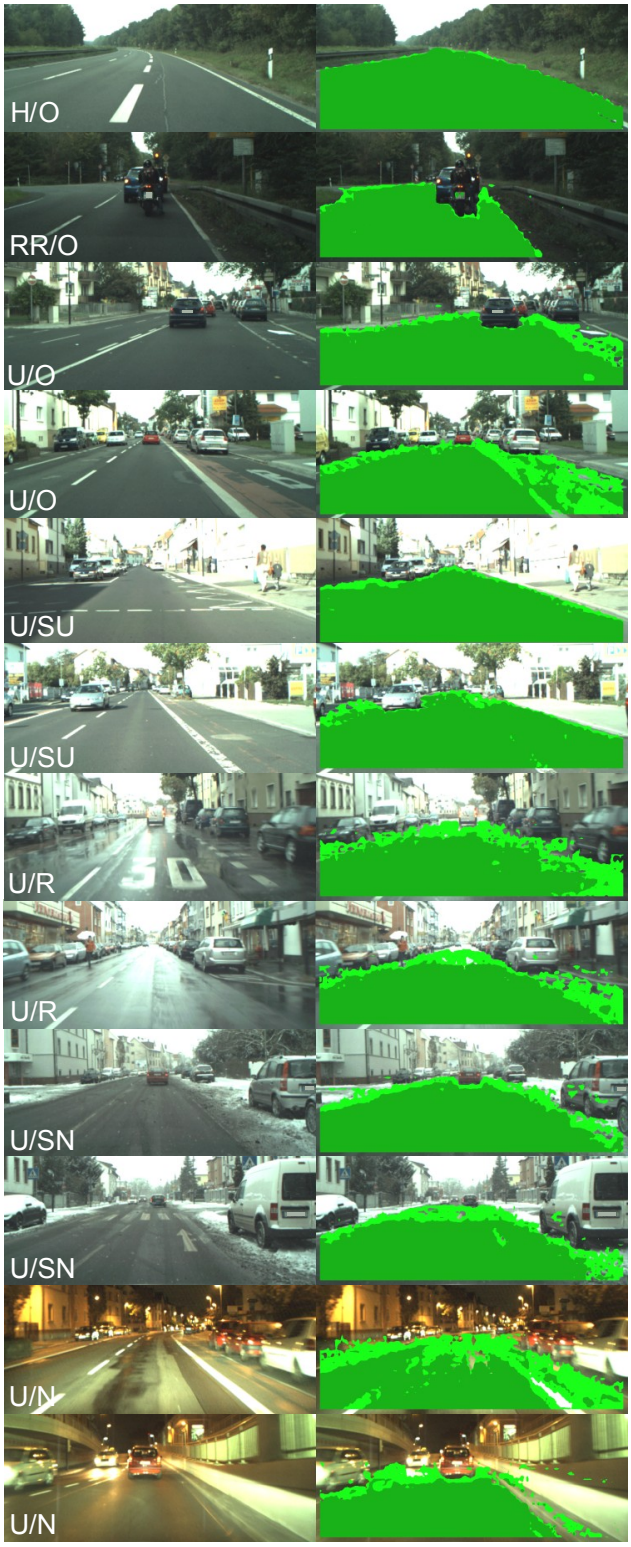

Fig. 7. Example images of all datasets (see Tab. I). Dark green shows the detected road from the basic segmentation, bright green is the road detected by the boundary detection.

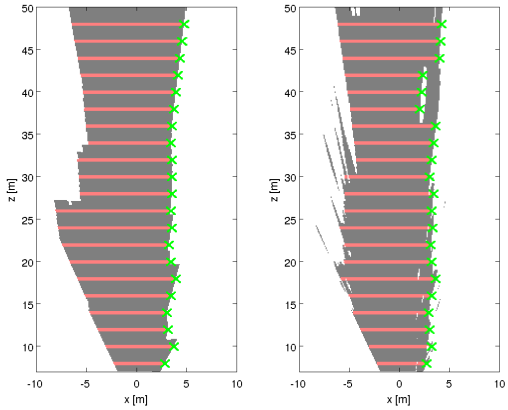

Fig. 8. Estimation of the driving corridor in BEV (frame 678) of the ground truth polygon (left) and the extracted road region (right). Extracted width and the right boundary are illustrated with red lines and green crosses.

took a subset of the urban / overcast dataset with 53 frames of roughly straight road. For the comparison with ground truth we introduce two error measurements: The standard deviation of the error of the corridor width σ_{w} and the standard deviation of the position error of the estimated boundary σ_{b}. From these results, given in Fig. 9, we can see that the performance of our system extension significantly reduces the application-based error, especially for the width estimation of distant road sections.

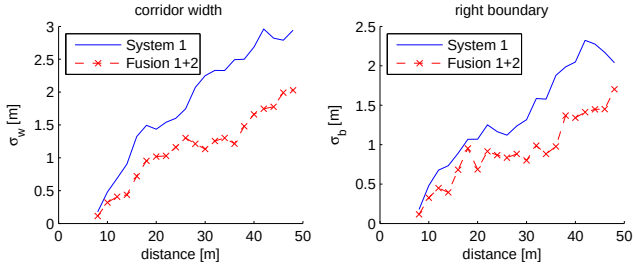

Fig. 9. Evaluation of the driving corridor. The standard deviation of the error of the corridor width (left) and the position of the right boundary of the corridor (right) are plotted over distance to the ego vehicle.

VI. Conclusion and Future Works

We proposed a novel two step approach for road-area segmentation. Tests with our challenging real world video data have shown that our segmentation approach can cope with arbitrary roads leading to comparable results on the pixel-level as state-of-the-art approaches. This is achieved with a single camera and no temporal integration. In addition we propose the assessment in a metric representation for automotive applications. We have shown that high accuracies on the perspective image do not guarantee high accuracies for application based metric measurements, like, e.g., a driving corridor measurement. The system extension with the boundary detection shows significant improvements on the metric quality which is highly relevant for constructing spatial representations [18]. The processing takes approximately 1 second per frame on a single core of a 2.7 GHz Intel Xeon CPU (Clovertown). The system, working in Matlab, is basically real-time capable, because it can be highly parallelized due to the patch-based system progression. In the future we will improve the system performance by enabling the system to automatically adapt to different weather and lighting conditions through selecting different classifiers.

VII. Acknowledgments

The authors gratefully acknowledge the support of Mathias Franzius for training the SFA and for the fruitful discussions and review comments.

References

[1] L. Wiskott and T. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation*, vol. 14, no. 4, pp. 715–770, 2002.

[2] C. Gackstatter, P. Heinemann, S. Thomas, B. Rosenhahn, and G. Klinker, "Fusion of clothoid segments for a more accurate and updated prediction of the road geometry," in *Proc. IEEE Intelligent Transportation Systems Conf.*, 2010, pp. 1691–1696.

[3] J. Siegemund, D. Pfeiffer, U. Franke, and W. Förstner, "Curb reconstruction using conditional random fields," in *Proc. IEEE Intelligent Vehicles Symp.*, 2010, pp. 203–210.

[4] M. Darms, M. Komar, and S. Lueke, "Map based road boundary estimation," in *Proc. IEEE Intelligent Vehicles Symp.*, 2010, pp. 609–614.

[5] M. Konrad, M. Szczot, and K. Dietmayer, "Road course estimation in occupancy grids," in *Proc. IEEE Intelligent Vehicles Symp.*, 2010, pp. 412–417.

[6] H. Weigel, H. Cramer, G. Wanielik, A. Polychronopoulos, and A. Saroldi, "Accurate road geometry estimation for a safe speed application," in *Proc. IEEE Intelligent Vehicles Symp.*, 2006, pp. 516–521.

[7] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. British Machine Vision Conference (BMVC)*, 2009.

[8] C. Wojek and B. Schiele, "A dynamic CRF model for joint labeling of object and scene classes," in *European Conference on Computer Vision (ECCV)*, vol. 5305, 2008, pp. 733–747.

[9] C. Guo, S. Mita, and D. McAllester, "Drivable road region detection using homography estimation and efficient belief propagation with coordinate descent optimization," in *Proc. IEEE Intelligent Vehicles Symp.*, 2009, pp. 317–323.

[10] Y. Sha, X. Yu, and G. Zhang, "A feature selection algorithm based on boosting for road detection," in *Proc. Conf. Fuzzy Systems and Knowledge Discovery FSKD*, vol. 2, 2008, pp. 257–261.

[11] T. Michalke, R. Kastner, M. Herbert, J. Fritsch, and C. Goerick, "Adaptive multi-cue fusion for robust detection of unmarked inner-city streets," in *Proc. IEEE Intelligent Vehicles Symp.*, 2009, pp. 1–8.

[12] M. Franzius, N. Wilbert, and L. Wiskott, "Invariant object recognition with slow feature analysis." in *Proc. Conf. on Artificial Neural Networks (ICANN)*, ser. Lecture Notes in Computer Science, vol. 5163. Springer, 2008, pp. 961–970.

[13] P. Berkes, "SFA-TK: Slow feature analysis toolkit for matlab (v.1.0.1)," 2003. [Online]. Available: http://itb.biologie.hu-berlin.de/berkes/software/sfa-tk/sfa-tk.shtml

[14] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[15] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 689–696.

[16] J. M. Alvarez and A. Lopez, "Novel index for objective evaluation of road detection algorithms," in *Proc. IEEE Conf. Intelligent Transportation Systems*, 2008, pp. 815–820.

[17] H. A. Mallot, H. H. Bulthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological Cybernetics*, vol. 64, pp. 177–185, 1991.

[18] R. Kastner, T. Michalke, J. Fritsch, and C. Goerick, "Towards a task dependent representation generation for scene analysis," in *Proc. IEEE Intelligent Vehicles Symp.*, 2010, pp. 731–737.