

Detecting Symbols on Road Surface for Mapping and Localization using OCR

Markus Schreiber, Fabian Poggenhans and Christoph Stiller

Mobile Perception Systems
FZI Research Center for Information Technology
76131 Karlsruhe, Germany
{schreiber,poggenhans}@fzi.de

Institute of Measurement and Control Systems
Karlsruhe Institute of Technology
76131 Karlsruhe, Germany
stiller@kit.com

Abstract—In this paper, we present a system to detect symbols on roads (e.g. arrows, speed limits, bus lanes and other pictograms) with a common monoscopic or stereoscopic camera system. No manual labeling of images is necessary since the exact definitions of the symbols in the legal instructions for road paintings are used. With those vector graphics an Optical Character Recognition (OCR) System is trained. If only a monoscopic camera is used, the vanishing point is estimated and an inverse perspective transformation is applied to obtain a distortion free top-view. In case of the stereoscopic camera setup, the 3D reconstruction is projected to a ground plane. TESSERACT, a common OCR system is used to classify the symbols. If odometry or position information is available, a spatial filtering and mapping is possible. The obtained information can be used on one side to improve localization, on the other side to provide further information for planning or generation of planning maps.

I. INTRODUCTION

For both driver assistance systems and autonomous driving, detection of lane markings plays a major role. As early as in the 80s, first approaches for autonomous driving used a camera to detect the lane markings and thereby the course of the road. Today, most upper class vehicles have a camera based system on board to detect lane markings and warn the driver when leaving the lane. Those approaches usually only detect boundary lines and ignore other road surface paintings.

Recent systems for autonomous driving highly rely on accurate maps. These maps usually fulfill two tasks: Providing information for localization and providing static information for path planning.

For localization, static landmarks are stored in the map and re-detected while driving. One approach is to compute generic feature descriptors for interesting points. With the position of the feature and the feature description itself, a relative pose between mapping pose and localization pose can be estimated (e.g. [1]). While this leads to good results, especially in urban areas, the necessary disk space for large maps must be considered.

Another approach is to extract more specific information such as lane markings (e.g. [2]). A map with all lane markings can be created based on any kind of high resolution birds-eye-view images of the desired road. In the localization step, lane markings are detected in the image and the result is compared to the previously generated map. Since no appearance based information is stored in the map, an update

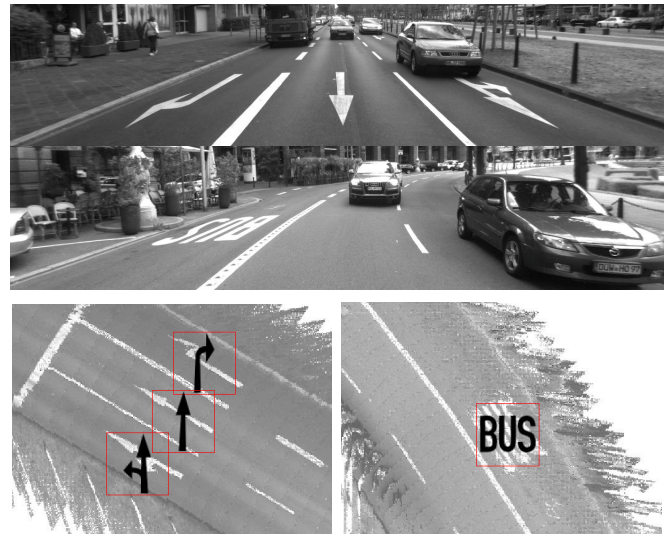


Fig. 1. Example images of the backward facing camera in the vehicle (top) and the results of the proposed algorithm (bottom). For better evaluation, the results are shown on a separately generated top view map.

of the map is not necessary until the road itself changes, e.g. due to road works. While lateral positioning is very accurate, the uncertainty in longitudinal direction is very high. This can be improved by detecting non-lateral lane markings such as arrows and stop lines.

Another important task of maps in autonomous driving is providing static information for trajectory planning (e.g. [3]). This can be drivable areas, position of traffic lights and corresponding stop lines, speed limits and information about connected driving corridors on large junctions. In contrast to the localization maps, which can be generated partially or completely automatically, for planning information maps a high manual effort is necessary. However, especially speed limits and lane directions are often painted on the road and hence seen by cameras.

Detecting the information painted on the road therefore would lead to further localization improvements, but also could serve as basis for planning map generation. Hence, our proposed system uses the knowledge of legal road marking definitions to detect them on road surface with an OCR system.

II. RELATED WORK

Road and lane marking detection is one of the oldest applications in camera based driver assistance systems. In 1990 the European research project PROMETHEUS used intensity and gradient information to detect lane markings in the image [4]. The development continued over the next years with different approaches, e.g. [5], [6], [7].

Today's research focuses on road geometry estimation in regions that do not provide clearly visible markings. Franke et al. [8] used particle filter to predict the road course on rural roads. Other approaches aim on the detection of curb stones [9], which usually delimit the drivable area in urban areas.

The goal of all those approaches is to detect the course of the road rather than to detect the lane markings as dedicated objects. A first approach to detect the markings itself is described in [10] with the goal to detect corrupted marking segments. A stereo camera system was used to generate a birds-eye-view on which a morphological filter was applied to remove everything except the arrow tips. With a neural network the frequency transformed images were analyzed. Nan et al. [11] also transformed an image in birds-eye-view perspective and used Haar wavlets to detect symbols similar to face detection algorithms. In [12] a system was proposed which detects lane markings in a given rectangular region and compares the binarized pattern to stored templates. Fourcher et al. [13] used the difference between a median filtered image and the original to extract lane marking borders. The result was compared to recorded templates with both, Hamming distance and histograms. Compared to [11] and [12] the system is more invariant to changes in lightning conditions, however, it showed problems with wide road markings. Another approach was recently proposed by Wu et al. [14], who used FAST corner descriptors and template matching to detect symbols on the road. Hence, manually labeled training images are necessary.

In contrast to most of the above applications Optical Character Recognition algorithms are designed to classify specific symbols. However, usually the input images are less perspectively distorted and the background is less noisy compared to cameras mounted on vehicles. One common library is TESSERACT [15]. It was developed by Hewlett-Packard in 1984-1994. Since 2005, the project is open source and is maintained by Google. They use TESSERACT for their Google Books project [16]. TESSERACT is available under Apache-License on code.google.com/p/tesseract-ocr/.

Posner et al. [17] applied OCR to natural scene images to detect visible text with the goal to improve scene interpretation. Also a system to detect text on road signs using OCR was proposed by Wu et al. [18]. However, both applications of OCR in robotic environments are restricted to detect common characters and human readable text.

We propose an algorithm that combines the knowledge of the definitions of allowed symbols with common OCR software to detect and classify road surface markings. There-

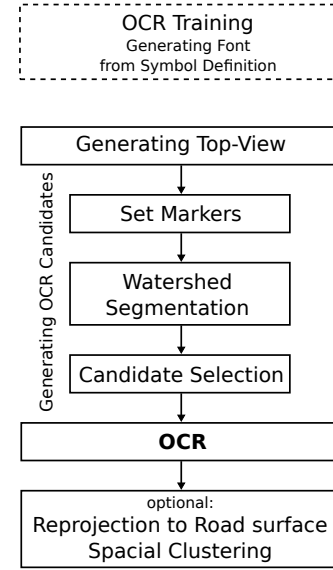


Fig. 2. Overview of the detection algorithm. OCR Training has to be done only once. Top-views are created from monoscopic or stereoscopic images, before candidates are generated and OCR is applied.

fore, the image is transformed to birds-eye-view, potential markings are extracted and OCR is applied. As final result we obtain a map with all symbols on the road.

III. ALGORITHM

The complete proposed algorithm consists of two main parts. In a preliminary step, the desired symbols must be transferred to a OCR-compatible format. In our case we simply generate a font from vector graphics of the desired symbols once.

The second part is the online detection algorithm, which is applied on each image. It can be divided into several further steps (see Fig 2). First, the images are transformed to top-view images. This can be done either by using stereo information and projecting the scene to a ground plane, or by estimating the vanishing point to perform an inverse perspective transform, if no stereo setup is available.

In the transformed image, markers on the road and on the road markings are generated initially. Afterwards, the image is binarized with a marker based watershed algorithm. Single elements are grouped for further processing and candidates for OCR are selected. OCR is applied by using TESSERACT-Library. In an optional last step, if odometry information is available, the detection of symbols in consecutive images can be integrated to one single optimized detection.

In the following sections we explain the main parts of the algorithm in detail.

A. Training of the OCR system

To prepare the OCR system, a font with the desired symbols is generated. The font consists in our case of all letters, symbols and arrows allowed on German roads, which is defined in [19]. However, any other symbol definitions, e.g. from other countries, can also be used. To increase

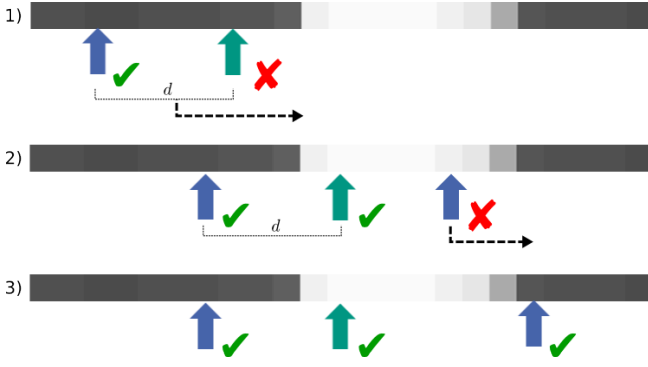


Fig. 6. Markers used for watershed algorithm are generated automatically. Here the process of setting a set of markers in an image row is depicted. Blue markers are outside, green markers inside the road markings. Each line is checked from left to right if a position is found where all conditions are fulfilled.

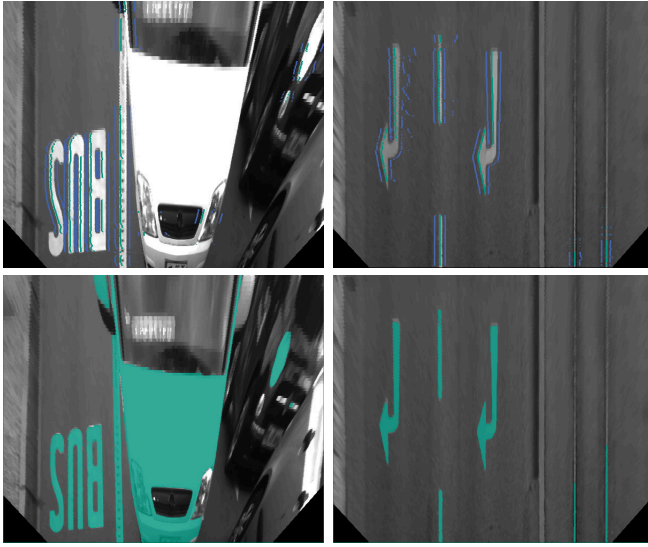


Fig. 7. Generated markers (top) and corresponding watershed Segmentation (bottom) on monoscopic top-view images.

is used to search first for candidates for left and middle markers with a given distance d . For each candidate pair the third marker is searched in the residual row. The process is shown in Fig. 6 while Fig. 7 (top) depicts the result of a complete image.

Afterwards, a common watershed algorithm [22] [23] is applied. The result can be seen in Fig. 7 (bottom). To avoid small gaps in the segmentation result, morphologic closing is applied additionally.

D. Candidate Selection

While the markings are usually clearly visible in the complete image, single candidates for OCR must be identified by grouping segmented parts that belong together (e.g. two digit numbers for speed limits). Moreover, the number of candidates for OCR can be limited by further assumptions. Helpful features for segmented parts are

- area of the bounding box
- aspect ratio of the bounding box
- length of the contour line.

Elements that are too small or large are rejected immediately. In addition the lane markings on the side of the road are detected based on their aspect ratio.

To group segments a Support-Vector-Machine is trained with common symbols consisting of more than one element, e.g. “BUS”, “30”, “50”, “70”, or symbol for reduction of the road width. Features are

- Distance d_u of bounding boxes in u -direction ($d_u = 0$ if bonding boxes touch)
- Distance d_v of bounding boxes in v -direction ($d_v = 0$ if elements begin in the same image row)
- Ratio of widths $r_w \geq 1$
- Ratio of heights $r_h \geq 1$.

The single parts of a group are still contained for further analysis in case of detection of the whole group was not successful.

E. Optical Character Recognition (OCR)

For OCR the common software TESSERACT is used. TESSERACT is not limited to Latin fonts, but complex characters, e.g. in Arabic or Asian scripts are possible as well. While there is no graphical user interface provided, TESSERACT can be directly accessed in C++ code. Furthermore, the software is able to segment a page of text into lines, single words and single characters. However, since road markings usually not occur in multiple lines of text and the segmentation is done separately (see Sec. III-C), those functions are not used in our algorithm.

According to [15] the following steps are performed within TESSERACT (see Fig. 8):

- 1) **Edge Detection** to obtain a closed outline.
- 2) **Approximation as polygon** to reduce amount of data.
- 3) **Feature Extraction** which are line segments describing the contour.
- 4) **Static classification** with all template characters.
- 5) **Adaptive classification** with the help of well recognized characters in the previous step.

TESSERACT returns the symbol with the highest probability for each candidate and the probability itself. The coordinates of the detected symbol are estimated by calculation the centroid of the input candidate segment. If the reduced training set is used for single symbol detection and the included character “I” is returned, the detection is automatically classified as dashed line segment and thus rejected. Finally, the image coordinates of the detected symbol can be converted into the vehicle coordinate system, to obtain relative metric pose information. If odometry and global reference is available, the symbol positions can be transformed to world coordinate system. Usually, in consecutive images a symbol is detected multiple times. Therefore, it is possible to cluster the results. All detections of the same symbol within a certain radius are merged to one final detection based on their matching possibility given by TESSERACT. In Fig. 9 an example with multiple detections is depicted on an accumulated top-view map. Blue dots indicate detections in single images, yellow and green dots the final decision.

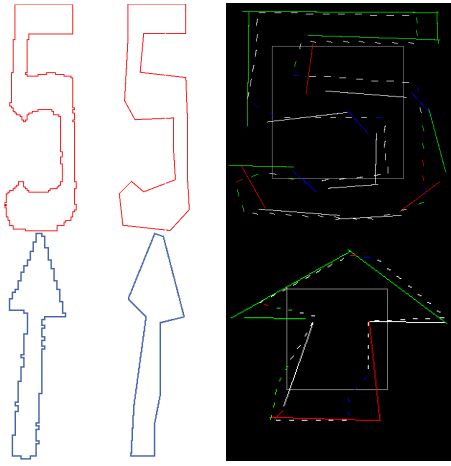


Fig. 8. Example of TESSERACT working process: Input data (left), polygonal approximation (center) and feature matching (right).

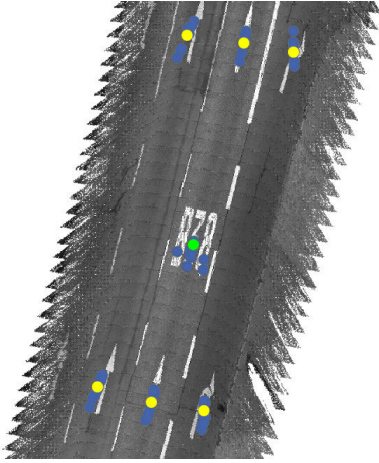


Fig. 9. Result of clustering (yellow and green dots) of all single detections (blue dots) on an accumulated top-view map.

IV. EVALUATION

Evaluation of the algorithm is done on the Bertha Benz Memorial Route [3]. This 104km track leads through both urban and rural areas in southern Germany. The sequence is recorded with two backwards facing cameras on our research vehicle “Anniway”. Base width of the stereo setup is approx. 38 cm. Both cameras provide an image resolution of $1238 \text{ px} \times 370 \text{ px}$ with an acquisition rate of 10 Hz. All images are rectified and the transformation between camera and vehicle coordinate system is known. In case of the monoscopic application, only the left camera is used.

To evaluate the system, all symbols on our ego lane are manually labeled. Those symbols are expected to be detected completely, since effects by occlusion can be neglected and stereo reconstruction is available. In total 398 symbols are painted on the 104km road, including 364 arrows, 12 speed limits and 9 “BUS” symbols. With both, monoscopic and stereoscopic system about 80% of the symbols are detected. Only 2.5% of the detected symbols are incorrectly classified. Table I provides a detailed overview of the result.

The large number of not detected straight arrows compared

Symbol	Existent	Detected Mono	Detected Stereo
Total	398	314 (80%)	317 (80%)
↑	222	175 (79%)	185 (83%)
↑↑	80	67 (84%)	60 (75%)
↑↑↑	42	41 (98%)	41 (98%)
↑↑↑↑	11	6 (55%)	7 (64%)
↑↑↑↑↑	8	7 (88%)	6 (75%)
↑↑↑↑↑↑	1	1 (100%)	1 (100%)
△	1	0 (0%)	0 (0%)
BUS	9	4 (44%)	3 (33%)
30	8	4 (50%)	3 (27%)
50	3	3 (100%)	3 (100%)
70	1	0 (0%)	1 (100%)
↑↑	5	5 (100%)	3 (60%)
↑↑↑	1	1 (100%)	1 (100%)
△	1	0 (0%)	0 (0%)
B38, FEU	5	4 (80%)	3 (60%)
Incorrect	-	8	8
False positives	-	20	31

TABLE I

DETECTION RATE OF ALL SYMBOLS ON OUR EGO LANE IN A 104 KM RURAL AND URBAN TEST DRIVE.

to other arrows is based on a misclassification as straight line if the painting is slightly damaged. The reversed case of damages dashed lines in curves, which are sometimes classified as straight arrows is the main reason for false positive detections. A further reason for false positives is the detection of nearby symbols not labeled in the ground truth data. While the ego lane is clearly visible in the accumulated top-view map we used for annotation, in the detection evaluation every symbol is considered as ego lane that has less than 1.5 m distance to the driven trajectory.

Another problem occurs with complex paintings, e.g. △. Here the approximation in TESSERACT (see Sec. III-E) is not fine enough to match the symbol correctly. A solution would be to only detect the triangular outline and classify the inner symbol separately.

Symbols on neighboring lanes are detected depending on visibility. While in the monoscopic case an area of $12 \text{ m} \times 15 \text{ m}$ behind the car is evaluated, we limited the stereo top-view detection area to half the width, due to large reconstruction errors in distance. Often other vehicles partly occlude the markings, so they are only visible in accumulated top-views. Thus, a comparable evaluation of the total number is not possible with our existing ground truth data. However, in the stereoscopic case, further 210 symbols outside the ego lane are detected, while with monoscopic images even further 539 symbols are found. Fig. 10 gives an example of detected symbols on a more complex scene.

To evaluate the position accuracy of the algorithm, we have manually labeled 50 straight arrows with start and end position. One problem occurs in the road markings itself: The mean length of an arrow is 5.02 m, while the shortest and longest labeled arrow is 4.70 m and 5.54 m respectively. Table II shows the lateral and longitudinal deviations from the result to the midpoint of the labeled data. Note that the

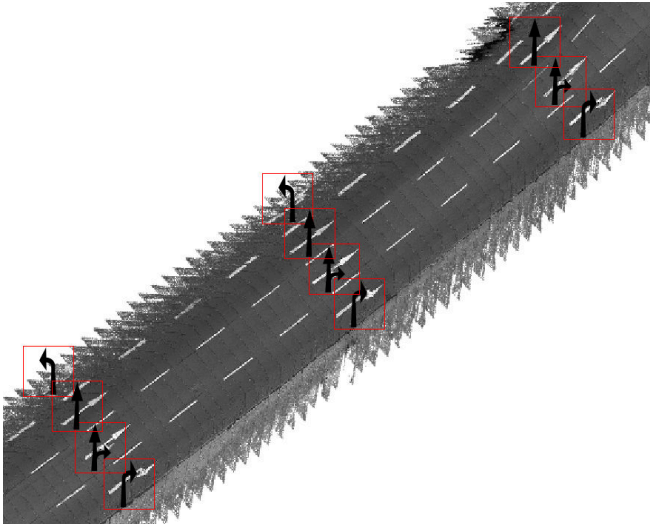


Fig. 10. Example with symbols detected on multiple lanes on accumulated top-view map for visualization.

Error	Monoscopic	Stereoscopic
Lateral		
mean deviation	0.009 m	0.044 m
standard deviation	0.075 m	0.065 m
min deviation	-0.105 m	-0.059 m
max deviation	0.305 m	0.274 m
Longitudinal		
mean deviation	0.316 m	-0.143 m
standard deviation	0.423 m	0.178 m
min deviation	-0.784 m	-0.816 m
max deviation	1.089 m	0.105 m

TABLE II

SPACIAL DEVIATIONS IN CASE OF STRAIGHT ARROWS.

system returns the centroid of the symbol, while only start and endpoint are labeled. This also affects images with partly occluded symbols, which leads to slightly shifted centroids in detection. Here, the advantage of stereoscopic top-views is clearly visible. Obtaining the position with vanishing point estimation and plane assumption in the monoscopic case leads to doubled standard deviation in longitudinal deviation compared to stereoscopic top-views. However, the lateral deviation is comparable in both cases.

Computing time on our system (Intel Xeon @2.5GHz) is between 20 ms and 70 ms per frame, depending on vanishing point detection and number of generated OCR candidates.

V. CONCLUSION

We presented a system to detect symbols on road surface. In contrast to other similar solutions, we completely rely on legal road symbol definitions, with which we create a special font for OCR. The system works both with monoscopic and stereoscopic camera systems to generate the required top-views. Stereo top-views provide good position accuracy, while monoscopic images benefit from a larger field of view with less distortions.

Both systems detect approximately 80% of the symbols on the ego lane. Two main problems still occur: Non standard road paintings and occlusion by others. The latter could be solved by accumulating image information similar to our top-

view generation for visualization and will be addressed in future work.

REFERENCES

- [1] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [2] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [3] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, E. Kaus, R. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoepfel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb, "Making bertha drive. an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, pp. 8–20, Summer 2014.
- [4] E. Dickmanns, B. Mysliwetz, and T. Christians, "An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 6, 1990.
- [5] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [6] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," in *IEEE Intelligent Vehicles Symposium*, 1995.
- [7] J. C. McCall and M. M. Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *IEEE Intelligent Vehicles Symposium*, 2004.
- [8] U. Franke, H. Loose, and C. Knoppel, "Lane recognition on country roads," in *IEEE Intelligent Vehicles Symposium*, 2007.
- [9] P. Greiner, M. Enzweiler, C. Knoepfel, and U. Franke, "Towards multi-cue urban curb recognition," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [10] J. Rebut, A. Benshair, and T. G., "Image segmentation and pattern recognition for road marking analysis," in *IEEE International Symposium on Industrial Electronics*, 2004.
- [11] W. Nan, L. Wei, Z. Chunmin, Y. Huai, and L. Jiren, "The detection and recognition of arrow markings recognition based on monocular vision," in *IEEE International Symposium on Industrial Electronics*, 2004.
- [12] L. Ziquiong, W. Shengjin, and D. Xiaoming, "Roi perspective transform based road marking detection and recognition," in *International Conference on Audio, Language and Image Processing*, 2012.
- [13] F. Foucher, Y. Sebsadki, J. Tarel, P. Charbonnier, and N. P., "Detection and recognition of urban road markings using images," in *14th International Conference on Intelligent Transportation Systems*, 2011.
- [14] T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *IEEE Intelligent Vehicles Symposium*, 2013.
- [15] R. Smith, "An overview of the tesseract ocr engine," in *International Conference on Document Analysis and Recognition*, vol. 2, 2007.
- [16] R. Smith, "Tesseract ocr engine." Presentation available on <http://tesseract-ocr.googlecode.com/files/TesseractOSCON.pdf>, 2007.
- [17] I. Posner, P. Corke, and P. Newman, "Using text-spotting to query the world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [18] W. Wu, X. Chen, and L. Yang, "Detection of text on road signs from video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 378–390, 2005.
- [19] German Department of Transport, *Richtlinie fuer die Markierung von Strassen (Instructions on Markings on Roads)*. 1980.
- [20] J. Ziegler, H. Lategahn, M. Schreiber, C. Keller, C. Knoepfel, H. Jochen, M. Haueis, and C. Stiller, "Video based localization for bertha," in *IEEE Intelligent Vehicles Symposium*, 2014.
- [21] J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic hough transform," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999.
- [22] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *International Workshop on image processing: Real-time edge and Motion detection/estimation*, 1979.
- [23] F. Meyer, "Color image segmentation," in *International Conference on Image Processing and its Applications*, 1992.