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

# Enriching a Spatial Road Representation with Lanes and Driving Directions

## Edoardo Casapietra, Thomas H Weisswange, Christian Goerick, Franz Kummert

### 2016

**Preprint:**

This is an accepted article published in IEEE 19th Intelligent Transportation Systems Conference (ITSC 2016). The final authenticated version is available online at: https://doi.org/10.1109/ITSC.2016.7795768

# Enriching a Spatial Road Representation with Lanes and Driving Directions

E. Casapietra, T.H. Weisswange, C. Goerick, F. Kummert

*Abstract*— **The detection of lane layout in the surroundings of the ego-vehicle is a key issue for modern ADAS and autonomous driving. Most modern systems rely on annotated spatial maps to provide lane information. However, these maps are not available everywhere, and thus have to be often supported by direct detection systems (e.g. cameras, lasers). Such systems detect lane boundaries by directly observing lane markings or sensing curbstones and pavements. However, well defined boundaries are not always there, or they can be difficult to detect, especially in rural or inner city roads. Furthermore, other traffic participants in the surroundings can significantly limit the field of view of the ego-vehicle. In order to help lane detection in this kind of scenario, indirect cues, such as the behavior of the other vehicles, can be a useful resource.**

**In this paper we propose a grid-based approach, that works on a road terrain representation and assigns a lane and a driving direction to each patch of road. Our approach segments the road terrain into multiple lanes based on geometrical considerations and prior knowledge, and then assigns a driving direction to each lane, taking into account the observed and predicted motion of the other vehicles in the scene. The road terrain is provided by a system we proposed in a previous work, which combines direct and indirect cues to build a probabilistic road representation grid. This new approach constitutes an additional layer to our representation, providing more awareness about the road regulations to motion control systems and allowing them to plan safer trajectories.**

## I. Introduction

Modern Advanced Driver Assistance Systems (ADAS) and future autonomous driving vehicles require a rich spatial representation of the environment they move in [16][10][11][13]. The mere knowledge of the road layout is not enough, as those systems require awareness of the rules and conventions that are necessary for moving safely in traffic. One of these conventions is the lane layout. In particular, it is necessary for the system to know which driving directions are allowed in which sections of the road. Lane information is critical on multiple levels of planning. On a high level, a system has to be aware of how many lanes each road segment has, and which direction they allow, so that it can plan maneuvers like overtakes or safely react to maneuvers of other vehicles. On a lower level, systems need to know exactly where those lanes are, so that they can compute a safe trajectory, without trespassing the correct lane. Some lane-level information is already used by many commercial ADAS, like lane keep assistants (LKA), which need to know where the lane boundaries are in order to keep the ego-vehicle inside the lane, or adaptive cruise control (ACC), which uses lane information to determine which vehicle to follow. To some extent, the arrangement of the directions of the various lanes is not arbitrary, and follows a few well established rules. For instance, in a multi-lane road, lanes with the same directions are adjacent to each other, and they are arranged in such a way that they have the nearest road boundary on their right side, with respect to their direction. It is possible for a system to infer some information from these rules, but ultimately each road is unique and for high quality results, more individual information has to be taken into account.

The most reliable way of obtaining lane information is currently the use of annotated spatial maps. These maps provide precise lane information, but they have to be actively annotated by human users, which poses the problem of availability of these maps especially for secondary roads. Furthermore, the ego-vehicle needs to be precisely localized and its pose has to be correctly estimated for a map to be useful. In order to cope with all these problems, many systems also use sensors, such as cameras, lasers or radars, to directly detect the lanes and to either support maps or completely replace them.

The primary method to detect lanes is defining them by sensing lane boundaries. This task can be achieved in different ways: radars and lasers can detect curbstones and guard rails by sensing height differences, but these object are often present only at the edge of the road, and thus do not help at detecting lane-to-lane boundaries. In order to detect those, the most common cues are lane markings. These can be detected in some cases by lasers[1] (measuring the intensity of the reflection from the road surface), but this option is limited in range, as laser beams need a certain angle to produce sufficient reflection. Furthermore, lasers have trouble detecting lane markings if the latter are dirty or worn out. Cameras provide a more versatile and cheaper method of detection, that can also convey a wider range of information. Image processing based methods have been extensively developed[1] for road and lane detection. Since the most useful and exploited cue are lane markings, the methods that use them are said to perform *marked* detection.

However, there are many cases where the lane-to-lane boundary is hardly visible, for example on secondary or rural roads, and frequently it may also be not marked at all. In these cases marked detection cannot provide sufficient information. The other type of lane detection, called *unmarked*, can be very effective to detect lanes, if those have different textures, which is not a common occurrence. In most cases it simply employs spatial priors such as the average width of a lane and geometric considerations. Furthermore, it has to be noted that camera-based detection systems are heavily influenced by environmental factors like shadows and bad lighting, and especially occlusions, which in case of heavy traffic can become a big issue also for lasers and any other

kind of direct sensor. Since these problem are intrinsic to direct detection and cannot be completely eliminated, research has investigated other forms of detection that are not based on directly sensing the road, but sensing other cues that are related to it instead. These approaches are said to do *indirect* detection. The most promising indirect cue is the behavior of other vehicles in the scene. In particular, the trajectories of other vehicles can be used as an additional source of information for both road and lane layout. Indirect detection exploits the road knowledge of other actors in the scene.

In this paper we present an indirect approach, based on a grid-based 2D road representation, where each patch of the grid is assigned to be either road or not. Our approach segments the given road representation into lanes based on width and geometry, and then uses the behavior of other vehicles in the scene to estimate the direction of each lane. This last step is done by using an approach (similar to our previous work[3]) where each patch holds a list of all the vehicles in the scene that can possibly traverse it in the near future or in the past, with an associated probability. This list will be used to determine the direction of each lane, creating rough but quick statistics of the driving directions of each area of the road, thus providing an immediate benefit even in case of low traffic. This approach can work with any regular square-grid road representation. We will show this by using the presented approach with the probabilistic road representation we described in our previous works. This approach can in fact be seen as a top layer of a full road terrain representation system.

## II. Related Work

Current state-of-the-art autonomous vehicles employ accurately annotated lane-level maps to obtain lane information. For these systems, the main problem to solve is self-localization and pose estimation[15][10][13]. The vehicles competing in the DARPA Urban Challenge carried multiple LIDARs, RADARs, as well as high-end GPS and IMU sensors, and were provided with a detailed digital map of the road network, complete with updated aerial images[13][1]. Daimler's *BERTHA*[16][5] relied on vision, radar and high precision GPS sensors to localize itself within an accurately annotated digital map. However, it is not feasible for commercial systems to rely only on annotated maps and high quality localization systems, as it would be too expensive and not flexible enough. For them, onboard sensing performed with affordable sensors is the most viable option[1].

Visual detection systems mainly focus on road detection, but most of them can also detect lanes, with particular attention to the ego-lane. An example is the RTDS, by Fritsch et al.[6]. It is a two-stage visual detection system. The first stage analyzes low-level features such as color and texture, and the second applies high-level spatial features (SPRAY) to classify road terrain and ego-lane.

The kind of representation employed for lanes vary greatly between different approaches, ranging from 2D probabilistic grids, to clothoid-model lane boundaries[4][9][14]. A notable lane representation is the one proposed by Bender et al.[2], that exploits the notion of lanelets (atomic, interconnected road segments) to represent both topological and geometrical aspect of lanes. Guo et al.[7] uses the past trajectories of other vehicles to refine the parameters of a clothoid modeling lateral lane boundaries, improving the robustness of lane detection in challenging scenarios. Thomas et al.[12] combine direct lane detection with the trajectory of vehicles in the scene, which are used as a cue to for the lane center line. The resulting representation is a probabilistic spatial grid, where semantic functions created from different cues (lane dividers, road boundaries, dynamic objects) are merged together to obtain the probability of being part of the center line for each square of the grid.

Our road representation[3], which we will use together with the present work, is a 2D spatial grid where each square has a probability of being road. The system computes the probability by merging different sources: a direct detection system (the RTDS[6]) and an indirect cue, the behavior of other vehicle in the scene. The vehicles are observed and their future and past positions are estimated. Assuming those vehicles are driving on roads, their motion is used to infer the road layout.

## III. Concepts

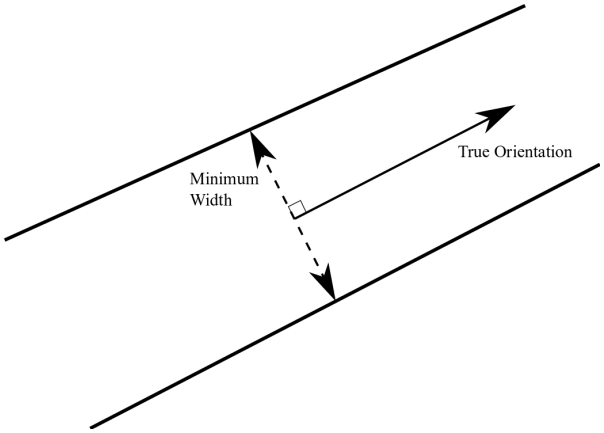
### A. Definitions

The following approach assumes a 2D regular square grid as input, where each square (hereby called "patch") holds a boolean value that indicates whether it is considered part of the road or not. The fundamental assumption of our model is that the true orientation of a road is orthogonal to its width segment (defined as the shortest distance segment between its two boundaries, figure 1a). This assumption arises from the basic model of a road with parallel boundaries. Since actual roads are not always like that, our approach will be carried out in a distributed fashion, where each patch will hold a "road width" and a "road orientation". Here, the "road width" of a patch is meant as the shortest distance segment passing through the patch itself, and its "road orientation" is the orientation orthogonal to its "road width".

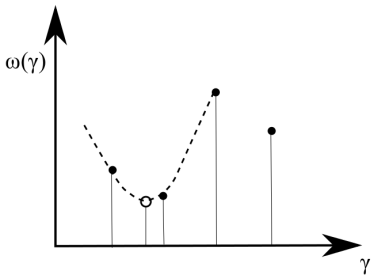
### B. Road Orientation and Width

The first step that our approach has to take is to estimate the direction and width of the roads. For every patch $i$ (we will use only one index for simplicity) that is considered road, the system estimates its road orientation $\widehat{\gamma}_i$ and road width $\widehat{\omega}_i$.

The algorithm starts by measuring the width of the road orthogonal to a limited number of orientations. For simplicity, we chose to use a set of the four main orientations (N-S, E-W, SE-NW and NE-SW, see figure 1a), that we define using the angles $\gamma_n$ in the set $\Gamma = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$ . This is done by slicing the whole map orthogonally to each orientation, looking for consecutive road patches. Every time a "string" of road patches is encountered, the systems assigns to each patch the width of the string and its relative position in it. After the process is completed for all main orientations,




(a) Parallel boundaries road model




(b) Exemplary road width samples and interpolation

Fig. 1: The road model (a) and the width interpolation (b).

each road patch holds temporarily four width values and four relative positions.

Each possible orientation angle (included in the interval $[0, \pi)$) delivers only one road width value by construction. Therefore, the road width can be seen as a function of the orientation angle. According to our hypothesis, the true road orientation is the one that delivers the minimum road width:

$$\begin{aligned} &\omega_i \longrightarrow \omega_i(\gamma) \\ &\widehat{\omega}_i = min\left[\omega_i(\gamma)\right] \\ &\widehat{\gamma}_i = argmin\left[\omega_i(\gamma)\right] \end{aligned} \tag{1}$$

The function $\omega_i(\gamma)$ is defined for every $\gamma \in \Re$, and is periodic so that $\omega_i(\gamma) = \omega_i(\gamma + \pi)$. The four road width values measured by the system (with $\gamma_n \in \Gamma$) are samples of this function. Measuring more road width values would increase the accuracy of the estimation, at the price of an increase in computational time. Since the true function $\omega_i(\gamma)$ has a shape which depends on the actual shape of each road (so it could be different in different patches), we estimate the minimum by a generic second-order interpolation: the system fits a parabola to the values of $\gamma_n$ at the lowest computed $\omega_i(\gamma_n)$, and its two neighbors.

$$\begin{aligned} &\widehat{\gamma}_{n,i} = argmin\left[\omega_i(\gamma_n)\right] \text{ for } \gamma_n \in \Gamma \\ &\begin{cases} \omega_i\left(\widehat{\gamma}_{n,i}\right) & = A\widehat{\gamma}_{n,i}^2 + B\widehat{\gamma}_{n,i} + C \\ \omega_i\left(\widehat{\gamma}_{n,i} + \frac{\pi}{4}\right) & = A\left(\widehat{\gamma}_{n,i} + \frac{\pi}{4}\right)^2 + B\left(\widehat{\gamma}_{n,i} + \frac{\pi}{4}\right) + C \\ \omega_i\left(\widehat{\gamma}_{n,i} - \frac{\pi}{4}\right) & = A\left(\widehat{\gamma}_{n,i} - \frac{\pi}{4}\right)^2 + B\left(\widehat{\gamma}_{n,i} - \frac{\pi}{4}\right) + C \end{cases} \end{aligned} \tag{2}$$

After solving the linear system and obtaining the parameters of the parabola $A, B, C$, finding the minimum is straightforward:

$$\begin{aligned} \widehat{\gamma}_i &= \frac{-B}{2A} \\ \widehat{\omega}_i &= C - \frac{B^2}{4A} \end{aligned} \tag{3}$$

The relative position of the patch into the road is obtained with a similar concept, considering it a function of $\gamma$ by linearly interpolating the relative positions measured for the two main orientations closest to $\widehat{\gamma}_i$ and computing the relative position in correspondence of $\widehat{\gamma}_i$.

In order to alleviate possible errors, orientation, width and relative position are filtered by spatial convolution with a 5x5 Gaussian mask, and finally stored into each patch.

### C. Lane Segmentation

The next step taken by the system is road segmentation. Each road patch of the map is scanned and neighboring patches are clustered together, based on their road width, road orientation and proximity. Two patches are considered part of the same road segment if they have no more than one patch between them, the difference between their road width is less than $1m$ and the difference between their orientation is less than $10\,\mathrm{deg}$. The road segments are distinguished by assigning to each patch the ID of its segment. After the clustering, the system computes the width of each road segment by averaging over the values of all its patches, and then assigns a certain number of lanes to each segment, based on its width and a predefined average width of a lane (for the experiments presented in section IV we used the average width of 3.2m, as a good compromise between typical lane widths for primary and secondary roads in Germany[8]). After the number of lanes is determined, the system divides each road segment into the assigned number of lanes with equal width. This step is performed by checking the relative position of each patch in the road and assigning an ID to it, depending on the lane it falls into. The system assigns ID 0 to the rightmost lane, assuming the driving direction equal to

its orientation $\widehat{\gamma}_i$. This direction is set as the default direction $\overline{\phi}_i$ of each patch in the lane. The other lane IDs are assigned incrementally towards the left. The system assumes each road segment as two-way, and assigns a default direction $\overline{\phi}_i = \widehat{\gamma}_i$ to all the patches belonging to the right half of the road segment (in case of odd number of lanes, the central lane is included), and assigns a default direction $\overline{\phi}_i = \widehat{\gamma}_i + \pi$ to the patches belonging to the left half.

### D. Driving Directions

The final step involves using motion of all past and present vehicles in the scene. Just as we did in our previous work[3], every patch $i$ of our road representation holds a list with all the vehicles $j$ in the scene that have a probability $p(C_{i,j})$ to traverse it in the future, present or past. Those probabilities are computed by observing their motion and estimating their possible future and past positions. The formula we adopted for estimating $p(C_{i,j})$ is the following:

$$p(C_{i,j}) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{X_{i,j}+\eta_j}{\sqrt{2}\sigma_j}\right) - \operatorname{erf}\left(\frac{X_{i,j}-\eta_j}{\sqrt{2}\sigma_j}\right)\right]$$
$$\sigma_j = \begin{cases} \zeta_j & for\,|Z_{i,j}| \leq \frac{L_j}{2} \\ \zeta_j + A\frac{Z_{i,j}}{|\vec{v}_j|} & for\,|Z_{i,j}| > \frac{L_j}{2} \end{cases} \quad (4)$$

Here $[X_{i,j}, Z_{i,j}]$ is the position of patch $i$ in the reference system of vehicle $j$, $L_j$ is the length of the vehicle $j$, $\eta_j$ is a parameter related to the width of it, $\zeta_j$ is related to the uncertainty of its position (accounting for precision and reliability of the detection system), and $A$ is a tuning factor.

The probabilities $p(C_{i,j})$ are stored in the aforementioned list, along with the direction $\phi_{i,j}$ that the vehicle $j$ had or would have (assuming a regular motion) on patch $i$. All these estimations are performed only once when they are related to past events (i.e. behind a vehicle that has just been detected), while they are refreshed at every time step for future events (i.e. in front of each vehicle).

### E. Final Directions

Only vehicles that have a probability $p(C_{i,j})$ higher than a certain threshold $\theta$ are considered. The threshold is arbitrary, and its sole purpose is to speed up the algorithm by discarding less relevant vehicles, something that can become an issue in case of heavy traffic. The estimated direction $\phi_{i,j}$ of all remaining cars in the list is checked and those that have an incompatible direction with respect to the road orientation are discarded. The remaining directions, that are similar to the road orientation within a certain angle limit, are used to decide by means of a weighted sum whether the direction of the patch has to be inverted or not. If $\Delta$ is the set of all vehicles whose directions are consistent with the default direction of the patch, and $\Theta$ the set of all vehicles whose direction is consistent with the opposite of the default, we have:

$$\phi_i = \begin{cases} \overline{\phi}_i & \text{if } \sum_j p(C_{i,j}) - \sum_k p(C_{i,k}) \geq 0 \\ \overline{\phi}_i + \pi & \text{if } \sum_j p(C_{i,j}) - \sum_k p(C_{i,k}) < 0 \end{cases} \quad (5)$$

Where $j \in \Delta$, $k \in \Theta$

Having the directions distributed over all patches can allow certain areas of a lane to have a different direction from the rest of it, depending on the motion of the traffic in the scene.

### F. Issues

The assumption stated in subsection III-A, derived from the model of a straight road with parallel-boundaries, does not hold inside intersections. Here, the geometry of the road is completely different from the original model, and finding a road orientation and road width for each patch is an ill-posed problem. However, we could still apply our algorithm to scenes where intersections are present, in order to see how the system behaves there. Figure 2 shows the output of an ideal $90\,\mathrm{deg}$ intersection, made by two orthogonal $10m$ wide roads. The road orientation inside the intersection seems inconsistent (figure 2a), showing a large variety of unrelated values changing abruptly in contiguous patches, while the road widths appear much larger than expected (figure 2b), approximately twice as large as the roads themselves. These features make it possible to distinguish and segment intersections in post-processing. Although the orientation and width values are not usable, the system currently is at least capable of recognizing intersections.

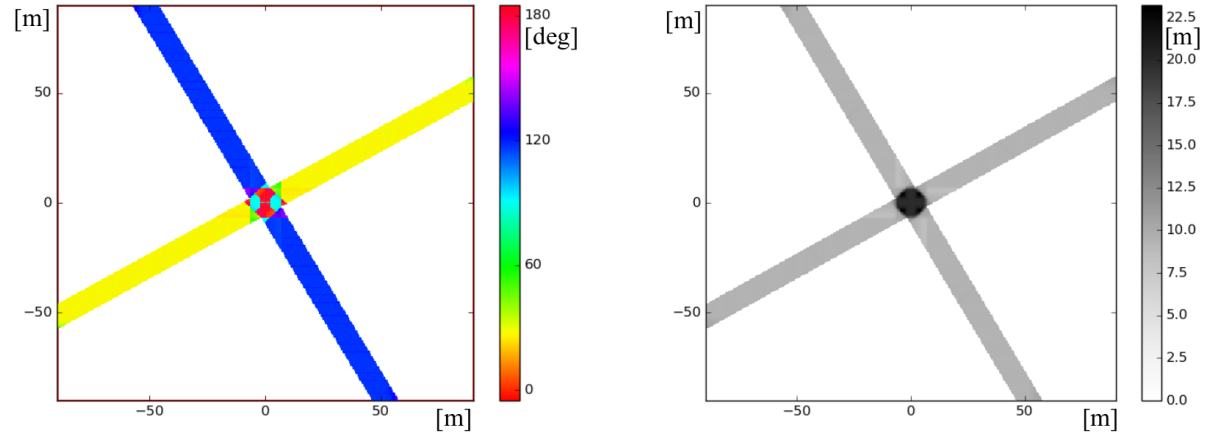
Another issue occurs in correspondence of curves in wide roads. It can happen that one of the orientations next to the one with minimum width presents a very large width, due to the road curvature. Keeping the notation as in (2), this occurs when $\omega_i\left(\widehat{\gamma}_{n,i} \pm \frac{pi}{4}\right) >> \omega_i\left(\widehat{\gamma}_{n,i}\right)$. This makes the second-order interpolation unsuited to find a reasonable minimum, often resulting in a value of $\widehat{\omega}_i$ that is much lower than the true width (sometimes negative). This problem could be dealt with by increasing the number of orientations to compute and interpolating with $\omega_i\left(\widehat{\gamma}_{n,i} \pm \frac{pi}{8}\right)$ instead.

## IV. Experiments

In order to show the potential benefits of our approach, we will show a few qualitative tests on real world data. For a proper evaluation, we will need a much larger dataset. Before the real world data, however, we evaluated the geometric approach explained in section III-B on ideal scenes, to show the performances of the estimation, which is fundamental in our whole system.

### A. Road Direction and Width Estimation

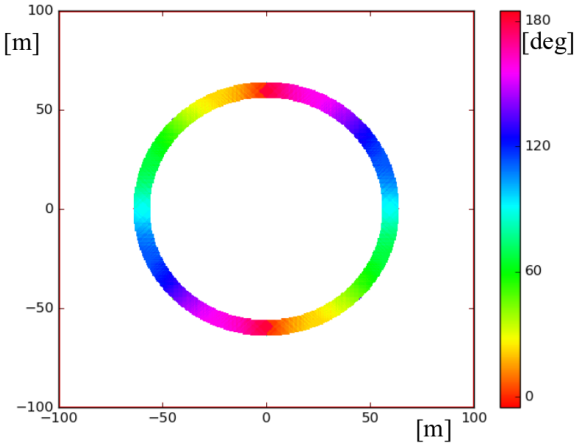
In this section we evaluate our model estimating only road orientation and road width. We will test our approach on two scenarios, drawn on a $200m$ x $200m$ square grid, with patches of size $0.5m$ x $0.5m$ . The first one is a long, straight road. We will test the performance of the algorithm on roads with different orientations and widths. Since our approach is based on the 4 main orientations stated in III-B, using those orientations for the test would



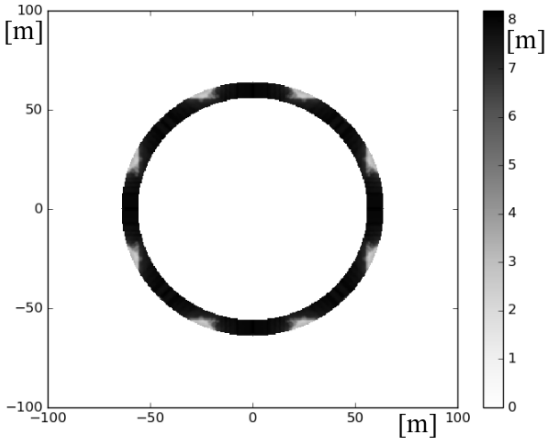

(a) Orientation angle




(b) Width




(c) Orientation angle

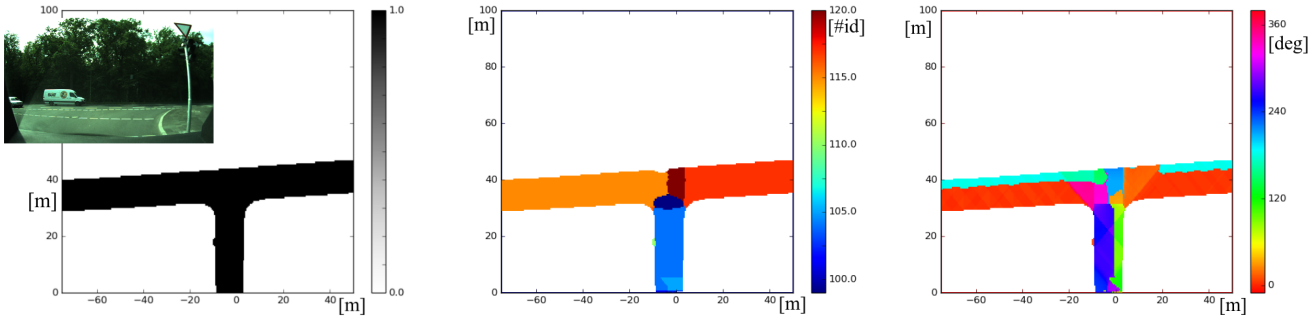



(d) Width

Fig. 2: (a)(b): Output of the orientation and width estimation, performed on a $90\,\mathrm{deg}$ intersection made by two $10m$ wide roads. (c)(d): Output of the orientation and width estimation, performed on a circular road with radius $60m$ and width $8m$.

not be significant. Instead, we will use the following set of orientations: $18\,\mathrm{deg}, 22.5\,\mathrm{deg}, 30\,\mathrm{deg}, 36\,\mathrm{deg}$. The results show that the approach is very robust in this particular scenario: the average error for road width was between $-1.3m$ (the worst scenario, occurring for a $12m$ road with an orientation of $22.5\,\mathrm{deg}$) and $-0.1m$ (occurring for a $4m$ road with orientation of $36\,\mathrm{deg}$). The mean square was between $0.9m^2$ and $3.3m^2$. The average absolute error for road orientation ranged from a best case of less than $0.1\,\mathrm{deg}$ to a worst case of $1.8\,\mathrm{deg}$, occurred once again at the widest road and the orientation of $22.5\,\mathrm{deg}$, which is the most distant from any of the main orientations we use in the algorithm. The percentage of patches with an orientation error of less than $5\,\mathrm{deg}$ was $80\%$ in the worst case and superior to $85\%$ in all the others, reaching $100\%$ in a few cases. The percentage of patches with an orientation error of less than $10\,\mathrm{deg}$ was always superior to $99\%$.

The second scenario was meant to test the performances on curves. For this purpose, we drew a perfectly circular road and tested our algorithm on it, using different radii and widths. By doing so, we make sure the approach is tested on all orientations. The radii we used were $40, 50, 60, 70, 80$ meters, while the road widths were $4, 6, 8, 10, 12$ meters. This experiment clearly shows the issue discussed in III-F: the width is underestimated at the outer part of wide curves. The effect is more evident at smaller radii, and it occurs systematically at certain orientations, while at others it does not. In figures 2c and 2d an example is shown, with a width of $8m$ and a radius of $60m$. It is apparent that the most troublesome orientations are indeed the ones that are in between the four main orientations used by the algorithm. Over the whole experiment the average error in width was between $-0.2m$ (occurring for all narrow roads, regardless of the radius) and $-1.8m$ (occurring for the $12m$ wide road and $40m$ radius), showing that the approach does seem slightly biased towards lower width values. As for the directions, the average absolute error was between $2\,\mathrm{deg}$ and $4\,\mathrm{deg}$, with the worst performances occurring in the same cases as the




(a) (b) (c)

Fig. 3: Exemplary image (top left) and annotated road representation input (3a), road segmentation (3b) and output of the lane direction estimation (3c), performed on a real world intersection.

worst width estimations. Here, the larger the radius, the better the performances. The percentage of patches with a direction error lower than $5\,\mathrm{deg}$ ranged from $76\%$ in the worst situation to $97\%$ in the best, being over $85\%$ in 23 cases out of 25. The percentage of patches with a direction error lower than $10\,\mathrm{deg}$ was $94\%$ in the worst case and was never lower than $98\%$ in all the others, reaching $100\%$ in multiple cases.

### B. Real World

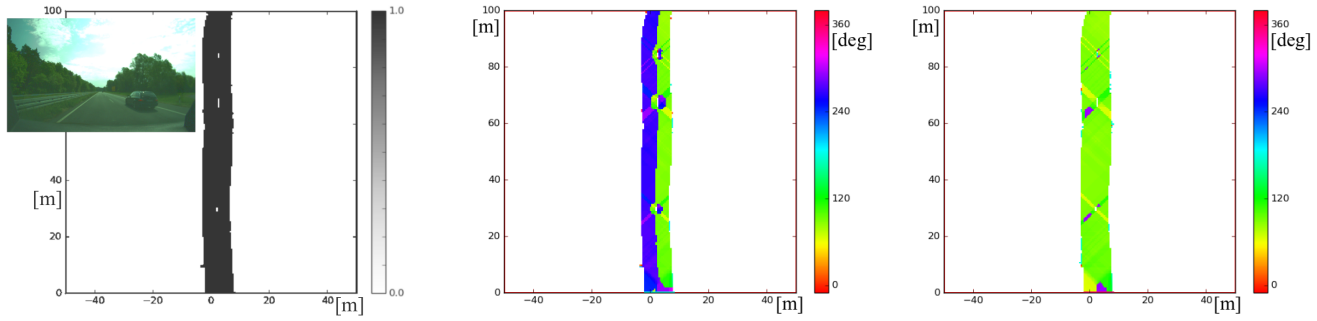
In this section we test our system on real world data, to show the potential of our approach. The data were taken by our mobile platform, equipped with 6 Ibeo Lux Lidars including Ibeo Lux Fusion System for 360 degree tracking, and an IDS camera in the front windshield. The recordings have been done in the inner city of Offenbach, Germany.

In the first experiment (figure 3) a complex scenario is shown. In this scene, the ego-car is standing at a traffic light, on the rightmost lane of the bottom road, and observes a T-shaped intersection, made by three roads with three lanes each. The input road representation (figure 3a) has been created from manual annotations. The Lidars detect two vehicles, both coming from the left road. One of them follows the middle lane, going straight into the right road, while the other follows the rightmost lane and turns into the same road as the ego-car stands. Figure 3c shows the direction of each patch as inferred by the system. The system correctly divides all three road segments into three lanes and assigns the correct direction to the middle lane, from left to right, helped by the vehicle that traversed it. As for the road where the ego-vehicle is, no vehicles traverse the middle lane, and thus the system has no proper means to estimate the direction. However, the car turning into that road has a chance to be directed towards that lane, and in absence of any other cue the system assigns its direction to the lane. In figure 3b it is possible to appreciate the road segmentation performed by the current system, which is not meant to handle intersections yet. Even though inside the intersection itself the segmentation is irregular, the three roads are still clearly segmented.

The second experiment (figure 4) shows a two-lane straight road, with only one direction allowed. Here the input (figure 4a) is provided by our road representation system, thresholded at $75\%$, which we determined as the most effective threshold[3]. It has to be noted that the resulting road representation is not perfect, as it shows three small artifacts near the center line, in the form of patches incorrectly shown as not road. In the stream, the ego-car travels on the left lane, and overtakes another vehicle on the right lane. Figure 4b shows the lane segmentation performed by the algorithm, with the default directions assigned. Figure 4c shows the final direction of each patch. The figures shows that the system correctly divides the road into two lanes and assigns the correct direction to both of them, by observing the two vehicles in the scene. The artifacts present in the representation input do cause artifacts in lane segmentation and direction estimation, although the figures show that the effect is spatially limited.

## V. Conclusions

We presented a spatial grid-based lane representation, that uses a geometric approach to achieve a distributed estimation of the road orientation and width, as well as an *a priori* lane segmentation, and then employs the behavior of other vehicles (both observed and predicted) in the scene to determine the direction of each lane. Unlike other current approaches [7][12], ours does not model lanes with their boundaries or center line. Instead, we use a purely spatial approach, where each patch has to be assigned to a lane and has its own direction. We tested the estimated road orientation and width on two simulated scenarios, a straight road, with different orientations, and a circular road, with different widths and radii. Finally, we tested the system on real world scenes, using it together with our own road detection system, showing the potential of the approach, although we could not perform a full evaluation due to our small dataset. As it estimates lane directions in a spatially distributed fashion, this lane representation may allow a control system to plan safer maneuvers and trajectories in traffic scenarios.




(a) (b) (c)

Fig. 4: Exemplary image (top left) and road representation input obtained using our own road detection system (4a), default directions (4b) and final directions (4c) assigned by the algorithm.

## REFERENCES


[1] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: A survey, 2014.

[2] Philipp Bender, Julius Ziegler, and Christoph Stiller. Lanelets: Efficient Map Representation for Autonomous Driving. In *IEEE Intell. Veh. Symp. (IV 2014)*, pages 420–425, Dearborn, Michigan, USA, 2014. IEEE.

[3] Edoardo Casapietra, Thomas Weisswange, Franz Kummert, and Christian Goerick. Inferring a grid-based road representation from the behavior of real world traffic participants. In *IEEE Intelligent Vehicles Symposium 2016*. IEEE, June 2016.

[4] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-D Road and Relative Ego-State Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):199–213, 1992.

[5] Uwe Franke, David Pfeiffer, Clemens Rabe, Carsten Knoeppel, Markus Enzweiler, Fridtjof Stein, and Ralf G. Herrtwich. Making bertha see. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 214–221. Institute of Electrical and Electronics Engineers Inc., 2013.

[6] Jannik Fritsch, Tobias Kühnl, and Franz Kummert. Monocular Road Terrain Detection by Combining Visual and Spatial Information. *IEEE Trans. Intell. Transp. Syst.*, 2014.

[7] Chunzhao Guo, Unichi Meguro, Koichiro Yamaguchi, Kiyosumi Kidono, and Yoshiko Kojima. Improved Lane Detection based on Past Vehicle Trajectories. In *17th IEEE Int. Conf. Intell. Transp. Syst. (ITSC 2014)*, pages 1956–1963. IEEE, 2014.

[8] L.E. Hall, R.D. Powers, D.S. Turner, W. Brilon, and J.W. Hall. Overview of cross section design elements. In *International Symposium on Highway Geometric Design Practices*, 1995.

[9] C.R. Jung and C.R. Kelber. A robust linear-parabolic model for lane following. *Proceedings. 17th Brazilian Symp. Comput. Graph. Image Process.*, pages 72–79, 2004.

[10] Richard Matthaei, Gerrit Bagschik, and Markus Maurer. Map-relative localization in lane-level maps for ADAS and autonomous driving. In *IEEE Intell. Veh. Symp. Proc.*, pages 49–55. Institute of Electrical and Electronics Engineers Inc., 2014.

[11] Isaac Miller, Mark Campbell, Dan Huttenlocher, ..., Mike Kurdziel, and Hikaru Fujishima. Team Cornell's Skynet: Robust perception and planning in an urban environment. *J. F. Robot.*, 25(8):493–527, 2008.

[12] Julian Thomas, Kai Stiens, Sebastian Rauch, and Raul Rojas. Grid-based online road model estimation for advanced driver assistance systems. In *IEEE Intell. Veh. Symp. Proc.*, volume 2015-Augus, pages 71–76. Institute of Electrical and Electronics Engineers Inc., 2015.

[13] Chris Urmson, Joshua Anhalt, Drew Bagnell, ..., Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *J. F. Robot.*, 25(8):425–466, 2008.

[14] Yue Wang, Dinggang Shen, and Eam Khwang Teoh. Lane detection using spline model. *Pattern Recognit. Lett.*, 21:677–689, 2000.

[15] Julius Ziegler, Philipp Bender, Thao Dang, and Christoph Stiller. Trajectory planning for Bertha - A local, continuous method. In *IEEE Intell. Veh. Symp. Proc.*, pages 450–457. Institute of Electrical and Electronics Engineers Inc., 2014.

[16] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G. Keller, Eberhard Kaus, Ralf G. Herrtwich, Clemens Rabe, David Pfeiffer, Frank Lindner, Fridtjof Stein, Friedrich Erbs, Markus Enzweiler, Carsten Knoppel, Jochen Hipp, Martin Haueis, Maximilian Trepte, Carsten Brenk, Andreas Tamke, Mohammad Ghanaat, Markus Braun, Armin Joos, Hans Fritz, Horst Mock, Martin Hein, and Eberhard Zeeb. Making bertha drive-an autonomous journey on a historic route. *IEEE Intell. Transp. Syst. Mag.*, 6(2):8–20, 2014.