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

Inferring a Grid-based Road Representation from the behavior of real world traffic participants

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

2016

Preprint:

This is an accepted article published in IEEE Intelligent Vehicles Symposium 2016. The final authenticated version is available online at: https://doi.org/10.1109/IVS.2016.7535540

Inferring a Spatial Road Representation from the Behavior of Real World Traffic Participants

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

***Abstract*—The detection of road area in the surroundings of the ego-vehicle is a key issue for modern ADAS. Camera-based direct detection systems are able to reliably accomplish this task only within a limited spatial range or in simple environments, due to hardware limitations and unfavorable situations, like shadows or occlusions. In complex environments, like inner city, traffic is a real issue, since the mere presence of other cars can significantly restrict the field of view of the ego-vehicle. In order to extend the spatial range of road detection, indirect detection systems are a viable resource. They can complement state-of-the-art direct detection systems and help motion control systems to plan smooth and stable trajectories.**

In this paper we propose a probabilistic grid-based approach based on the interpretation of the motion of other vehicles in the scene. The approach uses the position and velocity of those vehicles in order to infer the presence and location of occluded road area. We will show that this approach can complement an already established feature-based detection system, taking advantage of those situations that are the most challenging for the latter.

Evaluations on real-world scenes show that the union between this approach and direct road detection significantly extends the spatial range of detection, thus is able to provide a motion control system a longer horizon for planning trajectories.

I. Introduction

Advanced Driver Assistance Systems (ADAS) are increasingly important in the car industry, as they provide a wide array of functions, both as warning systems (e.g. parking assistance) and temporary control systems (e.g. cruise-control in highways, emergency breaking). All these systems exploit a large selection of possible sensors to extract spatial information about the immediate surroundings of the vehicle. The spatial detection of road area is essential for many ADAS applications, as estimating the geometry of the road ahead is a key piece of information for both safety routines and motion control: regarding safety, the potential danger associated to an object is dependent on whether or not it is (or will be) on the road. On the other hand, any motion control system needs to know where the vehicle is allowed to move on at any time. State-of-the-art ADAS approximate road area by detecting free space.

Roads are an extremely dynamic environment, where timing is a key factor. It holds true for road detection: a road detected in advance can help to plan the motion of the ego-vehicle, or even prevent an accident. Depending on the time scale they operate on, ADAS can handle different levels of planning: the lowest is the reaction to unexpected events, that deals with a very short time horizon, where high spatial accuracy is required. This level is managed by using a wide choice of sensors (e.g. lasers, cameras), that detect road by sensing it directly. In this scenario the most versatile sensor is the camera, which however presents a few drawbacks: for camera-based detection, illumination is always an open issue which can be very disruptive in real world situations. Also, the road texture can consist of different patches of asphalt (or different kinds of surface, like cobblestones), and thus be hard to segment and classify correctly in feature-based classification. On the other hand, the highest level is the route planning, that works on a time horizon of minutes or even hours, and it does not require a high spatial accuracy. Modern ADAS cope with this level by using offline navigational maps, that provide knowledge of the road ahead for a virtually unlimited time horizon, but the information they provide is mainly topological, and the spatial accuracy that can be extracted from them is very low. The mid-level is the one that plans smooth trajectories, and deals with time horizons of a few seconds. State-of-the art control systems cope with this level of planning by using a mix of all the resources mentioned above[3][17][21]. The result, however, is far from optimal, since the reliability of both direct detection and navigational maps is not as high as desired at this time scale.

The insufficient spatial accuracy of the direct detection methods at a mid-level time horizon is due to various reasons. While the ego-vehicle is moving, a time horizon of a few seconds translates to a spatial range of tenths of meters. The longer the range is, the lower the accuracy, firstly because of hardware limits every sensor has (e.g. camera resolution), and secondly because of the nature of the road environment itself, where occlusions are very frequent and are a key aspect of the scenario: the traffic can hide large parts of road from our field of view (FOV), and the same is true for buildings and other obstacles. The parts of road that are hidden by occlusions are progressively larger with the distance, so that the farther away the road we look at is, the likelier it is that our sensors cannot sense it at all. It is important to note that all the problems mentioned above are most frequent in inner city environments rather than in more simple environments like highways.

In this paper we develop and evaluate our indirect road detection approach[7], which uses the motion of other vehicles in a scene as a cue to infer road probability on a 2D grid map. While driving, the event of a moving vehicle appearing in our field of view (from outside the FOV, or from behind an occlusion) is very frequent. This event provides information: the vehicle must have come from a road area we could not detect. The same can be said about the opposite event, when a vehicle disappears: it went to a road area that is unknown to us. Considering the direction and speed of the vehicle, one

can also infer the likely position of said road area, within certain limits. Our approach exploits this reasoning, and it is based only on the position and velocity of the vehicles in the scene, features that at mid-range can be estimated with far more reliability than road area itself[19].

Since it exploits the vehicles in the scene, our approach shows the biggest advantages in those situations that are most challenging for direct detection systems. However, it cannot be a standalone system, as it is too dependent on the availability of other traffic participants. In order to cope with its limitations, we designed an architecture that blends direct and indirect detection together. Our system uses, as an example of direct detection algorithm, the SPRAY-based visual road terrain detection system (RTDS) developed by Fritsch et al.[10]. We chose this system for its good performances in the KITTI benchmark[11], but any kind of direct detection system providing confidence levels can be used in this framework.

The evaluation made on real-world recordings shows that our approach improves significantly the road detection with respect to the RTDS performances. It can help greatly the detection of roads in intersections, and can also detect side roads that lie behind occlusions, and therefore are invisible to any direct detection system.

II. Related Works

Detecting the area where the vehicle can move is one of the most critical issues for ADAS, in particular for those that perform any kind of motion control task. A very popular system to represent free space is the occupancy grid, a spatial representation that describes the surroundings of the ego-vehicle in terms of free or occupied space. This kind of representation can be achieved by using a vast array of different sensors, like lasers[6][22], radars[6], or stereo cameras[4], and can also include prior map knowledge[13] or theoretical models, stochastic[4] or particle-based[8]. Another possible spatial representation for drivable road that has been implemented is the probabilistic drivability map[14][20].

When the attention shifts from drivable area to proper *road*, ADAS employ road detection systems. The most common and versatile sensor used for road detection is the camera. The first instances of visual road detection systems used to infer road area by detecting lane markings and applying prior knowledge about road geometry. This kind of systems, still valid and in use today, is now called ”marked” road detection, and it is most commonly used in simple and structured environments like highways. In environments where lane markings are scarce or absent, a different approach for visual road detection is necessary. ”Unmarked” road detection systems are able to infer road by observing low-level features, such as color and texture. Stixels have been employed to infer road area[15]. Visual feature have also been fused with data from Lidars to detect road area[24]. The road detection system that we use in this work is the SPRAY-based road detection system[10]. This method consists in two stages. The first is purely visual and extracts three base classifiers for the scene, while the second extracts spatial features (SPRAY) from the three base classifiers to classify road terrain as output. Researchers have also explored the possibility to realize hybrid approaches by combining low-level features with contextual cues (e.g. vanishing points and lane markings) and road priors obtained from navigational maps[2].

All these methods are considered direct detection, since they rely on sensors that directly interact with the road. However, in the latest years the research has started developing alternatives, indirect cues, to infer road area, due to the challenges that ADAS face in complex environments. The most promising indirect cue that has been investigated is the behavior of the other entities in the environment. Initially their behavior was the target to be inferred, and the road was the cue. Behavioral assumptions have been exploited to improve the tracking of vehicles[1], and to predict their motion at intersections[16]. Information about context has been used to predict the behavior and classify the related danger of pedestrians[5] or vehicles[23]. The approach we present in this paper does the opposite: it exploits the behavior of vehicles to infer the context. In this direction, the behavior of pedestrians has been used to infer navigational maps for robots[18]. The behavior of vehicles has been used, together with lane markings detection and context models, to infer road geometry at long range in highways[9]: this work used a Bayesian inference algorithm to compute the parameters of a clothoid-based road model. Similarly, the behavior of vehicles has also been used together with other visual cues in order to infer the topological and geometrical layout of road intersections[12], out of a predefined set of possible topologies, using a probabilistic generative model. However, both of these latter systems used only the current behavior of vehicles, while the approach we present in the following section exploits also the predicted behavior.

III. Concepts

A. Behavior Interpretation

The representation map we use is a 2D regular squared grid, where each square (called a ”patch”) is associated to a global spatial position and holds a probability of being road. We chose a fixed, global representation because our approach can potentially benefit from multiple passages over the same area, improving the representation at each passage. However, a local representation centered on the ego-vehicle, covering a certain spatial range (similar to the evaluation method we use in section IV-B), would also be an alternative and would alleviate the memory burden of the system.

The core concept of our approach is to take advantage of one of the main problems of direct visual road detection systems: other vehicles in the scene, that create occlusions. Our reasoning is that the motion of those vehicles holds valuable information that can potentially cover the loss of direct visibility. In fact, every vehicle in the scene has a different FOV, due to their different position and orientation, and thus a different internal representation of the surroundings (be it a human-driven vehicle or an autonomous one). By

observing and interpreting the motion of those vehicles, we can infer parts of their representations to complement ours.

The basic assumption of our approach is that all vehicles move almost exclusively on road area. Therefore, all patches that are traversed by a vehicle must be road. Similarly, if we can estimate which patches will be traversed by the vehicle in the future, or have been in the past, we can infer the road probability of those patches as well. We compute the road probability of patch i (for simplicity we will indicate a patch by using only one index instead of two) by using the following formula:

$$p(R_i) = p(R_i|\bar{C}_i) + \left[p(R|C) - p(R_i|\bar{C}_i)\right] p(C_i) \quad (1)$$

Here R_i is the event that the patch is road, and C_i is the event that the patch is traversed by any of the vehicles in the scene. The conditioned probability $p(R|C)$ (the probability of a patch being road, if it is traversed by a vehicle) is assumed constantly equal to 1 for every patch i (therefore it has no index), but could be set differently if necessary. Conversely, the conditioned probability $p(R|\bar{C}_i)$ is not constant but varies between different patches, and will be discussed later in this section.

In case there are multiple vehicles in the scene, $p(C_i)$ is computed by calculating first the single-vehicle probabilities $p(C_{i,j})$ (the probability that patch i was or will be traversed by vehicle j), and then summing them together as independent events:

$$\begin{aligned} p_n(C_i) &= p(C_{i,1}) + p(C_{i,2}) - p(C_{i,1})p(C_{i,2}) && \text{if } n = 2 \\ p_n(C_i) &= p_{n-1}(C_i) + p(C_{i,n}) - p_{n-1}(C_i)p(C_{i,n}) && \text{if } n > 2 \end{aligned} \quad (2)$$

We model the probability for a single vehicle with a 2D function, that provides a probability value given the relative location of the patch i in the reference system of vehicle j (X is the pitch axis and Z is the roll axis), and the speed of the vehicle itself. Since we want to estimate both past and future positions of the vehicle, we use a function symmetrical with respect to the X axis of the vehicle's reference system. Intuitively, the function produces high values immediately in front of the car, which fade away with distance. We chose to keep this probabilistic approach also under the car, so that we can model possible position errors due to inaccurate detection. The function is formulated as follows:

$$p(C_{i,j}) = \frac{1}{2}\left[\operatorname{erf}\left(\frac{X_{i,j} + \eta_j}{\sqrt{2}\sigma_j}\right) - \operatorname{erf}\left(\frac{X_{i,j} - \eta_j}{\sqrt{2}\sigma_j}\right)\right] \quad (3)$$

$$\sigma_j = \begin{cases} \omega_j & for\,|Z_{i,j}| \leq \frac{L_j}{2} \\ \omega_j + A\frac{Z_{i,j}}{|\vec{v}_j|} & for\,|Z_{i,j}| > \frac{L_j}{2} \end{cases} \quad (4)$$

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, η_j is a parameter related to the width of it, ω_j is related to the uncertainty of its position, and A is a tuning factor. A contour plot of this function is shown in figure 1a. The values in the points that lie in front of the car involve estimated *future* positions, and are refreshed at every frame, since they change with new observations of the vehicle, The values behind the vehicle instead are *permanent*, as they do not depend on the successive trajectory of the vehicle, and as such they are computed only once, when the vehicle information is available for the first time.

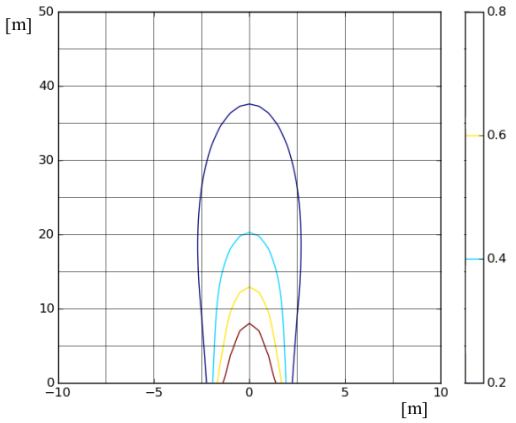
From (1) follows that vehicles can only increase $p(R_i)$. This fact could be potentially dangerous for the system, as multiple vehicles in the same area could easily end up saturating the whole representation. To compensate for this, we introduce a *negative* contribution, based around the term $p(R_i|\bar{C}_i)$. Since this term represents the probability that patch i is road given that no vehicles traverse it, it would be reasonable to think that its value is not the same for every patch. In fact, the event of a patch being left untouched holds information only if it can be argued that the vehicles *avoided* it. If no vehicle was likely to traverse it in the first place, the fact that no one did is not relevant. We model this reasoning by defining $p(R_i|\bar{C}_i)$ as dependent on $p(C_{i,j})$. Starting from a base value of 0.5, $p(R_i|\bar{C}_i)$ decreases with every vehicle that has a chance to traverse it (or to have traversed it). The formula is the following:

$$p(R_i|\bar{C}_i) = \frac{1}{2}\prod_{j=1}^{N}\left(1 - K p_{max}(C_{i,j})\right) \quad (5)$$

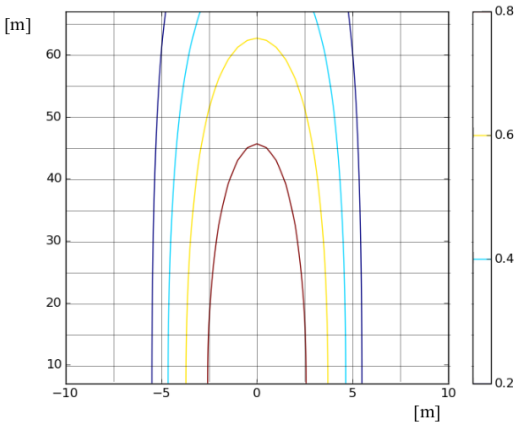
Here N is the total number of observed vehicles, K a tuning factor and $p_{max}(C_{i,j})$ is the maximum $p(C_{i,j})$ ever computed by the system. The contribution of each vehicle remains available also after the vehicle exits the scene (see III-B), and it always keeps being used, so that $p(R_i|\bar{C}_i)$ tends to decrease the more vehicles are observed in the vicinity. Its effect on the overall road probability is appreciable only when $p_{max}(C_{i,j})$ is significantly higher than the current $p(C_{i,j})$, which is the situation where the patch has been avoided by the vehicle.

B. Updating

As the detected position and velocity of vehicles change at every measurement, it is necessary to define a way to update the estimated $p(C_{i,j})$. For this purpose we distinguish two main cases: *temporary* and *permanent* estimations. Temporary estimations needs to be refreshed at every frame, as they depend on the current state of the observed vehicles. The estimated future positions of vehicles are considered temporary by our system. At every time step, the previous estimation will be dropped and replaced with a newer one. Conversely, permanent estimations are those that depend on the state of the



(a) Contour plot of $p(C_{i,j})$. The vehicle j is in $(0, 0)$, facing forward.



(b) Contour plot of $p(V_i)$. The ego-vehicle is in $(0, 0)$, facing forward.

Fig. 1

vehicle in a particular instant, and thus they do not change with further measurements. Estimated past and present positions are permanent: they are computed only once, and will never change, no matter the successive behavior of their vehicle. To handle this difference, each patch of the representation grid does not only hold a probability value, but also two lists: the first list includes all the vehicles that have a temporary estimated probability to traverse the patch in the future, along with the value $p(C_{i,j})$ itself. The second list includes all the vehicle that have a permanent probability to have traversed the patch, again with $p(C_{i,j})$.

When a vehicle disappears from the scene, its future $p(C_{i,j})$ does not disappear with it, but instead is just moved to the permanent list. This is done because the disappearance of a vehicle is potentially a very useful event: it means that it went to a road we cannot see, and therefore we want to save the last estimation we had of the area it was heading to. It will not change anymore, if we assume the vehicle has disappeared for good. If it does reappear (i.e. the disappearance was due to a misdetection), the system will be able to retrieve its $p(C_{i,j})$ and prevent inconsistencies. After some time, all the old permanent $p(C_{i,j})$ are merged into a single value, in order to avoid an unreasonable memory burden for the system.

Finally, each patch also contains a third list, which holds all the $p_{max}(C_{i,j})$ of each vehicle and serves to compute $p(R_i|\bar{C}_i)$.

C. Integration with Direct Detection

The presented approach cannot work as a standalone system, since it fully relies on the presence of other vehicles. In order to complete it, we can incorporate a direct visual detection system, that adds to the system another source of information about the road area. Any kind of direct detection system can be incorporated, as long as it provides a Birds' Eye View (BEV) confidence map that can be translated into a probability of each patch being road. In order to merge the probabilities coming from the behavior interpretation and the direct detection, we define a *visibility* function $p(V_i)$. This function expresses the probability that the patch has been seen correctly by the direct detection. As such, it will have higher values on the center of the field of view and in the vicinity of the ego-vehicle, and it will be 0 outside of it, or behind occlusions. Having defined a visibility function, the road probability formula (1) becomes the following:

$$p(R_i) = p(R_i|\bar{C}_i) + \left[p(R|C) - p(R_i|\bar{C}_i)\right] p(C_i) \left[1 - p(V_i)\right] + \\ + \left[p(R_i|V_i) - p(R_i|\bar{C}_i)\right] p_{max}(V_i) \qquad (6)$$

Here $p(R_i|V_i)$ is the weighted average of the confidence values provided by the direct detection system, which we interpret as the probability that the patch is road, given that it has been seen correctly, and $p_{max}(V_i)$ is the maximum visibility ever assigned to that patch. Note that the formula should have $p(R_i|\bar{C}_i, \bar{V}_i)$ instead of $p(R_i|\bar{C}_i)$, but the two terms are equal, since we assume that the event of a patch not being seen correctly is independent from it being road or not. The term $p(R_i|V_i)$ is weighted using the $p(V_i)$ associated to each measurement. This formula can be interpreted as a trade-off between direct and indirect system, modulated by $p(V_i)$: if the visibility is high, the biggest contributor will be the direct detector, while if it is low, the behavior interpretation will take over.

The visibility function assigns values to all the patches inside the direct detection output area (which is a rectangle in front of the ego-vehicle) and 0 to all the patches outside of it. The 2D function that assigns the visibility is:

$$p(V_i) = \cos\left(D\frac{X_{i,e}}{b}\right) \cos\left(F\frac{Z_{i,e}}{2l}\right) \qquad (7)$$

Where the rectangle has a width of $2b$ and a length of $2l$, $[X_{i,e}, Z_{i,e}]$ is the position of patch i in the reference system of the ego-vehicle, and M, N are two constant parameters. A contour plot of this function can be seen in figure 1b. It is important to note that the domain of the visibility function does not start at $Z_i = 0$, as the camera field of view does not include the area immediately in front of the car. After the function is applied, the systems checks for possible occlusions, and all the patches that are found to lie behind an occlusions are set back to $p(V_i) = 0$.

IV. Evaluation

A. Building Ground Truth

In order to evaluate the performance of our system, we recorded real world scenes with a mobile platform. The platform is equipped with 6 Ibeo Lux Lidars including Ibeo Lux Fusion System for 360 degree tracking (see figure 2) and an IDS UI-3580 camera in the front windshield. The results that will be shown in section IV-B refer to a stream of over 2 minutes, part of a larger recording. In this stream our mobile platform makes a full lap around a block, occasionally stopping due to other cars. The recordings have been made in the inner city of Offenbach, Germany.

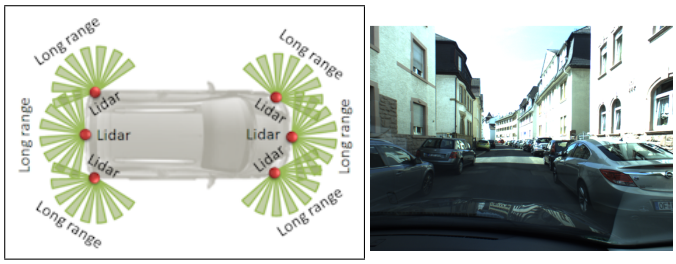


Fig. 2: This figure shows the Lidar setup of our mobile platform (left) and a frame of the stream, as example (right)

In order to get a ground truth to compare our systems representation against, we hand-labeled the road area in an amount of frames necessary to get a good coverage of the whole stream, and then built a grid map, with the same size as the representation our system will create, by mapping every labeled frame onto it. The mapping has been performed by firstly transforming every image from perspective into BEV, and then placing it with the proper position and orientation on the grid. Every square whose center location has been annotated as part of the road for at least two frames, is considered true road area. The placement onto the grid has been made by estimating the motion of the mobile platform using vehicle CAN data. However, the estimation of the trajectory, based on these data, is far from optimal. As a result, after a full lap the placement of the car is off by 1-2m. As this method is the same we used to build our representation grid, the two representations will be properly aligned, as long as we run our tests only on the same lap we built the ground truth with.

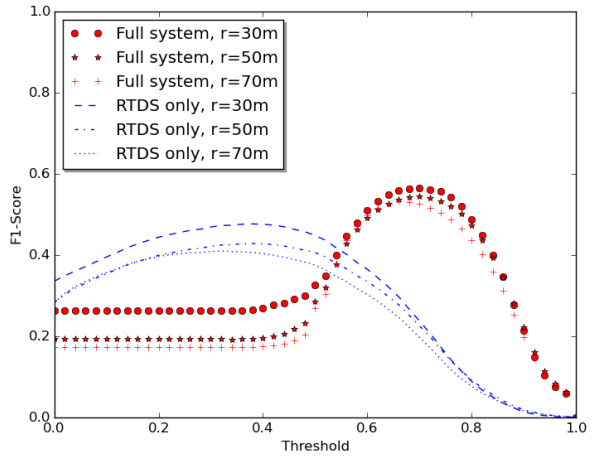
We chose this method for multiple reasons. First of all, our representation cannot be compared to the ground truth frame by frame, like it is usually done for evaluating visual direct detection systems, because road areas that are behind occlusions are relevant for our approach, and in a single frame they cannot be annotated properly. Also, road area that lies outside the field of view can be detected by our approach, as it can detect vehicles at 360-degrees, and of course that cannot be compared with a single-frame ground truth. The evaluation has been made by comparing the performances of our full system with the ones of the direct detection system it uses, the RTDS[10], trained on 80 images for the first stage and other 80 for the second. The images used for the training came from a different area of the same recording. For the sake of this comparison, we had to evaluate the RTDS on the same ground truth grid we presented. In order to do so, we chose to evaluate the average of the RTDS confidence over all the past frames, instead of the standard way of evaluating the single-frame output. Taking into account the average of all past frames also enables the RTDS to detect road area behind the ego-vehicle, by simply keeping it detected from past frames.

B. Results

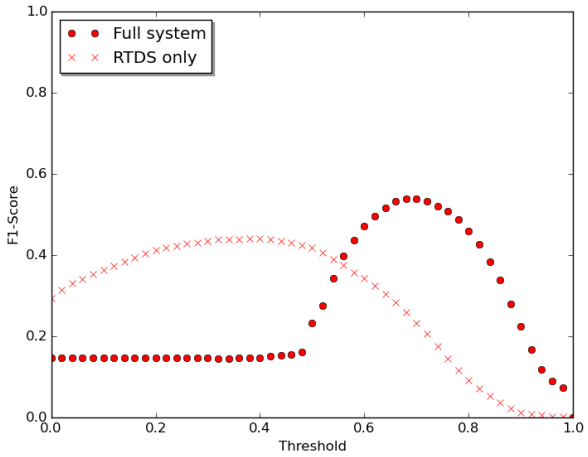
For this test we chose a representation grid of 200m x 250m, with 0.5m x 0.5m squares. The size of the representation has been chosen large enough to cover the whole block and some of the external roads. It is important to note that a few roads in the ground truth are beyond the scope of the RTDS, as the ego-vehicle never drives on them, while they can be detected by the behavior interpretation instead.

For this experiment we decided to rescale the confidence levels of the RTDS input from $[0, 1]$ to $[0.5, 1]$. This choice is dictated by the fact that our current system can only detect occlusions due to other vehicles, and not other obstacles, like buildings. Therefore, the visibility value in correspondence of those occlusions would give the RTDS too much weight in our system. By rescaling the confidence levels, we make the RTDS only contribute to the positive side of the $road(1) \leftrightarrow nonroad(0)$ dichotomy, that in this case becomes $road(1) \leftrightarrow unknown(0.5)$. The result of this arrangement is that our system will exhibit higher values with respect to the ones provided by the RTDS alone, and consequentially it will have higher effective thresholds. Our full-fledged system will use the original confidence interval, and will therefore be able to distinguish non-road with both behavior interpretation and direct detection.

Both configurations (full system and RTDS only) have been run offline and the evaluation has been performed by comparing the output representation every 10 frames with the ground truth, setting a variable threshold that maps uniformly the interval $[0, 1]$, with a step of 0.02, for a total of 51 thresholds: a patch with a road probability higher than the threshold is considered road, while it is considered non-road in the opposite case. The classification on each of the considered frames was then compared patch-wise with the ground truth, thus building a confusion matrix. We chose to



(a) F1-score plot within different ranges



(b) F1-score plot of the representations at the end of the stream

Fig. 3: F1-score plots of the RTDS and our full system.

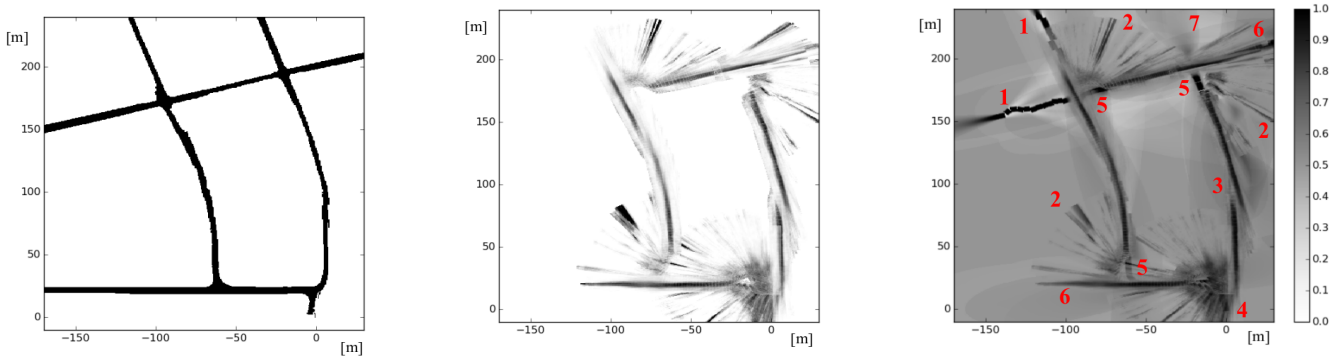
evaluate the performance by computing the F1-score, which is the harmonic mean between precision and recall.

Figure 3a shows the average performances of the two configurations over the whole stream, comparing for every frame the internal representation of each system versus the ground truth only inside a limited radius from the ego-car, and then averaging the F1-score over all the considered frames. The choice of evaluating the representation only in a limited radius has been made because evaluating over the whole map would be unfair with respect to the early frames, when both configurations do not have enough data to represent the whole area. From this evaluation it can be seen that the full system has a significantly higher performance than the RTDS alone. Additionally, the comparison within different ranges shows the increased spatial range of our full system, which within 70m exhibits performances superior to what RTDS does within 30m. Note that, as expected, the threshold at which the full system works best is much higher than the one of RTDS.

Figure 4 shows a comparison between the ground truth and the representation of the whole block obtained at the end of the stream by the two configurations. This representation has been evaluated along the whole map, instead of inside a radius. The F1-score of these representation can be seen in figure 3b. A few important areas of the final representation of our full system (figure 4c) have been marked with numbers: the two roads marked with (1) could not be detected by a direct system at all, but the vehicles moving on them enabled the behavior interpretation to do so. The areas marked with (2) are artifacts caused by the RTDS, and are due to its limited training set and challenging imaging conditions. Those artifacts lie in correspondence of buildings, and the full system would correct them easily if it could detect buildings as obstacles. The discontinuity marked with (3) is a bend where particularly unfavorable imaging conditions and the absence of other vehicles prevented the system to see the missing segment. The intersection marked with (4) is the starting position of the ego-vehicle, which starts at the position $(0, 0)$. The many false positives presented by the RTDS in the vicinity are caused by the maneuver that the ego-vehicle made in order to enter the traffic from its parking spot. Marked with (5) are three intersections where the presence of a vehicle in front of us allowed the behavior interpretation to partially fill the discontinuity caused by the weak performance of the RTDS, that did not detect road accurately in all intersections, as seen in figure 4b. The RTDS did, however, detect the external roads marked with (6), as they were clear from traffic and placed in front of the ego-vehicle. The full system failed to properly detect only one road, marked with (7). This road lied behind an occlusion, so that it could not be directly detected, while the only observed vehicle coming from it was moving at a very low speed, and as a result only a hint of road can be seen in the representation. The comparison also shows the different range of values that the two configurations output: the main difference is that, while the RTDS simply gives 0 in absence of information, our full system gives 0.5. This is a potential benefit of our system, as it is not bound to a rigid classification between road and not road like the RTDS, but it can classify areas as ”unknown”.

V. Conclusions

We presented a behavior interpretation approach that detects road area by estimating future and past position of the other vehicles in the scene. This approach works on a larger system that couples it with a direct visual road detection system. We showed the potential benefit of our system by evaluating its F1-score on a real-world stream and comparing it against a purely direct detection method. The evaluation demonstrates that our approach has better performance and a larger spatial range, thus being a useful tool for ADAS. Next research will focus on improving our method to extract ground truth from the



(a) Ground truth

(b) RTDS average confidence

(c) Full system representation

Fig. 4: Comparison between ground truth and final representation of the two configurations.

images so that it is possible to test on longer streams. In order to do so we will need a better ego-car motion approximation or a precise self-localization system (e.g. DGPS). Another important step will be to extend our occlusions detection with the help of the lidar, to be able to set the visibility to 0 behind any kind of obstacle and thus be able to explicitly use the RTDS non-road detection. Further attention will then be dedicated to couple our approach with an external trajectory planner, in order to prove the benefit of the extended range on a concrete application.

References

[1] Andreas Alin, Martin V. Butz, and Jannik Fritsch. Tracking moving vehicles using an advanced grid-based Bayesian filter approach. *2011 IEEE Intell. Veh. Symp.*, (Iv):466–472, 2011.

[2] Jose M. Alvarez, Antonio M. Lopez, Theo Gevers, and Felipe Lumbreras. Combining Priors, Appearance, and Context for Road Detection. *IEEE Trans. Intell. Transp. Syst.*, 15(3):1168–1178, 2014.

[3] Andrew Bacha, Cheryl Bauman, Ruel Faruque, ..., David Van Covern, and Mike Webster. Odin: Team VictorTango's entry in the DARPA Urban Challenge. *J. F. Robot.*, 25(8):467–492, 2008.

[4] Hernan Badino, Uwe Franke, and Rudolf Mester. Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming. In *ICCV 2007 Work. Dyn. Vis.*, Rio de Janeiro, Brazil, 2007.

[5] Sarah Bonnin, Thomas Weisswange, Franz Kummert, and Jens Schmüdderich. Pedestrian Crossing Prediction using Multiple Context-based Models. In *17th Int. IEEE Conf. Intell. Transp. Syst.* IEEE, 2014.

[6] Mohamed Essayed Bouzouraa and Ulrich Hofmann. Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors. In *IEEE Intell. Veh. Symp. (IV 2010)*, pages 294–300, San Diego, USA, 2010. IEEE.

[7] Edoardo Casapietra, Thomas Weisswange, Christian Goerick, Franz Kummert, and Jannik Fritsch. Building a probabilistic grid-based road representation from direct and indirect visual cues. In *IEEE Intelligent Vehicles Symposium (IV 2015)*, pages 273–279. IEEE, 2015.

[8] Radu Danescu, Florin Oniga, and Sergiu Nedevschi. Modeling and Tracking the Driving Environment With a Particle-Based Occupancy Grid. *IEEE Trans. Intell. Transp. Syst.*, 12(4):1331–1342, 2011.

[9] Maryam Fatemi, Lars Hammarstrand, Lennart Svensson, and Ángel F García-Fernández. Road Geometry Estimation Using a Precise Clothoid Road Model and Observations of Moving Vehicles. In *17th IEEE Int. Conf. Intell. Transp. Syst. (ITSC 2014)*, pages 238–244, Qingdao, China, 2014. IEEE.

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

[11] A Geiger, P Lenz, and R Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR 2012)*, pages 3354–3361. IEEE, 2012.

[12] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3D Traffic Scene Understanding from Movable Platforms. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–14, 2013.

[13] Tobias Gindele, Sebastian Brechtel, Joachim Schröder, and Rüdiger Dillmann. Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge. pages 669–676, 2009.

[14] Jungwon Kang and Myung Jin Chung. Stereo-vision based free space and obstacle detection with structural and traversability analysis using probabilistic volume polar grid map. *2011 IEEE 5th Int. Conf. Robot. Autom. Mechatronics*, pages 245–251, 2011.

[15] Dan Levi, Noa Garnett, and Ethan Fetaya. Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 109.1–109.12. BMVA Press, 2015.

[16] Martin Liebner, Michael Baumann, Felix Klanner, and Christoph Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *IEEE Intell. Veh. Symp. (IV 2012)*, pages 1162–1167, Alcalá de Henares, Spain, 2012. Ieee.

[17] 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.

[18] Simon T. O'Callaghan, Surya P. N. Singh, Alen Alempijevic, and Fabio T. Ramos. Learning navigational maps by observing human motion patterns. In *IEEE Int. Conf. Robot. Autom. (ICRA 2011)*, pages 4333–4340, Shanghai, China, 2011. IEEE.

[19] Anna Petrovskaya and Sebastian Thrun. Model based vehicle detection and tracking for autonomous urban driving. In *Autonomous Robots*, pages 123–129, 2009.

[20] Sayanan Sivaraman and Mohan Manubhai Trivedi. Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance. *IEEE Intell. Transp. Syst. Mag.*, pages 1–11, 2014.

[21] 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.

[22] Thorsten Weiss, Bruno Schiele, and Klaus Dietmayer. Robust Driving Path Detection in Urban and Highway Scenarios Using a Laser Scanner and Online Occupancy Grids. In *IEEE Intell. Veh. Symp. (IV 2007)*, pages 184–189, Istanbul, Turkey, 2007. IEEE.

[23] Thomas H. Weisswange, Bram Bolder, Jannik Fritsch, Stephan Hasler, and Christian Goerick. An integrated ADAS for assessing risky situations in urban driving. In *Intell. Veh. Symp. (IV), 2013 IEEE*, pages 292–297, 2013.

[24] Liang Xiao, Bin Dai, Daxue Liu, Tingbo Hu, and Tao Wu. Crf based road detection with multi-sensor fusion. In *IEEE Intelligent Vehicles Symposium (IV 2015)*, pages 192–198. IEEE, 2015.