

Towards a unified architecture for mapping static environments

Tim Kubertschak and Mirko Maehlich
AUDI AG
I/EE-31
85045 Ingolstadt, Germany
Email: tim.kubertschak@audi.de

Hans-Joachim Wuensche
Universität der Bundeswehr München
Technik Autononomer Systeme (TAS)
85577 Neubiberg, Germany

Abstract—Sensor fusion systems are becoming more and more important in automotive applications, especially for future driver assistance systems and autonomous driving. While it was sufficient to use single sensors in early driving assistance systems, future systems will rely on several sensors with different measurement principles. The probably conflicting measurements need to be fused to retrieve a complete picture of the current surroundings. However, an increased number of sensors leads to difficulties while incorporating their measurements, since every sensor provides its data via special interfaces. This inconvenience of current sensor fusion architectures has been solved for moving objects. A fusion architecture based on a generic interface – the object list – has been developed over the last years. It allows an easy use of additional sensors as long as they provide their measurements as object lists. Similar solutions for static environments are rare, but the recent proposal of the *Fences*-approach [1] is promising.

This work shows, how sensors and occupancy grids are represented in the *Fences*-architecture. Steps and algorithms are presented to transform sensor reading into *Fences* as well as extracting *Fences* from common grid-based strategies for mapping static environments. The algorithms are applied to the conversion of ultrasonic sensor readings and LiDAR measurements to *Fences*. Furthermore, the construction of *Fences* from Bayesian occupancy grids is presented.

Keywords—Advanced Driving Assistance Systems, Multisensor Data Fusion, Static Environment, Unified Architecture

I. INTRODUCTION

Sensor fusion systems are becoming an integral part of future vehicles. Many currently available advanced driving assistance functions and almost all future driver assistance functions build upon several sensors to ensure a reliable support of the driver of the vehicle. Since many sensors utilize different physical principles to observe the environment, some kind of data fusion must be applied to simultaneously use their provided information, i.e. several rules must be implemented to augment single sensor readings or dissolve conflicting measurements.

Among the first driving assistance functions that have been brought to market, basic crash prevention [2] and adaptive cruise control [3] were the first who evolved towards an integration of several different sensors. Current systems for longitudinal and lateral guidance rely on objects that are detected primarily by video sensors and different kinds of scanning radars. Since those systems support the driver in

highly dynamic situations, they usually provide information of moving objects in the surroundings of the vehicle and fusion architectures are tailored towards this demand.

However, recent development in driving assistance functions are targeting in another direction. To achieve fully autonomous driving, a representation of the whole environment is required, i.e. dynamic and static objects. First driving assistance functions have been developed that actively use information of the static environment. The functions are driving path detection [4], parking assistants [5] and others. They require information about the location of buildings, pavements, parked cars, traffic signs and light poles and all kind of road markings.

Building representations of the static environment is tedious for automotive and industrial applications. Since the subject is quite new to these areas, there exists no consensus about common representations, interfaces and architectures as they exist for moving objects. Every sensor provides its data in unique styles and every driving assistance function needs to know other details of the static surroundings of the vehicle in different representations. In current mapping architectures, a conversion between all representations must be done. Thus, integrating several sensors and driving assistance functions is laborious since in general, another conversion routine needs to be implemented for each sensor and function.

To overcome the difficulties with current architectures for static environments, the *Fences*-architecture has recently been proposed [1]. This architecture unifies the interfaces between sensors, the mapping subsystem and subsequent functions. As a side-effect, the participating subsystems and conversion modules have been rearranged to simplify integrating multiple sensors and driving assistance functions. However, [1] just presented the motivation for *Fences*, its changes to the conventional mapping techniques and properties and benefits of using it. It has not been shown how to create the topological representation from single sensor readings and from the results of sensor data fusion.

The mentioned parts that are missing in [1] are detailed in this work. Mechanisms and algorithms are presented that demonstrate the conversion between sensor readings and the *Fences*-representation. In particular, the transformation from ultrasonic sensor echoes to *Fences* and from scanning LiDAR measurements to *Fences* is presented. The difficulties and caveats of the conversion-process are discussed to emphasize

the advantages of using *Fences* over the conventional process. Furthermore, an algorithm for turning a grid-based representation of the static environment into the topological *Fences*-representation is presented to provide an unified interface for all driving assistance functions.

The paper is organized as follows. Section II presents related work of other authors regarding generic and unified architectures. Section III discusses the conventional architecture for mapping along with its disadvantages and gives a short overview of *Fences*. The conversion from sensor readings and grid-based maps of the environment to the topological representation is demonstrated in section IV and V. Finally, some concluding remarks are given in section VI.

II. RELATED WORK

Unified and generic architectures for fusion systems haven't been of much interest in the past. In most of the previous works it was rather a tool to demonstrate other ideas. Furthermore, most of the authors separated the interface between sensors and sensor fusion as well as between sensor fusion and subsequent functionality, thus not proposing a unified architecture.

First steps towards unifying sensor interfaces have been performed by Matthies and Elfes [6] in the framework of occupancy grids. Due to their rules for fusing different sensors, every sensor reading is transformed to small occupancy grids that are fused afterwards. Thus, they provided a simple generic interface for sensors. Chatila and Laumond [7] proposed a geometric approach for building maps and implicitly required a polygonal representation from every sensor. A taxonomy of different sensor classes was defined by Coué et al. [8] based on their physical characteristics. During the first decade of the current century, a generic representation for moving objects has been developed for automotive use. Those object lists ([9]) consists of quite general information like position, uncertainties and classes, but can easily be tailored to fit special needs. Contrary to this development, Scheunert et al. [10] proposed a grid-based interface as common representation for moving objects and interface for their tracking system.

Generic interfaces between the fusion system and subsequent functions have not been considered so far. However, Thrun [11] proposed to use topological representations on top of fused maps of the environment to simplify and accelerate the task of navigating a robot through known environments. This approach is close to the one proposed by *Fences*.

The first generic architecture for static environments has been proposed by Grewe et al. [12]. They adapted the idea of Matthies and Elfes and used occupancy grids as generic interface between all parts of the fusion system. To avoid the high memory consumption of this representation, in particular due to sensors with a huge field of view, they further proposed to run-length-encode the local occupancy grids. Besides the different styles of interfaces, their approach is closest to *Fences*.

III. MAPPING

This section reviews the conventional architecture for static environment mapping, its difficulties and the improvements

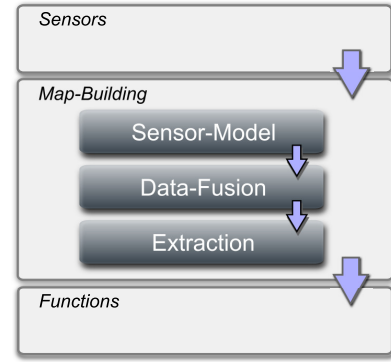


Fig. 1. Conventional architecture for mapping static environments.

made by *Fences* along with its properties and benefits of using it.

A. Conventional Architecture

The conventional architecture as used by [6] and [13] is depicted in figure 1. It basically consists of three main subsystems: sensors, map-building and subsequent functions. Sensors provide measurements, map-building combines these measurements to build a representation of the current environment and the functions use this representation to implement certain features. Each subsystem performs additional tasks but only the most important ones are shown.

Sensors and driving assistance functions are both connected to the map building-subsystem. Each sensor provides its measurements in some special way. Ultrasonic transducers provide their measurements as multiple echoes, while scanning LiDARs supply a distance measurement for every "beam" that emerges from it. Other sensors use still other representations to provide their measurements. The same issue arises at the other end of the whole process. Driving assistance functions require in general very specific information about the environment. For example, parking assistants need parking lots in the vicinity of the vehicle and a trajectory to get there. On the other hand, driving path detecting uses elongated objects like guard railings, buildings and groups of guiding posts and pillars to predict a possible trajectory of the vehicle.

Since each sensor provides its measurements in very specific representations they can't be fused directly. Each measurement must be transformed in order to build a map of the surroundings of the vehicle. This transformation uses a sensor model that is unique for each type of sensor and even for the same type of sensor from different manufacturers. Building this sensor model is a task of the map-building subsystem in today's mapping architectures. However, in order to build a reliable and accurate sensor model very detailed knowledge about the sensor's physical characteristics and internal signal processing is necessary. While the underlying physics is well known, the signal processing is a crucial part of the manufacturers intellectual property. Since this knowledge is in general not available, the sensor model is usually a rough approximation of the sensor's real behavior.

The extraction of information for driving assistance functions is currently a task of the map-building subsystem as well. The task is equally demanding, because an extractor must be

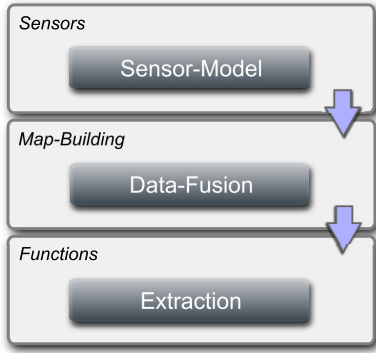


Fig. 2. Unified architecture for mapping static environments.

provided for every function. Each extractor must incorporate additional information that is related to a certain functionality and usually not accurately specified. The provided information is thus incomplete and of limited use.

In summary, conventional architectures spend a lot of time transforming between several representations. The results are usually crude since very specific information is necessary that is often not available. Incorporating additional sensors and subsequent functions is laborious, because further sensor models and extractors must be implemented.

B. Unified Architecture

To overcome the difficulties of conventional mapping architectures as described in the last section, Kubertschak et al. [1] presented the *Fences* approach for mapping static environments. They proposed to use the same interface between sensors and map-building and map-building and functions which results in a fully unified architecture. The homogeneous interface allows to relocate some tasks performed by each subsystem and thus simplifying the process of fusing several sensors. In particular, all tasks that require detailed knowledge about sensors and driving assistance functions have been moved to their related subsystems. Figure 2 shows a diagram of the *Fences*-architecture along with the new location of each task.

A unification of the interfaces of all subsystems is possible, because all sensors and functions basically exchange the same information. Each sensor provides information about objects and regions of their field of view that may be assumed to be free. On the other hand, driving assistance functions need to know which parts of the environment may be safely traversed and which parts are occupied by obstacles. For example, ultrasonic sensors provide information about the location of objects with compatible physical characteristics. Simultaneously, the region between the sensor's origin and an object may be assumed to be free. Otherwise the object could not have been detected. As an example for driving assistance functions, parking assistants require regions near the vehicle that are shaped like a parking lot. That are parts of the environment that fully contain free space and may be limited by obstacles, road markings or the field of view. In addition, these regions must be connected to the vehicle by free space in order to plan a secure path to the parking lot. The same arguments count for ranging LiDARs, video sensors and other

sensors as well as obstacle avoidance, driving path detection and other functions.

To efficiently describe free spaces and obstacles a topological representation extended with a set of certain attributes was proposed in [1]. All entities of the static environment are represented by *contours* that are composed of several significant features called *vertices*. Features that are connected in the environment are connected in the representation as well. Contours are used to describe connected regions that are either fully or only in parts known. Fully known regions like free spaces, holes and ditches are described by polygons. All other regions like buildings, cars and elevations are represented by polylines if only parts of them are known. The vertices are used to describe the shape of the contours. To do so, the topological representation must be converted to a geometrical one. This is achieved by extending each vertex with a position along with uncertainties that arise due to imperfect sensor readings. The uncertainties are twofold: a positional uncertainty to account for inaccuracies of the position and existential uncertainties to account for errors due to the physical characteristics of the detected objects. Figure 8 shows instances of the topological representation. As can be seen, they look almost like a set of real fences with irregularly distributed poles. The homogeneous interface and the unified architecture is therefore called *Fences*.

Using the proposed architecture has several advantages. The first has already been sketched in the previous paragraphs: Moving the sensor and function-specific parts to their corresponding subsystem leads to improved and more accurate sensor-models and extractors. In order to make them as accurate as possible the manufacturers may incorporate all available knowledge – the knowledge that is publicly available and the knowledge that is part of the intellectual property of the manufacturer. A second benefit evolves from the introduction of homogeneous interfaces. Every subsystem is now independent from each other and can thus be added, replaced or removed unnoticed. Additional sensors and functionality can be integrated without any additional efforts. The map-building component is already able to fuse sensor-readings in *Fences*-representation and provides information about obstacles and free spaces in the close proximity of the vehicle as *Fences*. It is even possible to silently replace the mapping strategy. Some instances of the fusion system may use metrical maps like [6], other instances may use topological maps like [7]. Even other instances of the fusion system may completely remove the map-building component. Especially in low-cost applications it is possible to plug sensors directly to functions. These situation are further supported by another property: The architecture can easily adapt to bandwidth limitations. Certain methods can be applied to reduce the size of the topological representation in exchange for accuracy.

IV. SENSOR-MODELS FOR *Fences*

This section and the following two subsections discuss several aspects of building adequate sensor-models to transform sensor readings to the desired topological representation. There are a number of steps that must be performed for each sensor. These steps are defined first and applied to frequently used sensors afterwards. Since there are a lot of different sensors used today in robotic and automotive applications, two of them

```

1: procedure SENSORMODEL(m)
2:   s ← DETERMINEFIELDOFVIEW(m)
3:   u ← DETERMINEUNCERTAINTIES(s, m)
4:   f ← CREATEFENCESREPRESENTATION(s, u)
5:   return f
6: end procedure

```

Fig. 3. A procedure to convert sensor readings to *Fences*.

are exemplary chosen and the steps to create their sensor-models are detailed. The chosen sensors are ultrasonic sensors and ranging LiDARs due to their popularity and importance in current applications.

Before describing the steps to construct models for certain sensors, a specific instance of the *Fences*-representation is defined. Since the topological representation summarized in section III-B does not contain any further details for describing the environment, a set of attributes for contours and vertices must be specified. The *Fences*-instance used here has the same attributes as proposed in [1]. These attributes are:

- position as 2D Cartesian coordinates
- positional uncertainty
- obstacle or field of view classification
- existential uncertainty

for vertices and

- polyline contour for obstacles
- polygonal contour for free spaces

for contours.

The particular steps that must be carried out to build a sensor-model for a specific sensor are shown in figure 3. The input to the procedure are measurements **m**. The first step is to determine the field of view of the sensor. This is the area in the positive half-plane of the sensor, where the sensor is in general able to detect objects and possibly free spaces. The shape of this area depends highly on the physical characteristics of the utilized carrier and the signal processing that is applied. The resulting shape **s** is further altered by applying **m**, i.e. free spaces are somehow limited and obstacles are defined. The next task is to assign existential uncertainties **u** to each position of **s**. The certainty of free spaces should be chosen as pessimistic as possible, since the region that can safely be traversed is defined by them and false negatives in free space can lead to severe damage. On the other hand, the certainty for obstacles and objects should be chosen optimistic to ensure a reliable detection of all objects and minimize the chance of false negatives as much as possible. However, there are objects for all kind of sensors that cannot be detected due to unfavorable physical properties. The existential uncertainties must account for these objects as well. The final step of the procedure is to create *Fences* **f** from the shape **s** and uncertainties **u**. The shapes of free spaces and obstacles are converted directly to contours of type polygon and polyline, respectively. The field of existential uncertainties is applied to the vertices of the contours only. That is, the uncertainty that is assigned to each vertex is the one at the location of the vertex. This behavior seems odd, because most of the information

about uncertainties is lost. But as the examples in the next two subsections will show, the information is sufficient since the remaining uncertainties can be determined by interpolation from the available information.

The steps given in figure 3 specify only a coarse procedure. When implementing each step for a particular sensor, its specific characteristics must be considered to make the sensor-model as accurate as possible. These characteristics evolve from the measurement principle and from the signal processing that is applied while receiving the signal and is special to each sensor.

A. Ultrasonic Sensors

Ultrasonic or sonar sensors are one of the most frequently used sensors. Their application in the robotic and automotive field reaches back to the beginning of the development of robotics and intelligent vehicles. However, they have some very special properties that make it difficult to work with them and to derive correct interpretations from their measurements.

Sonar sensors use ultrasound to detect certain entities in front of them. Ultrasound is a sound pressure wave with a frequency that is above the limit of 20 kHz of the human hearing range. These waves spread primarily as longitudinal waves with equal intensity in all directions and are thus forming a spherical wave. Once the spherical wave reaches an object, it is in parts reflected, refracted and transmitted. If the object has a favorable shape and favorable physical properties, i.e. small or scattered surfaces that are perpendicular to the sensor, the sound wave may return to the sensor. The distance between the object and the sensor's origin can then be retrieved from the travelling time of the sound wave. Ultrasonic sensors thus use a time-of-flight principle to measure the distance to objects. If the emitted sound wave does not return to the sender, there is either no object in front of the sensor or there are objects that cannot reflect the sound wave back to the sensor.

DETERMINE FIELD OF VIEW. The discussion of the preceding paragraph indicates that the detection of objects with sonar sensors is based on the physics of sound waves. Following the theory of acoustics, a spreading ultrasound wave is described by local changes in pressure of its carrier medium. The sound pressure decreases with an increasing distance from the sensor and vanishes once the sound wave has reached a given distance. Objects can only be detected if they are reachable by sound waves, i.e. within the area of positive sound pressure. The sound pressure field is thus a good choice to determine the field of view of an ultrasonic sensor.

Since sonar sensors are using an oscillating membrane of radius R to generate ultrasound waves, they are best modeled as piston radiator. The sound pressure field of piston radiators consists of two parts: an intensity part p and a directivity part Γ . According to [14] they are described mathematically by equations 1 and 2.

$$p(r) \approx 2\rho_0 c v_0 \cdot \sin\left(\frac{kR^2}{4r}\right) \quad (1)$$

$$\Gamma(\varphi) = \frac{2 \cdot J_1(kR \cdot \sin \varphi)}{kR \cdot \sin \varphi} \quad (2)$$

In these equations, ρ_0 is the static pressure, c is the speed of sound, v_0 the sound particle velocity of the carrier medium

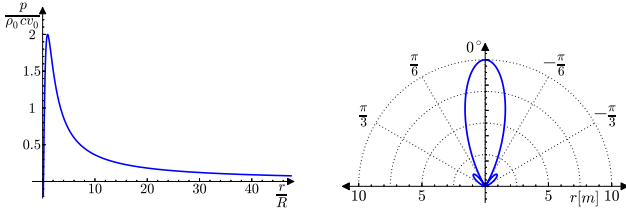


Fig. 4. Sound pressure intensity (left) and directivity (right) of an ultrasonic sensor with $R = 0.01$ m, $f = 40$ kHz using air as carrier medium.

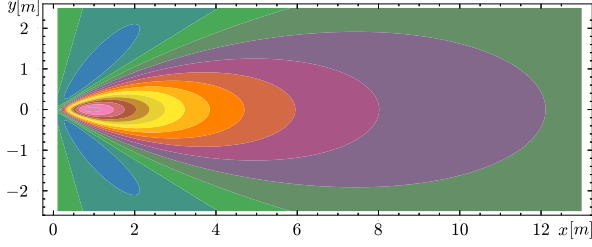


Fig. 5. Contour plot of the sound pressure field of equation 3. The acoustic source is located at the origin.

(i.e. air) and k is the wave number of the sensor's frequency. J_1 is the Bessel function of first kind and order one. Plots of both equations are shown in figure 4. The left image shows the normalized intensity along the principal axis of a sensor with a membrane of radius 1 cm, frequency 40 kHz and air as carrier medium ($\rho_0 = 1.2041 \frac{\text{kg}}{\text{m}^3}$, $c = 344 \frac{\text{m}}{\text{s}}$, $v_0 = 5 \cdot 10^{-8} \frac{\text{m}}{\text{s}}$). The right image shows the corresponding directivity, which is the iso-contour of the pressure-level at a distance of 10 m from the sensor.

DETERMINE UNCERTAINTIES. Existential uncertainties describe the probability that a detected object exists in reality and the probability that free spaces are not occupied by objects. The uncertainty is thus the probability of the events object exists and free space exists. To determine the uncertainties at every position inside the sensor's field of view, the sound pressure field is used again. Combining the intensity and directivity from equations 1 and 2 yields:

$$\begin{aligned} p(r, \varphi) &= p(r) \cdot \Gamma(\varphi) \\ &\approx 4\rho_0 c v_0 \cdot \sin\left(\frac{kR^2}{4r}\right) \cdot \frac{J_1(kR \cdot \sin \varphi)}{kR \cdot \sin \varphi} \end{aligned} \quad (3)$$

A contour plot of equation 3 is shown in figure 5. According to the figures, the sound pressure is very high close to the sensor and decreases nonlinearly with increasing distance from the sensor and increasing angle from the medial axis.

The justification for using the sound pressure field as basis for existential uncertainties follows from the fact that it is more likely to have free spaces in reality, if there is enough energy of the sound wave left to return to the sender. That is, the less energy is left at some location, the more likely it is to miss some objects because the sound wave can't reach the sensor due to starving. Similar arguments apply to obstacles that have successfully been detected. Since ultrasound waves spread as spherical waves, it is impossible to retrieve the true angular location of objects. However, it is much more likely that an object is located near the medial axis of the sensor since there

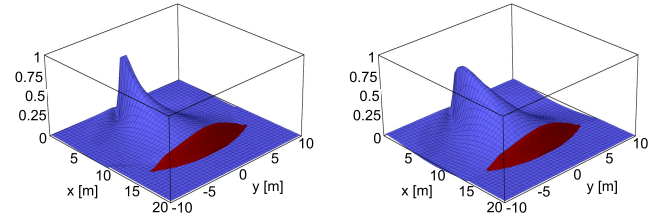


Fig. 6. Real distribution of existential uncertainties derived from the sound pressure field with an successfully detected object at a distance of 15 m (left) and its *Fences*-representation (right). A base confidence of 0.7 and 0.5 is used for free space and obstacles. The existential uncertainties of free space and obstacles are displayed in blue and red, respectively. The sensor is located at the origin, pointing along the positive X-axis.

is more energy left for the wave's way back to the sensor than near the field of view-boundary. Finally, since sound energy is directly related to sound pressure ([14]), the pressure field can be used to gather information about existential uncertainties.

In order to derive uncertainties from the sound pressure field it must be normalized. Each normalized pressure value is multiplied with a base certainty level. This base certainty level should already account for objects with unfavorable characteristics, i.e. false positives and false negatives.

CREATE FENCES REPRESENTATION. After the sensor's field of view and existential uncertainties have been determined, it is an easy task to derive *Fences* from it. The field of view is sampled regularly to derive the free space polygon. The polygon is cut at a given distance from the origin if an object has been detected. The radial line at the location of the object is sampled to create the obstacle-polyline. The existential uncertainty is calculated by applying the position of each vertex to equation 3. The resulting value is normalized and the base certainty value multiplied to it. Note that different base certainty levels for free spaces and obstacles can be used. In addition, vertices of the free space-polygon are classified as *field of view*, vertices of polylines as *obstacle*.

The resulting *Fences*-representation is depicted on the left of figure 8, a three-dimensional plot of the representation with its confidences is shown in figure 6. The ability of *Fences* to represent complex sensors is quite good, although there are some differences compared to the real uncertainties. Since the free space confidences are linearly interpolated between all vertices, they do not fit the real nonlinear distribution. However, the effect of the error can be decreased by applying two improvements. First, the number of vertices can be increased to follow the outline of free spaces and obstacles more closely. Second, instead of using a linear interpolation to retrieve the uncertainties, a higher order interpolation method can be used. In order to apply e.g. spline-interpolation, the interpolation parameters must be attached as attributed to the affected contours.

The presented method to build a sensor-model can be further extended. Today's ultrasonic sensors are able to detect several objects in their field of view as long as they have different distances from the sensor. Each object must be appropriately modeled. Another property of sonar sensors that hasn't been accounted for are specular reflections. An appropriate approach has been described by Lim and Cho in [15]. At last, effects of the applied signal processing must

be modeled. However, these effects can only be modeled by the manufacturer, since the signal processing is part of their intellectual property.

B. Ranging LiDARs

Laser range finders have evolved to a mature sensor-technology that has found its way to automotive applications. This development has happened due to the sensor's beneficial properties. LiDARs use focused light that is in general outside of the human visible range to detect objects in front of them. Once the light beam reaches an object it is reflected back to the sensor. The distance of the object is determined from the time the light beam was travelling.

However, a single laser beam covers only a small area. In order to detect objects across a wide area LiDARs use several consecutive laser beams, each covering another part of the sensor's field of view. If several light beams are not returning to the sender, then there are two cases. There is either no object that could have reflected the laser light or there is an object with unfavorable physical characteristics that cannot return the light beam to the sensor, e.g. shiny black objects. Especially the last case must be considered while converting to *Fences*.

DETERMINE FIELD OF VIEW. Determining the field of view of LiDARs is quite easy compared to ultrasonic sensors. Since the sensor uses very focused beams of light, there are almost no restrictions. In theory, it is possible to detect objects that are several hundreds of meters away from the sensor. The maximum possible distance is limited by the length of the receiving window only. That is, the more time is used to listen for incoming signals, the more distant objects can be detected. Since LiDARs scan the area in front of them in regular intervals, not every location can in general be sampled. The ability depends highly on the characteristics of the sensor, i.e. how much each beam is defocused with increasing distance from the sensor. However, this effect can be neglected, because it is considered during the determination of existential uncertainties. So each laser beam can be connected with its neighbours to determine a description of the field of view.

In summary, the field of view of a ranging LiDAR is a circular sector of the same angle as the opening angle of the sensor. This is true for the environment and the sensor being static. Otherwise the circular sector degrades due to the motion scan effect. This degradation is not further regarded here, since only the basic steps described in section IV are demonstrated.

DETERMINE UNCERTAINTIES. While determining the existential uncertainties, several aspects must be considered. As Weiss et al. [16] noticed, the certainty for free space must decrease linearly with increasing distance from the sensor. The degradation follows from the radial expansion of the laser beams. With increasing distance from the sensor, some small objects might be missed and thus the certainty must be decreased. The linear degradation already accounts for assumptions made during the determination of the field of view. Bouzouraa [17] noticed some similar effects for objects. They might appear larger as they are in reality, especially if the objects are very shiny. Since nothing is known in advance about the shininess of an object, its outer parts should express an increased existential uncertainty compared to the inner

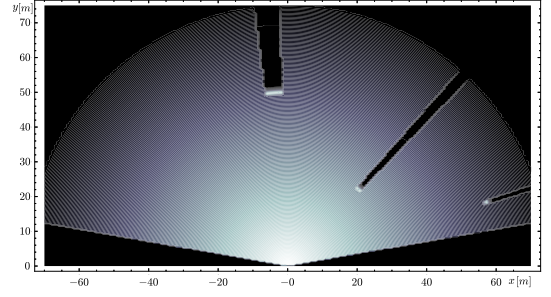


Fig. 7. Distribution of existential uncertainties of a ranging LiDAR with a maximum field of view of 75 m, an opening angle of 160° and three obstacles. The sensor is located at the origin.

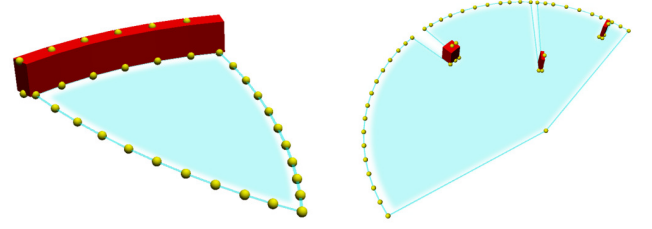


Fig. 8. *Fences*-representation of ultrasonic sensors (left) and ranging LiDARs (right) as described in sections IV-A and IV-B. The cyan and red contours stand for free spaces and obstacles, respectively. Vertices are represented by yellow spheres. The positional uncertainty is illustrated by the width of each line. Existential uncertainties are omitted for clarity. Each sensor is located near the lower right vertex.

parts. The resulting distribution of existential uncertainties is shown in figure 7.

CREATE FENCES REPRESENTATION. The conversion of raw LiDAR measurements to fences is again straightforward. A vertex is created for each laser beam. The vertices of consecutive laser beams are connected to setup the topology. The location of each vertex is determined as follows: If a laser beam hasn't returned, the distance is used where the existential uncertainty vanishes. The vertices are labeled as *field of view*, the existential uncertainty is sampled from the description of 7.

In order to create polylines for all detected objects, some more work is necessary. Raw measurements of scanning LiDARs provide in general no information about spatial connected objects. The shape information of objects must be retrieved from the raw measurements. Several methods for clustering laser scanner measurements have been proposed in the literature. The approach of Klasing et al. [18] has been used to cluster the objects. Each cluster is turned into a polyline whose vertices are labeled as *obstacle*. The base certainty level for obstacles is used as existential uncertainty with a slight degradation towards the outer vertices of the polyline.

The *Fences*-representation of a ranging LiDAR with maximum field of view of 75m and opening angle of 160° is shown on the right of figure 8. Since the existential uncertainties of free space are linearly decreasing it is easy to represent them with *Fences*. Like ultrasonic sensor there are effects due to signal processing that can only be accurately modeled by the manufacturer.

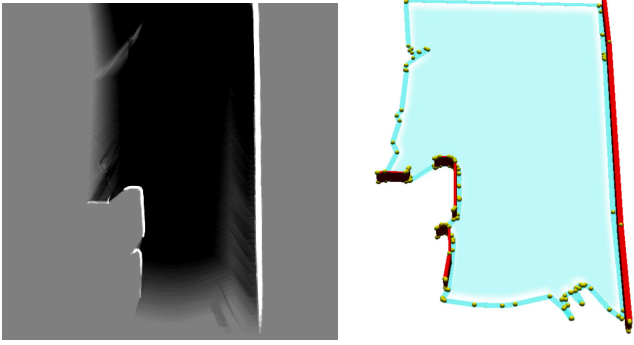


Fig. 9. Occupancy grid (left) and *Fences*-representation (right) that is built from ultrasonic and LiDAR measurements over time by a car that is moving from top to bottom. Free spaces, obstacles and vertices are illustrated by the cyan area, red objects and yellow spheres. The positional uncertainty is expressed by the width of each line.

```

1: procedure FENCES_EXTRACTOR( $g, p_f, p_o$ )
  ▷ Extraction of free space-Fences  $f_f$ 
2:    $t \leftarrow \text{THRESHOLD}(g, p_f)$ 
3:    $f_f \leftarrow \text{CONTOUR\_EXTRACTION}(t, g)$ 

  ▷ Extraction of obstacle-Fences  $f_o$ 
4:    $t \leftarrow \text{THRESHOLD}(g, p_o)$ 
5:    $d \leftarrow \text{DISTANCE\_TRANSFORM}(t)$ 
6:    $s \leftarrow \text{MEDIAL\_AXIS\_TRANSFORM}(t)$ 
7:    $f_o \leftarrow \text{LINE\_EXTRACTION}(s, d, g)$ 

8:   return  $f_f + f_o$ 

9: end procedure

```

Fig. 10. Algorithm for extracting *Fences* from occupancy grids.

V. EXTRACTION OF FUSED *Fences*

The sensor-models described in the last section are next used for data fusion. The mapping strategies can be arbitrary. However, to demonstrate the ability of *Fences* to represent the environment of a vehicle, a grid-based approach is used. Each contour of the *Fences*-representation of a certain sensor is rasterized using methods from computer graphics. The cells of the grid are updated according to the approach of Matthies and Elfes [6]. The resulting occupancy grid built from successive readings of an ultrasonic sensor and ranging LiDAR is shown in figure 9. This grid is used as basis to extract a fused *Fences*-representation of the environment.

In order to accurately represent the fused environment with *Fences*, the algorithm must have certain properties. Since the attributes that are used for the representation are the same as specified in section IV, the position, existential and positional uncertainties and the class needs to be determined for each vertex. In addition, each contour must describe free spaces and obstacles as accurate as possible. The *Fences*-extraction algorithm is outlined in figure 10.

The algorithm is invoked with the current state of the occupancy grid and a probability for free space and obstacles as parameters. The occupancy grid is interpreted as an image during the procedure. The probabilities specify bounds for a

classification of each cell either as free space or as obstacle. A binary decision according to free space and obstacle is possible since driver assistance functions need a definite classification of the environment. Regions with different levels of confidence are usually not wanted. So a set of thresholds is chosen to distinguish between secure and insecure regions of the environment. The bounds must be in the range $0 \leq p_f < 0.5$ and $0.5 < p_o \leq 1$, because Bayesian occupancy grids in the spirit of [6] simultaneously aggregate the likelihood of free spaces and obstacles.

Because occupancy grids can be interpreted as images, tools and algorithms from the field of image processing can be used to extract *Fences*. In order to get a description of all free spaces, a segmentation with respect to the classes free space/ other is performed. Simple thresholding-techniques together with the free space-bound p_f yield this classification. To get a polygonal description, contour extraction-techniques are applied to the segmented image. Finally, the *Fences*-representation is derived by using the existential uncertainty at the given grid cell. The positional uncertainty is set to 0, because free spaces have always accurate positions. The vertices are labeled as *field of view*.

The retrieval of obstacle-fences from an occupancy grid is more involved, because the position of objects is in general not known with full certainty. The position is rather known within some positional bounds. The bounds are determined by applying the same thresholding algorithm to the occupancy grid but with threshold p_o . The most likely position must then be determined within these bounds. Successive locations of an object follow usually a line of cells with high probability. This results from the following observation: Superimposing the positional uncertainties of several measurements of the same objects results in a unimodal probability distribution. Thus, the most likely location of the outline of an object is at the mode of this distribution. Finding the most likely location is not easy because the distribution must not be symmetric. However, the approach can be simplified by assuming a symmetric distribution. This assumption is sufficient for extracting *Fences*, since the positional uncertainty-attribute defined in section IV is understood as a symmetrical uncertainty. The most probable locations s and positional uncertainties d are retrieved by applying a medial axis transform and distance transform. The medial axis transform reduces the uncertainty bounds to a single line: the mode of the assumed symmetric distribution of each location. The distance transform calculates for each location the shortest distance to the bounds.

The most probable locations are finally used as vertices to build a contour for each connected obstacle. Each vertex is extended with several attributes: the x-y-coordinate of the location from s , positional uncertainties from d , existential uncertainties from the grid g and the label *obstacle*. The final representation is created by joining free spaces f_f and obstacles f_o .

The *Fences*-representation of the occupancy grid shown on the left side of figure 9 is depicted on the right side of the same figure. As can be seen, the algorithm described in figure 10 has the ability to accurately extract information from the environment. Free spaces and obstacles are adequately represented by *Fences*. The positional uncertainty is reliably determined from the occupancy grid. Existential uncertainties

and the classification of each vertex is not shown, since this information is directly read from the grid.

VI. CONCLUSIONS AND FUTURE WORK

This work presented crucial algorithms for implementing the unified *Fences*-architecture. Recipes are given to transform sensor readings and grid-based mapping-representations into the topological description that is central to *Fences*. These recipes have been applied to the interfaces between sensors and map-building as well as map-building and driver assistance functions. Simultaneously, it has been shown that the *Fences*-architecture provides an adequate representation for common sensor technologies and mapping strategies.

The recipe has been applied to the conversion of measurements of some frequently used sensors into *Fences*, i.e. the measurements of ultrasonic sensors and ranging LiDARs. The implementation of each step of the algorithm is deduced from the physical properties of each sensor. This approach led to a definition of existential uncertainties for sonar sensors that is new to the field of sensor fusion systems. The ultrasound pressure field along with the energy of a sound wave is used as foundation to derive existential uncertainties. However, this result is merely used as a tool to proof a central claim of the *Fences*-architecture: it is possible to use an attributed topological description for all kind of frequently used sensor principles. The *Fences*-representation is able to describe linear relationships of the environment. It is even possible to describe non-linear relationships by attaching appropriate attributes to the entities of the interface. The claim is further intensified due to the application to ranging LiDARs and its totally different measurement principle. It can thus be concluded that *Fences* is a sufficient representation for sensor readings.

Besides a transformation-directive for sensors, an algorithm has been proposed to extract fused *Fences* from grid-based representations. Since grid-based approaches are almost standard in today's sensor fusion systems, the second central claim of the *Fences*-architecture can be presumed to be proven: the current surroundings are representable with *Fences*. However, this claim is only true if it is sufficient to describe real existential uncertainties by some kind of analytic distribution. This assumption is true for almost all driving assistance systems and can also be applied to some applications in the field of robotics and thus sensor fusion systems in general.

However, there are some aspects that need some further investigation. Some of the proposed properties of the architecture haven't been demonstrated. Of special interest is the ability to exchange the mapping strategy unnoticed and to seamlessly adapt the representation to different requirements and limitations (adaptive coarsening). Second, the applicability of the architecture to several sensors and function must be intensified. Although a wide range of sensors have been investigated, the applicability of *Fences* for an important class of sensors must be shown, i.e. the class of video sensors. Besides, the usefulness for further currently available driver assistance systems has to be evaluated. And lastly, the presented algorithms can be improved. Especially the extraction of *Fences* from grid-based representations can be made more accurate, by enhancing the determination of the most likely position and the real positional uncertainties.

REFERENCES

- [1] T. Kubertschak, M. Maehlich, and H.-J. Wuensche, "Fences - a unified architecture for mapping static environment for driver assistance systems," in *Proceedings of 9th Workshop Fahrerassistenzsysteme*, 2014, pp. 105 – 114. [Online]. Available: http://www.uni-das.de/documents/FAS2014/FAS2014_Tagungsband.pdf
- [2] S. Tokoro, M. Moriizumi, T. Kawasaki, T. Nagao, K. Abe, and K. Fujita, "Sensor fusion system for pre-crash safety system," in *Proceedings of 2004 IEEE Intelligent Vehicles Symposium*, 2004, pp. 945 – 950.
- [3] M. Mählich, R. Schweiger, W. Ritter, and K. Dietmayer, "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection," in *Proceedings of 2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 424 – 429.
- [4] F. Himm, N. Kaempchen, and D. Burschka, "Fusion of laserscanner and video based lanemarking detection for robust lateral vehicle control and lane change maneuvers," in *Proceedings of 2011 IEEE Intelligent Vehicles Symposium*, 2011, pp. 969 – 974.
- [5] T. Kubertschak and M. Mählich, "Extraktion von Parklücken auf probabilistischen Ultraschallkarten," in *Forum Bildverarbeitung 2012*, F. Puente León and M. Heizmann, Eds. KIT Scientific Publishing, 2012, pp. 303 – 314. [Online]. Available: <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/2335958>
- [6] L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid-based representation," in *Proceedings of 1988 IEEE International Conference on Robotics and Automation*, vol. 2, 1988, pp. 727 – 733.
- [7] R. Chatila and J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 138 – 145.
- [8] C. Coué, T. Fraichard, P. Bessière, and E. Mazer, "Multi-sensor data fusion using bayesian programming - an automotive application," in *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 141 – 146.
- [9] K. Dietmayer, A. Kirchner, and N. Kämpchen, "Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme," in *Fahrerassistenzsysteme mit maschineller Wahrnehmung*, M. Maurer and C. Stiller, Eds. Springer-Verlag, 2005, pp. 59 – 88.
- [10] U. Scheunert, N. Mattern, P. Lindner, and G. Wanielik, "Generalized grid framework for multi sensor data fusion," in *Proceedings of 11th International Conference on Information Fusion*, 2008, pp. 814 – 820.
- [11] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21 – 71, February 1998.
- [12] R. Grewe, A. Hohm, S. Hegemann, S. Lueke, and H. Winner, "Towards a generic and efficient environment model for adas," in *Proceedings of 2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 316 – 321.
- [13] T. Weiherer, S. Bouzouraa, and U. Hofmann, "An interval based representation of occupancy information for driver assistance systems," in *Proceedings of 16th International IEEE Annual Conference on Intelligent Transportation Systems*, 2013, pp. 21 – 27.
- [14] R. Lerch, G. M. Sessler, and D. Wolf, *Technische Akustik*. Springer-Verlag, 2009.
- [15] J. H. Lim and D. W. Cho, "Physically based sensor modeling for a sonar map in a specular environment," in *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, vol. 2, 1992, pp. 1714 – 1719.
- [16] T. Weiss, B. Schiele, and K. Dietmayer, "Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids," in *Proceedings of 2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 184 – 189.
- [17] M. E. Bouzouraa, "Belegungskartenbasierte Umfeldwahrnehmung in Kombination mit objektbasierten Ansätzen für Fahrerassistenzsysteme," Ph.D. dissertation, Technische Universität München, December 2011.
- [18] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3d laser data," in *Proceedings of 2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 4043 – 4048.