Sensor Data Representation

A System for Processing, Combining, and Visualizing Information from Various Sensor Systems

Bernd Helge Leroch[1], Jochen Hirth[1], Daniel Schmidt[2], and Karsten Berns[2]

[1] Robot Makers GmbH, Trippstadter Straße 110, D-67663 Kaiserslautern, Germany
[2] Robotics Research Lab, University of Kaiserslautern, D-67663 Kaiserslautern, Germany

Key words: sensor data representation, sensor fusion, assistance functions, autonomous systems

Abstract. One of the big challenges when developing intelligent assistance functions for commercial vehicles is how to handle complex data of various sensor systems. Therefore, a software system has been developed which realizes an abstraction of the concrete sensor system and allows to process the data in a uniform way. Furthermore, it allows to visualize the data in various ways. Experimental results show the functional capabilities and possible applications of the system.

1 Introduction

The development of assistance systems and even autonomous functions is an important topic in the area of commercial vehicles. Especially the domains of construction machines, agricultural vehicles, or municipal vehicles, have evolved in recent years. A precondition for such (semi-) autonomous systems is the perception of their environment. In particular dynamic objects like humans or animals in the machines work space must be detected. It is common sense that for outdoor scenarios a reliable perception can only be guaranteed by using multiple sensor systems, as one sensor can compensate the flaws of another one. One of the biggest challenges in this context is sensor data fusion and sensor data processing.

Therefore, this work describes a representation system, which stores information in so called “object maps”. These maps are software technically realized in a way that various object types can be stored. Furthermore processing algorithms can access all of these objects the same way. Therefore, these object maps realize an abstraction from the actually used sensors. Information from different sensors can be combined to increase the probability of the properties of a detected object. It is also possible to track objects over time in this map. If the sensor itself provides tracking information this can be used to improve the tracking but it is no precondition. Tracking of objects is also possible if the sensor does not provide any tracking information. Beside the requirements for processing and fusion of sensor data there is also the request for representing or visualizing the data. For the realization of different tasks, different views or representations of the data are required. To implement assistance functions or typical robot functionality, grid maps (2D or 3D) and sector maps are required. Developers usually visualize lots of information about the different objects and their history. Operators of vehicles, on the other hand, prefer reduced information sets on the current situation. Therefore, the introduced system provides possibilities to generate different views. Various types of maps can be generated on the basis of the object maps. Additional information can be added to the display if required. The system itself takes care about keeping the different views in a synchronous state. Therefore, the developed system is a very powerful tool, which specially supports processing information from various sensor systems and creates a basis for realizing assistance functions for mobile automation, especially in the domain of commercial vehicles.

2 State of the Art

Autonomous or semi-autonomous vehicles which need to travel beyond their sensory horizon, can either rely on potentially ineffective or misleading local search strategies or use some kind of world model, often referred to as a “map”. It can be provided a priori or build from experience to be able to store traveling paths, obstacles, ground characteristics, and other properties which are required to fulfill the task at hand.
Different data storage or map types have been developed which can be divided into the three classes of metrical, topological, and hybrid maps.
Metrical maps like occupancy grids [1,2,3,4], digital elevation maps [5,6], multi-level surface maps [7], and three-dimensional grids [3] divide the space into specific elements. The insertion of an object leads to the rasterization of this object which means that either one, or multiple cells are filled with the objects properties depending on its size. Afterwards, the information that all of these cells originally belonged this object is usually lost.
Topological maps represent the world as relevant places and their connections on an abstract level [8]. As they just store the detected objects and their properties and no free or undetected space, they are very storage efficient. Graph structures are often used to efficiently store data within these maps and retrieve their contents. In case of a navigation task, the place recognition is very important [9] to not produce artificial ghost nodes.
Many researchers have used combinations of metrical and topological approaches for solving the problems of autonomous mapping, robot localization, and navigation. These hybrid maps combine the advantages of both map types and can be classified by two approaches using hierarchy or abstraction to build the map. Hierarchical hybrid maps are organized in multiple layers. Highly precise metrical maps are used for local places where the precision is needed, while large scale navigation utilizes the computational tractability and compactness of topological data structures [10]. Abstraction-based hierarchical maps are usually built from metrical maps by creating an abstract topological graph.
Examples are Generalized Voronoi Graphs (GVG) [11] which model the topology of the free space. First an occupancy grid is constructed using Bayesian probability techniques that delivers the basis for a Voronoi diagram. Local minima of the base point distance of cells are called critical points. Lines between those points and their base points divide the space into topological regions that are used to construct a topological map.
In summary it can be said that the challenging task is to put sensor data, especially data from different sensors, into a map without loosing information. Furthermore, a uniform representation of information is an important precondition for efficiently developing autonomous or assistance functions. Therefore the idea of so called “object maps” is formulated in [12]. The existing *Sensor Data Representation* system has been extended by those object maps. In the remainder of this work the concept as well as some application examples and experimental results will be presented.

3 Concept of the Sensor Data Representation System

When designing a system for sensor data representation different views should be taken into account: From the control point of view, stored context knowledge should be provided on an as-needed basis in order to keep functional units as simple as possible. In essence, perspective, resolution, and semantic content should reflect the functional units' needs. From the sensor processing point of view, a wide variety of different properties should be storable in order to represent the environment as effective and natural as possible. Perspective, resolution, and semantic content should therefore reflect the specifics of the sensor system in question. In order to balance between expressiveness on the one hand and uniformity on the other, it is proposed to modularize representations into *Structure* and *Content*. While *Structure* reflects the general arrangement (i. e. logical location) of information, *Content* models the information itself. The interfaces of *Structures* represent one element of standardization introduced by this scheme. The second element of standardization is related to the modality of modeling *Content* with property sets. The standard *Structures* with accordant default content constitute the common communication interface which should preferably be used to transfer data between functional units. On the basis of these commitments, a uniform interface is provided which further allows the deployment of several standard transformation algorithms which are wrapped into standardized functional units called *Handlers*. Generic extensions for *Content* and *Handlers* provide additional expressiveness for the realization of flexible representations with appropriate base functionality as for example required by sensor processing or mapping facilities (both have a strong primary focus on data storage).

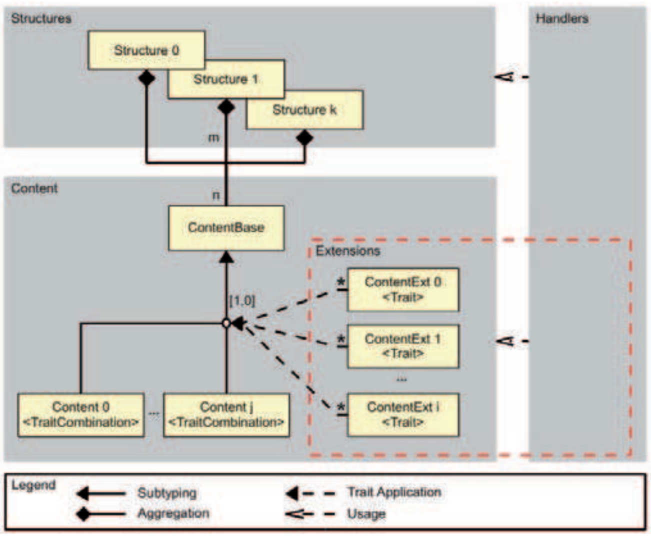


Fig. 1: Refined representation scheme.

As a first modularization step, the separation of locality and semantics shall be highlighted. For that purpose, Figure 1 illustrates the refinement of the top-level partitions *Structures* and *Content*. As already mentioned above, a preferably small number of structural arrangements form the basis of the representation scheme. Typical structures for navigation tasks would for example be topological, grid-based, and sector-based arrangements. Structural entities contain generic content elements which can be accessed using the standardized structural interfaces. Structural entities possess full information on content locality which allows for the uniform treatment of

coordinate transformations between different structural entities. Note that such transformations – due to the strict separation of structure and content – can be applied independently of content translation on the semantic level.
While structure holds information on locality, content is to represent the semantics tied to entities at a given location. To meet the flexibility requirement, content elements have to be tailorable to represent arbitrary sets of features and context. Nonetheless, further elements of standardization shall be introduced to allow for generic translation between different semantic levels. That way, interdependencies between layers (e. g. sensor processing and control) can be avoided. The standardized part of content elements is encapsulated into type *ContentBase* while the generic part is distributed over a set of well-defined *ContentExtension*s (see Figure 1). Typical content extensions are the probabilistic classification, object pose information etc.

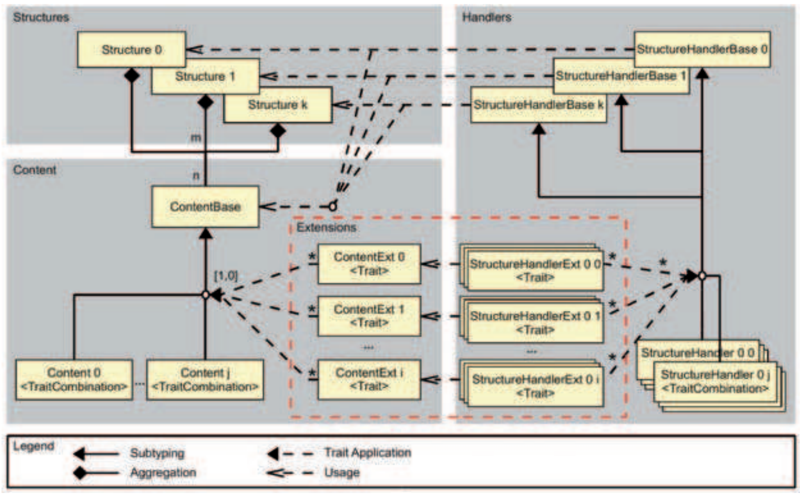
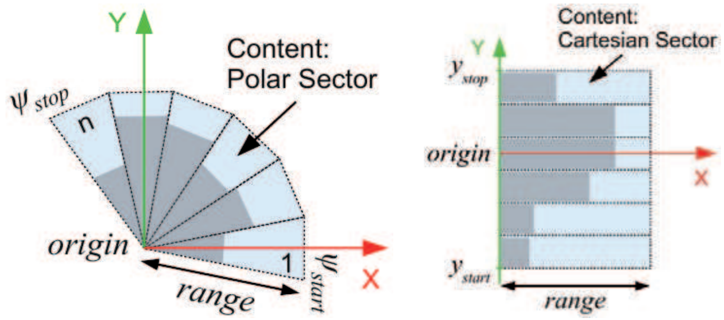


Fig. 2: Refined representation scheme featuring handler constructs.

Generic algorithms are implemented to handle structure and part of the content in a transparent fashion. In particular the transformations between different structures and the translation between various semantic levels of abstraction are of importance. *Handlers* are realized in a similar fashion as content elements. The standardized part is implemented in terms of parametric *HandlerBase* types which take a representation configuration as parameter. The generic part is modularized into *HandlerExtension* traits which may be pulled into a concrete handler to deal with their corresponding *ContentExtension* trait. Figure 2 indicates the combination of the *HandlerBase* with suitable *Extensions* to yield concrete handler types. Furthermore, the relationship between *ContentExtensions* and *HandlerExtensions* is highlighted. Note that a concrete handler type may only provide the basis for handling the content. In sensor processing for example further algorithms are required to extract information from sensor data and to actually fill the representation. In some cases, however, standard combinations of *Handler/ContentBase* and accordant *Content/HandlerExtensions* provide fully functional algorithmic units. One evident class of such functional units are display facilities. The common top-level interface in terms of standardized

structure and content may be used for rudimentary drawing. Further details can gradually be added by introducing functionality via deploying further traits as outlined above.
The representation strictly separates *Structure* and *Content* in order to distinguish clearly between locality and semantics of the represented entities. Typical representation schemes for automation of commercial vehicles are e.g. gird map based structure. A grid can be defined by the cell coordinates of two corners yielding a rectangular structure. Furthermore, the cells are assigned a width and height which defines the resolution of the *Structure*. An other typical map type used in this context are sector-based *Structures* (Figure 3). Either as polar sector maps or as Cartesian sector maps. These configurations have been regarded as most suitable for describing relevant regions of the environment in a simplified and formal way. The *Sensor Data Representation* system also provides the possibility to combine different kinds of maps. An example of a combination of grid map and sensor maps is depicted in Figure 4.



(a) (b)

Fig. 3: Sector-based structure with polar (a) and Cartesian (b) sector arrangements.

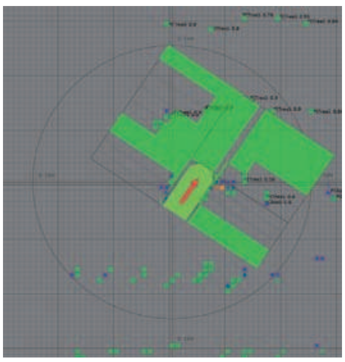


Fig. 4: An example of a combination of grid map and sector map representation.

4 Applications and Experiments

In the context of a joined project the *Sensor Data Representation* system is applied to a wheeled bucket excavator as well as to a garbage collection truck. The goal of the project is to develop assistance functions to increase safety. Therefore, the vehicles are equipped with four cameras, a 3D laser range finder, and four radar distance sensors. The cameras as well as the 3D laser range finder utilize their own evaluation box and provide lists of detected objects with specific properties. All sensor systems are connected to a central processing unit, which runs the *Sensor Data Representation*. The *Sensor Data Representation* combines the information of the sensors. Furthermore, it provides a visualization for the driver as well as a uniform representation for the assistance system and automatic warning system. The whole set up is visualized in Figure 5.
To evaluate the *Sensor Data Representation*, the sensor systems had been mounted to a test vehicle. As test environment a residential area with trashcans in front of the houses had been chosen. This situation provides lots of dynamic objects in the environment as well as the trashcans, which are typical for the garbage collection scenario. Figure 6a provides a color picture of a typical scene. The internal representation of the sensor data can be visualized as an annotated object maps (Figure 6b). All the information on the different objects gathered by the different sensors is stored in a special data structure. These object maps are used whenever detailed information on an object and its properties are required. For example when complex manipulation tasks should be realized those information are used.
On basis on this internal representation different views or maps can be generated. The *Sensor Data Representation* keeps all different maps synchronized. A representation which is typically required for navigation tasks is a so called “grid map”. In that kind of map obstacles are represented by grids. Each grid provides information whether it is blocked or not. An example of a grid map is depicted in Figure 4. Besides those Cartesian grid maps also polar grid maps, also called “seclet maps”, can be generated. While the Cartesian grid maps and the objects maps are mainly used for computer algorithms, the human operators prefer a reduced representation of the scene. Therefore, the polar grid maps, which just display whether there is an obstacle, or even a special type of obstacle, or not seem to be appropriate.

5 Conclusion

This paper presented the *Sensor Data Representation* system. A software tool, which realizes an abstraction of the specific sensor system and allows to process the data in a uniform way. It provides functionality for sensor abstraction, sensor fusion and sensor processing. Furthermore, it provides different possibilities for sensor data representation and visualization. After a short introduction and a brief discussion of requirements for a sensor data representation system, the concept of the developed system has been explained. Afterwards the functioning of the *Sensor Data Representation* is explained using an application example. This application comes from a joined project, which aims to increase safety for commercial vehicles. It has been show how the system can be used to realize an abstraction of the used sensor systems, as well as to generate multiple views to the store information.
The next steps in context of this project will be to generate a special view for the operators of the vehicles. As figured out in interviews with multiple machine operators, they prefer a simple visualization with colored boxes indicating newly detected obstacles. Human operators also prefer an overlay of the camera image, that means that the map is drawn in the current video image. The realization of these requirements will be one of the next steps within the *Sensor Data Representation*.

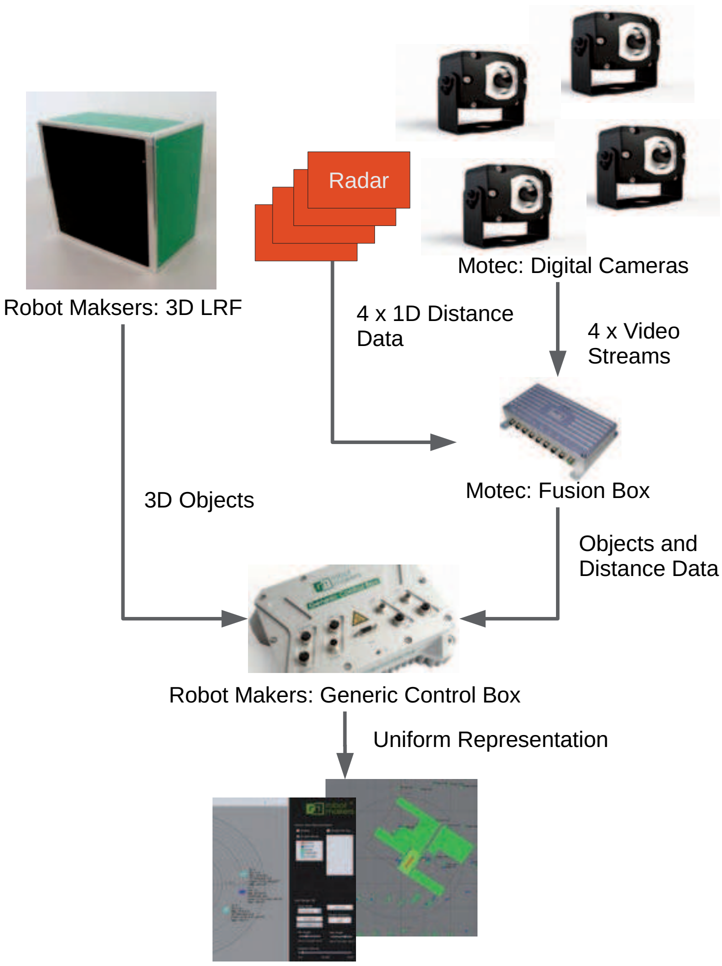
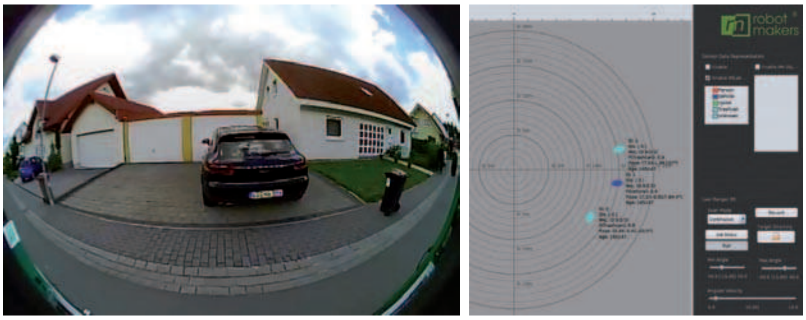


Fig. 5: Structure of the sensor and evaluation system: The Fusion Box evaluates the 4 video streams and the distance information, and generates a list of detected objects. The 3D laser ranger also evaluates the scan and provides an object list. The Generic Control Box is running the *Sensor Data Representation* and generates a uniform representation of the data.



(a) (b)

Fig. 6: (a) Camera image of a typical scene in the residential area. (b) Screen shot from the *fingui* showing a 2D map visualizing object types, that were detected in different colors. In this case one vehicle (blue) and two trashcans (turquoise) were detected.

Acknowledgement

The work at hand was funded from the Federal Ministry of Education and Research (BMBF) under grant agreement number 01|S13027D.

References

1. Alberto Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, 1989.
2. J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.
3. Hans Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 1996.
4. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. In *Artificial Intelligence*, volume 99 of *1*, pages 21–71, 1998.
5. T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166, 2003.
6. Chunhe Yu and Danping Zhang. A new 3d map reconstruction based mobile robot navigation. In *International Conference on Signal Processing (ICSP)*, 2006.
7. R. Triebel, P. Pfaff, and W. Burgard. Multi level surface maps for outdoor terrain mapping and loop closing. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
8. Rodney A. Brooks. Visual map making for a mobile robot. In *Readings in computer vision: issues, problems, principles, and paradigms*, pages 438–443. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

9. Tom Duckett and Ulrich Nehmzow. Performance comparison of landmark recognition systems for navigating mobile robots. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 826–831, Austin, Texas, USA, July 30 - August 3 2000. AAAI Press / The MIT Press. ISBN 0-262-51112-6.
10. Nicola Tomatis, Illah R. Nourbakhsh, and Roland Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44(1):3–14, 2003.
11. S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. In *Machine Learning*, volume 31, pages 29–53, 1998.
12. Bernd-Helge Schäfer. *Control System Design Schemata and their Application in Off-road Robotics*. RRLab Dissertations. Verlag Dr. Hut, 2011. ISBN: 978-3-86853-865-6.