Künstl Intell (2013) 27:161–167
DOI 10.1007/s13218-013-0244-5

RESEARCH PROJECT

From Object Recognition to Activity Interpretation and Back, Based on Point Cloud Data

Sven Albrecht · Thomas Wiemann · Joachim Hertzberg · Hans W. Guesgen · Stephen Marsland

Published online: 28 March 2013

Abstract Semantic mapping of static environments has become a hot topic in robotics. The aim of the MERMAID project was to investigate the transfer of a sensor data interpretation approach for mapping to the problem of activity recognition in smart home applications such as elderly care. The basic structure of the semantic mapping approach, i.e., to assemble hypotheses of object aggregates in a closed-loop process of bottom-up raw data interpretation and top-down expectation generation from a domain ontology, can be extended to the temporal domain to include activity interpretation. This paper reports initial results, based on a study using point clouds from depth (RGB-D) sensor data.

Keywords Semantic mapping · Activity recognition · Smart home · Symbol grounding

S. Albrecht (✉) · T. Wiemann · J. Hertzberg
Institute of Computer Science, Osnabrück University, Osnabrück, Germany
e-mail: sven.albrecht@uos.de

T. Wiemann
e-mail: thomas.wiemann@uos.de

J. Hertzberg
e-mail: joachim.hertzberg@uos.de

H.W. Guesgen · S. Marsland
School of Engineering and Advanced Technology, Massey University, Wellington, New Zealand

H.W. Guesgen
e-mail: h.w.guesgen@massey.ac.nz

S. Marsland
e-mail: s.r.marsland@massey.ac.nz

1 Background and Related Work

Semantic maps extend regular environmental maps (either 2D or 3D) in two respects: (1) mapped objects can be classified, and (2) data from some knowledge base about the objects (or their generic classes) is available [15, Chap. 8.2].

Obviously, some variant of the Symbol Grounding problem [14] needs to be solved in the process, but luckily, not in full generality. In fact, semantic mapping is not the only sub-problem in robot control that requires some aspects of the symbol grounding problem to be solved: any hybrid robot control architecture (i.e., any architecture that amalgamates reactive and deliberative control components) needs to do so [18]. Consequently, the robotics literature contains quite a number of more or less special, restricted solutions to the symbol grounding problem, from pragmatic hacks such as canned object recognition in the 1960s SHAKEY work [22] to principled specializations, e.g., object anchoring [8].

Semantic mapping simplifies or specializes symbol grounding in that only physical objects of known classes need be recognized, as in object anchoring. However, individuals need not be identified and tracked; in most cases, it is sufficient to recognize an anonymous instance of some object class. For example, it might suffice to label an object in a semantic indoor map as *Table*; the individual name *Table-22* may not be needed.

In own previous work on semantic mapping, starting with [23], we have investigated the closed-loop nature of the semantic mapping process. Using 3D point clouds (generated from 3D laser scans in the original work) as sensor data we start from a seed of object detection or hypothesis, such as the detection of a *Table* object (see, e.g., [13] for a sketch how this is achieved). With these seed objects, we use the knowledge base to generate hypotheses about classes of objects that may occur in the vicinity of the object just recognized; practically, we have been using handcrafted DL ontologies for knowledge representation here. These hypotheses help recognize further objects, and so on. Without going into detail here, the point is that knowledge about aggregations of objects and relations between objects helps the sensor data interpretation process. This approach was originally inspired by an approach to Cognitive Vision [21].

The hypothesis behind the work reported in this paper is this: the closed-loop sensor data/knowledge base process that we have been using for semantic mapping can be extended into the temporal domain so that in addition to objects, observed activities can be recognized. In terms of symbol grounding, that means that our approach is extended to grounding symbols denoting activity observed in the environment.

In investigating this ‘activity grounding’, we are entering another currently hot topic, namely, activity recognition; the collection [5] gives a recent overview, and see also [20] for a specification of use cases for a smart home and [6, 12] for other approaches to activity recognition. Our project aimed to develop a case study to investigate whether the methodology used in semantic mapping could be extended to activity recognition, ending in fact in some form of semantic perception of both static parts and dynamic processes and/or events in the environment.

The domain chosen for this project is that of a smart home providing monitoring for the elderly; the project partner Massey University has been involved in this for some years (see http://muse.massey.ac.nz). In that domain, activity (or behavior) recognition is often based on machine learning algorithms applied to sets of tokens that arise from state-change sensors such as motion detectors, electrical usage sensors, and cupboard sensors. In this project we focus on change detection based on analysis of scenes. This is not as simple as using state-change sensors, but the aim is the same: to provide a set of tokens concerning the use of objects that will enable behavior recognition algorithms to identify and analyse the actions of the house inhabitant(s).

As a running example for objects and activities from that domain, consider breakfast time at a kitchen table: relevant objects are plates, mugs, cereal bowls, cutlery, and diverse place settings, which are aggregates of these; expected activities are laying the table, eating breakfast, and so on.

In the rest of this paper, we first sketch the process of scene segmentation that allows regions in the scene to be clustered and objects segmented. Then we turn to describing how activities are represented and recognized, based on prior recognition of objects and possibly other context information.

2 Segmenting Scenes and Hypothesizing Objects

In the project we use an RGB-D camera (e.g., MS Kinect) to capture 3D point clouds of the observed environment. A 3D point cloud is a collection of points in space that samples object surfaces at a certain resolution and frame rate. The Kinect produces point clouds consisting of up to 640 × 480 3D points at a maximum frame rate of 30 Hz. The point cloud data is relatively sparse and noisy, but still delivers enough data to create a realistic polygonal mesh of its environment. A polygonal mesh represents objects more compactly than a point cloud and yields actual surface definitions rather than just point samples. Our approach to detecting changes in the monitored environments is to create a labeled reference mesh of the scene and compare it to the reconstructions from the incoming camera frames.

Surface reconstruction and labeling is done using the Las Vegas Reconstruction Toolkit [26]. It contains a set of tools for triangle mesh generation and optimization from unorganized point clouds. Since it was mainly developed for mobile robotic applications, it has a strong focus on noise compensations, data compression and execution speed. Surfaces are reconstructed by an optimized Marching Cubes implementation [19] using Hoppe’s distance function [16] that was tuned to cope with sparse data and the presence of noise.

A scene is coarsely segmented by clustering connected patches in the computed triangle meshes using a region growing approach. The generated meshes are stored in a linked data structure that allows adjacent triangles to be found in constant time, which provides a significant speed up compared to point based clustering. Each created cluster can be associated with a semantic label. For basic geometries like floors, walls and ceiling, labeling is done automatically by analyzing the size, position and orientation of every extracted cluster. Other relevant clusters, such as chairs and table tops, can either be labeled manually or automatically by employing more constraints in form of domain knowledge about the spatial relationship between individual planar clusters. For example, a *Chair* is described by a horizontal planar cluster of a certain size and height (the seating surface) and another planar cluster perpendicular to it (the backrest), cf. [13].

This allows ‘change detection’ to be done, where variations in a scene over time can be identified and analyzed. When dealing with static sensors in an environment, such as motion sensors, electrical use sensors, and door open/closed sensors, the percept is typically turned in a token representing the fact that the feature perceived by the sensor has changed. With a more complex sensor, this is harder to arrange. However, we can detect change here by identifying that objects have appeared and disappeared from a scene, and so labeling these changes with tokens.

The basic idea for change detection in the environment is to generate a reference mesh of the observed scene and

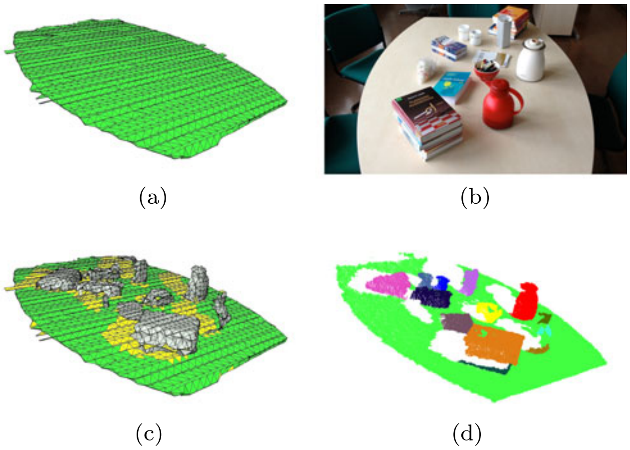

Fig. 1 Example for scene segmentation: (**a**) Mesh corresponding to the reference scene of an empty tabletop. (**b**) Photo of the tabletop with breakfast tableware (and more) present. (**c**) The mesh corresponding to (**b**). Detected holes are shown in yellow, new object clusters are depicted in gray. (**d**) Point cloud corresponding to (**b**). Individual object clusters are signaled by *color* (Color figure online)

label the relevant clusters. In the case of the breakfast example, this is done by reconstructing the empty breakfast table and labeling the table top plane. This reference mesh gets compared frequently with the polygonal reconstructions generated from the current sensor output. To detect changes in the scene, the size and shape of the table top cluster of the reference mesh is compared to the reconstruction from the current camera frame by calculating the distance of each triangle of the table top cluster to the nearest triangle in the current observation. If such a triangle is found within a predefined tolerance distance, we treat this triangle as confirmed, i.e., nothing has changed within this region.

If new objects are moved into the scene, the number of confirmed mesh triangles is reduced due to occlusions. Therefore, if the number of confirmed triangles decreases, we assume a new object was added to the scene. An increasing number in turn suggests that some object was removed. For added objects, clusters in the mesh appear above the tabletop plane that have no correspondence to the reference mesh and can thus be identified using a similar distance threshold to the one used to detect the shadows. Figure 1 presents an example of our segmentation results.

New clusters are used to generate object hypotheses from an ontology. In a first step the spatial dimensions of each cluster are considered to restrict the number of object candidates. For example, if a cluster is higher than a certain threshold, it is sensible to exclude plates from the potential object candidates. Similarly if the height is below than a certain threshold, this cluster can be no pitcher. After this first coarse filtering, the shape of the measurement points is examined.

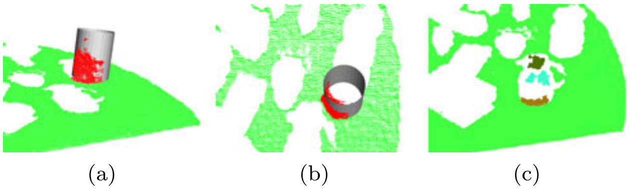

Fig. 2 (**a**) and (**b**): Result of automatically fitting a geometric primitive (cylinder) against the cluster of a pitcher. (**c**) Illustrates that one object (the *red pitcher* in Fig. 1b) is split into 3 distinct clusters. For these clusters, no plausible hypothesis exists in our knowledge base. The data depicted here is the same as in Fig. 1b, but only the data points for the object in question and the tabletop are shown (Color figure online)

Most tableware objects can be approximated by geometric primitives, for instance most mugs are cylindrical, with a handle added, and many bowls resemble a hemisphere. Geometric primitives are fitted in a RANSAC fashion, similar to Schnabel et al. [24]. Currently we do not look for multiple primitives in one cluster, but determine which primitive fits the given points best. Schnabel et al. [24] have gone beyond that, combining several primitives. Note that while the thresholds used in the RANSAC approach could be learned in principle, we chose to set them manually using domain knowledge and experimental results.

This association between real objects and geometric primitives is not guaranteed to yield sensible results in general, but it works well for restricted scenarios like the breakfast setting, where the focus is to detect subsets of given objects. An example for this approach is given in Figs. 2a and 2b. On the downside of this approach, we are unable to generate sensible object hypotheses if the point cloud data is too noisy, possibly due to reflections. For example, if one object is split into several point clusters, or multiple objects are too close to each other, forming just one cluster, our approach will fail. Figure 2c shows an example.

Our approach here [1, 13] is to close the loop from knowledge representation to object type recognition. We are using a Description Logic (DL) ontology, including SWRL rules, formalizing scenes (such as breakfast) and aggregates in these scenes (e.g., a breakfast setting). A small excerpt from the ontology is depicted in Fig. 3 and a exemplary SWRL rule is shown below:

```
Mug(?p) ← CylindricalPointCluster(?p)
∧ isOnTableTop(?p, True)
∧ hasHeight(?p, ?h) ∧ swrlb:greaterThan(?h, 0.06)
∧ swrlb:lessThan(?h, 0.15) ∧ hasWidth(?p, ?w)
∧ swrlb:greaterThan(?w, 0.05)
∧ swrlb:lessThan(?w, 0.1%2)
```

So once an object of some type has been hypothesized from the sensor data using the approach just described, other hypotheses are generated from the DL domain model, which may introduce additional object types; this is then used for

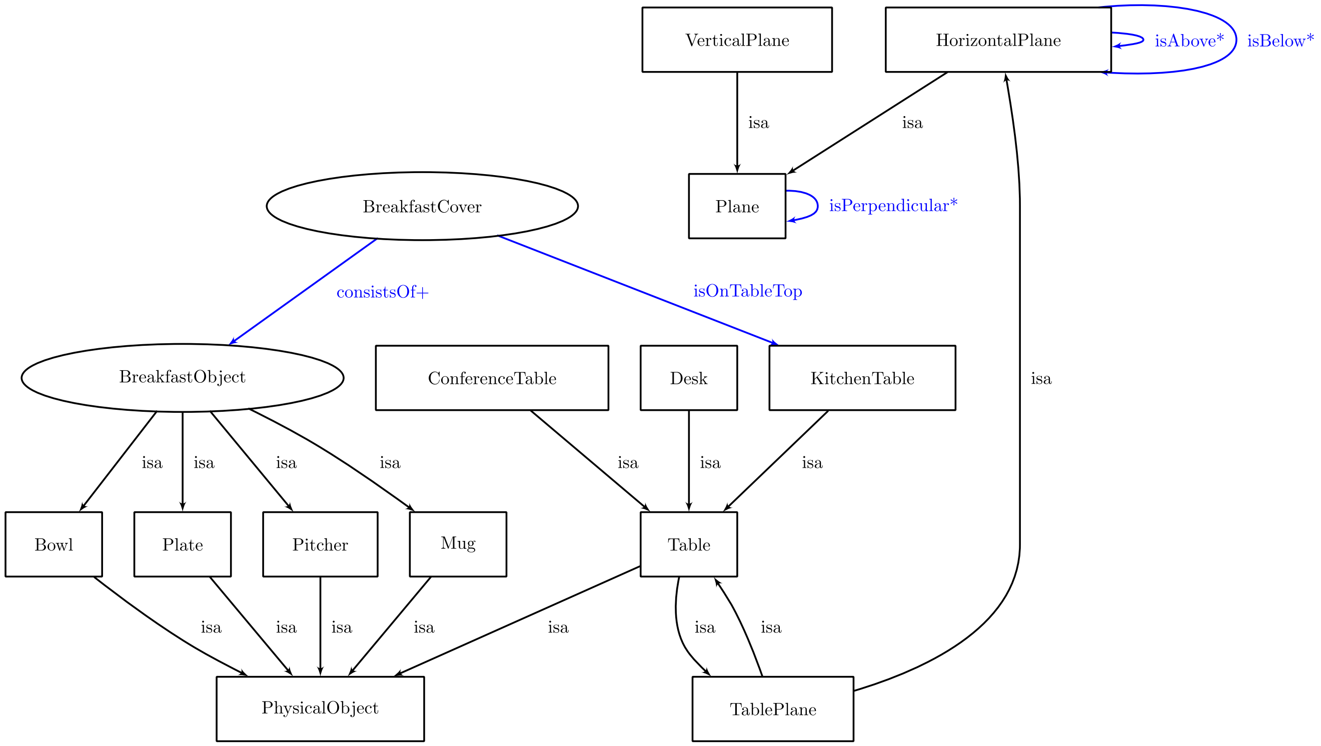

Fig. 3 Excerpt from the DL ontology connecting the scene segmentation with abstract aggregate *BreakfastCover*. Physical entities, which are directly associated with measured data, are drawn on *boxes*. *Blue edges* indicate relations going beyond inheritance

forming expectations what is likely to be found in the sensor data. For example, having detected a plate and mug on one side of the table, it is likely that a knife is present, too. Context information from other sources, for instance the time and day, may get used in the reasoning [12]. This approach is inspired by Neumann and Möller [21]. It has been applied to activity recognition, too [4], for a highly standardized family of activities, namely, aircraft service activities on the gate; the sensor data used here was video.

Comparing the current sensor data with the reference scene over a period of time thus yields a temporal sequence of appearing and vanishing objects. Given such a sequence, the question arises of how to interpret the changes occurring in the sequence. To answer this, we have to perform two interdependent tasks: firstly, divide the data stream into segments that can be associated with activities (such as setting the breakfast table); and secondly, attribute the correct activity type to each segment. Neither of these tasks is trivial, and we next sketch our approach to perform them.

3 Segmenting and Hypothesizing Activities

In activity interpretation, just like in object interpretation, there are the two mutually dependent sub-problems of segmenting and recognizing activities, where the segmentation has to happen in space-time, rather than just space. The underlying assumption here is that an activity is an entity in 4D space-time, which can be formalized in some variant of Allen's calculus of relations [2], such as 4D Region Connection Calculus [9]. Segmenting two activities is easy if they are set apart in time (such as toasting and buttering a slice of bread, with some cooling time interval in between), but it is harder if they meet in time (like taking a clean plate from the dishwasher and placing it on the table), or if they are even overlapping (eating breakfast while reading the newspaper). An approach to simultaneous segmentation and recognition using Hidden Markov Models, with time as an additional variable, was considered by Chua et al. [7].

Researchers in related areas, such as activity recognition using wearable sensors [27, 28], often assume that the data stream is already segmented in time, i.e., different activities are separated by some phase of inertia, which can be identified from the sensor data. A scene that differs from the sequence of unchanged scenes marks the start of the next activity. To segment activities that meet or overlap, sliding-window approaches (e.g., [10, 25]) are commonly used. The size of the window(s) is determined by activity(-ies) length, assuming that each activity that we can label in the data stream has a range of typical lengths. For example, *Set-breakfast-table* would normally take only a couple of minutes. If knowledge is available about contexts that influence the duration, it has to be part of the knowledge base. Chua et al. [6] describe an alternative based on the likelihood computation in Hidden Markov Models.

Our approach to activity recognition is to extend the closed-loop approach that we have been using for generating object hypotheses, treating the segmentation and recognition problems simultaneously. Moreover, we rely on recognizing the appearance of objects, as described previously, to signal the start and end of activities in the current context. For example, if a mug appears in the scene around the time of a weekday breakfast time, it is likely to signal the start of a table setting activity. This approach is reminiscent of the one used by Kim et al. [17] for gesture recognition, where the starting point of a gesture is detected. In their work a window is then slid across the observation sequence until an end point is reached.

Using context knowledge is crucial for generating targeted activity hypotheses. Context information in our application context can be temporal, spatial, environmental, or health information. Guesgen and Marsland [11] suggest context maps as a possible means to that end. A context map is a graphical representation that combines context information of a particular type. For example, a temporal map contains all entries related to time, such as time of the day, day of the week, month, season, etc. The nodes in the graphical representation can be ordinal values, indicating how often an activity happened in the context represented by the node, or probabilities, indicating the likelihood of the activity in this context. Bhatt and Loke [3] propose the integration of formal methods in spatial representation and reasoning with logic-based methods in reasoning about events, actions, and change.

In order to get from the segmented object hypotheses to assumed activities, we propose to encode the current sensor input into a state containing the object hypotheses along with the relevant contextual information. Table 1 shows an example sequence of such states. From the sequences of these states we aim at extracting the activities associated with several objects that co-occur over multiple consecutive states. Of course we have to take into consideration that some objects could occasionally disappear for short periods from our state representation, due to sensor noise or occlusion (the latter being detectable from the sensor data), or even a change in routine. For example, if *Eat-breakfast* is associated with *Mug*, *Plate* and *Bowl* all being present, if we encounter a state where one of the necessary objects is not present, we cannot simply assume that *Eat-breakfast* has been completed and the next activity of *Clear-breakfast-table* is in progress. Instead, we have to take the subsequent states into consideration, to see if we failed to detect the object just for a short time or if it actually disappeared (in which case it may well be appropriate to assume that *Clear-breakfast-table* has started, which hypothesis can then be tested at subsequent time steps). An example of such a missed detection is given in Table 1: state s_{i+3}, where *Mug* vanishes, to reappear in the following state.

Table 1 Example sequence of states, symbolizing *Set-breakfast-table*. Note that while the table does not contain individual objects, an additional line labeled *Mug* would appear if a second instance of *Mug* was detected in the point cloud data

State	s_i	s_{i+1}	s_{i+2}	s_{i+3}	s_{i+4}	s_{i+5}	s_{i+6}
Mug	T	T	T	F	T	T	T
Plate	F	F	F	T	T	T	T
Bowl	F	F	F	F	T	T	T

In the current state of our research we are able to extract temporal sequences of states from the Kinect point cloud data. As a next step we intend to use machine learning techniques to start identifying activities from the state sequences of recorded data.

4 Conclusion

Our approach in the MERMAID project to recognizing (or rather, hypothesizing) both objects and activities is to view it as a closed-loop process among interpretation of the raw data coming from the sensor and reasoning based on a DL ontology plus SWRL rules about the context and object information that is currently available. The sensor data are point clouds taken from the depth component of RGB-D data of a narrow scene, namely, a kitchen table. The "bottom-up" part of detecting objects in the sensor data stream and the "top-down" part of generating object and activity hypotheses are closely intertwined—in fact, there is no order among them, but they both work conceptually concurrently.

In relation to the Symbol Grounding problem, our approach has two lessons to impart. First, grounding object and activity symbols are two closely related, in fact, intertwined issues. Some lines of work (including our own) exist today that have started to tackle this. Their respective settings are restricted, and it has not been shown how they scale to unrestricted ones. An open question here is, how unrestricted "unrestricted" actually is—in our approach, context plays an important role for restricting the reasoning, and it seems to be generally applicable in principle. Second, the naïve approach of first grounding the symbols in sensor data and then starting the reasoning based on the symbols, is not the only one. It makes sense and is feasible to close the loop between sensor data interpretation and reasoning. It appears that the more comprehensive problem of doing both, rather than tackling "just" the symbol grounding part, becomes in fact easier to solve.

Acknowledgements Work in the project MERMAID reported here was partially funded by BMBF (grant NZL 10/003) and the RSNZ (grant FRG10-11) under a bilateral research collaboration agreement. This is gratefully acknowledged.

References

1. Albrecht S, Wiemann T, Günther M, Hertzberg J (2011) Matching CAD object models in semantic mapping. In: Proc ICRA 2011 workshop on semantic perception, mapping and exploration
2. Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843
3. Bhatt M, Loke S (2008) Modelling dynamic spatial systems in the situation calculus. Spat Cogn Comput 8(1):86–130
4. Bohlken W, Koopmann P, Neumann B (2011) Scenior: ontology-based interpretation of aircraft service activities. Tech rep FBI-HH-B-297/11, Department of Informatics, University of Hamburg
5. Chen L, Nugent C, Biswas J, Hoey J (eds) (2011) Activity recognition in pervasive intelligent environments. Atlantis Press, Amsterdam
6. Chua SL, Marsland S, Guesgen HW (2009) Behaviour recognition from sensory streams in smart environments. In: Proc AUS-AI
7. Chua SL, Marsland S, Guesgen HW (2009) Spatio-temporal and context reasoning in smart homes. In: Proc COSIT workshop on spatial and temporal reasoning for ambient intelligence systems. 2009
8. Coradeschi S, Saffiotti A (2003) An introduction to the anchoring problem. J Robot Auton Syst 43(2–3):85–96
9. Galton AP (1993) Towards an integrated logic of space, time and motion. In: Proc IJCAI-93, pp 1550–1555
10. Gao J, Hauptmann AG, Bharucha A, Wactlar HD (2004) Dining activity analysis using a hidden Markov model. In: Proc ICPR2004, pp 915–918
11. Guesgen H, Marsland S (2011) Recognising human behaviour in a spatio-temporal context. In: Chong NY, Mastrogiovanni F (eds) Handbook of research on ambient intelligence and smart environments: trends and perspective. IGI Global, Hershey, pp 443–459
12. Guesgen HW, Marsland S (2010) Recognising human behaviour in a spatio-temporal context. IGI Global, Hershey
13. Günther M, Wiemann T, Albrecht S, Hertzberg J (2011) Model-based object recognition from 3d laser data. In: Proc KI-2011, pp 99–110
14. Harnad S (1990) The symbol grounding problem. Physica D 42:335–346
15. Hertzberg J, Lingemann K, Nüchter A (2012) Mobile Roboter. Eine Einführung aus Sicht der Informatik. Springer-Vieweg, Berlin
16. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. Comput Graph 26(2)
17. Kim D, Song J, Kim D (2007) Simultaneous gesture segmentation and recognition based on forward spotting accumulative HMMs. Pattern Recognit 40(11):3012–3026
18. Kortenkamp D, Simmons R (2008) Robotic systems architectures and programming. In: Siciliano B, Khatib O (eds) Springer Handbook of Robotics. Springer, Berlin, pp 187–206, Chap 8
19. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3D surface construction algorithm. In: ACM SIGGRAPH
20. Lyons P, Cong AT, Steinhauer HJ, Marsland S, Dietrich J, Guesgen HW (2010) Exploring the responsibilities of single-inhabitant smart homes with use cases. J Ambient Intell Smart Environ 2(3):211–232
21. Neumann B, Möller R (2008) On scene interpretation with description logics. Image Vis Comput 26(1):82–101
22. Nilsson N (1984) Shakey the robot. Tech rep TN 323, SRI International
23. Nüchter A, Hertzberg J (2008) Towards semantic maps for mobile robots. J Robot Auton Syst 56(11):915–926
24. Schnabel R, Wahl R, Klein R (2007) Efficient ransac for point-cloud shape detection. Comput Graph Forum 26(2):214–226
25. Tapia EM, Intille SS, Larson K (2004) Activity recognition in the home using simple and ubiquitous sensors. In: Proc PERVASIVE, pp 158–175
26. Wiemann T, Lingemann K, Nüchter A, Hertzberg J (2012) A toolkit for automatic generation of polygonal maps—Las Vegas reconstruction. In: Proc ROBOTIK-12, pp 446–451
27. Wu JK, Dong L, Xiao W (2007) Real-time physical activity classification and tracking using wearable sensors. In: Proc ICICSP 2007, pp 1–6
28. Zappi P, Stiefmeier T, Farella E, Roggen D, Benini L, Tröster G (2007) Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness. In: Proc ISSNIP'07, pp 281–286

Sven Albrecht is a Ph.D. student in the Knowledge-Based Systems group at Osnabrück University. His main research interests are semantic mapping and 3D data interpretation.

Thomas Wiemann is currently finishing his Ph.D. thesis on automatic generation of polygonal maps for robotic applications at the KBS group at Osnabrück University. An important aspect of his research is generating semantic scene interpretations from point cloud data.

Joachim Hertzberg is a full professor for computer science at Osnabrück University, heading the KBS group; he is head of the Osnabrück branch of DFKI's Robotics Innovation Center, too. His areas of interest are AI and Mobile Robotics, with a focus on plan-based robot control.

Hans W. Guesgen is a professor of computer science and leader of the Computer Science and Information Technology cluster at Massey University. His research interests include ambient intelligence, knowledge representation, constraint satisfaction, and spatio-temporal reasoning.

Stephen Marsland is a professor of scientific computing at Massey University. His primary areas of research are Euler equations on diffeomorphism groups and machine learning.