A FIVE LAYER ARCHITECTURE FOR REAL TIME OBJECT DETECTION IN INDOOR ENVIRONMENTS

Hartmut Surmann *

* *German National Research Center for Information Technology, Institute for Autonomous intelligent Systems, Schloß Birlinghoven, D-53754 Sankt Augustin, Germany, email: surmann@gmd.de*

Abstract: The prognosis for autonomous mobile service robots for transportation tasks in indoor environments, e.g. multistory buildings differs a lot from todays reality. The robots have to act in dynamic environments with a huge number of components. The robots total repertoire of affordances and intelligence is highly according to the complexity of the building and its respective task. Difficult tasks can only be achieved of the base on immediate sensing of the environment and real-time evaluation of the sensor processing modules. This paper describes two components; a five layer sensor architecture which integrates a spatial database for multistory buildings and different modules of medium and limited intelligence but with real-time capabilities. A basic feature extraction algorithms of layer II is presented as an example of the modules.

Keywords: Programming and sensor architectures, collision avoidance and sensor-based control, robot sensing and data fusion.

1. INTRODUCTION

Intelligence can be considered from a variety of perspectives. One of these approaches - named "artificial intelligence"- is defined as "making machines intelligent", which means to make them act as we would expect humans to act. Therefore intelligent autonomous mobile service robots should act in "natural" human environments, e.g. office buildings. The robots need knowledge and information to behave intelligently. Knowledge may be given explicitly (e. g. about the robot's environment), implicitly or tacitly. To let the robots act knowledgeably, various types of knowledge are necessary (Arkin 1998) e.g. spatial world knowledge, object knowledge, perceptual knowledge, behavioral knowledge, ego knowledge and intentional knowledge. Explicit knowledge can be given in advance to the robots, e.g. as a spatial database but current information about its actual environment must be acquired too (situated concept of the embodied AI (Pfeifer and Scheier 1999)). The machine must make sounds, change the environment in some ways, move, draw something, produce signs that can be interpreted by others (Pfeifer and Scheier 1999). Both, knowledge and information from the sensors, have to be combined in **real time**.

Prognoses at the beginning of the nineties claimed for the year 2000 a number of about 50,000 independently operating autonomous service robots in different areas of production and also service sector (Schraft and Volz 1996). The reality looks different. In industrial environments Guided Vehicles (GV) i.e. vehicles guided by a magnetic or optical line are standard (PSLT 1998). However in practice autonomously acting mobile service systems i.e. systems not restricted by a line are used extremely rarely although many research groups work on this for several years – particularly with mobile systems for transportation tasks (Engelberger 1993, Graf and Weckesser 1998, Vestli 1996). One of several reasons for the gap between prognoses and reality is the complexity of the environment in which the robots have to act, particular for multistory buildings. Current robot control architectures (Arkin 1998, Konolige and Myers 2000) are not intelligent enough to handle the complexity of multistory buildings because the inherent sensor architecture is too weak. The robots can only operate in one level like the museum robots (Thrun et al. 1999). Multistory buildings are dynamic environments with elevators and fire doors that lead to special demands for the mobile units.

This paper describes a five layer sensor architecture with an integrated spatial database for indoor environments. The spatial database serves as knowledge base for the three upper layers. The total representation is based on natural landmarks which have to be detected by the evaluation modules in real time. The paper presents one real time module of layer II for a robust feature extraction in indoor environments. The real time requirement leads to major design problems. From the large group of principly possible algorithms only a few can be selected. All the remaining are (at present) not real time capable. At best one of the remaining algorithms is brilliant and fulfills all requirements (e.g. real time). But often only simple and non-universal algorithms remain. These simple algorithms then must be merged in an architecture which has to absorb their drawbacks.

Know-how and manpower have been invested in the architecture of the system. Architecture here means the correct selection of hardware and software components and their interconnections, not the implementation of algorithms for the solution of sub-problems. The overall performance (i. e. the visible "intelligence") of the robots is situated mainly in the architecture of the complete system.

The paper is structured as follows: Chapter 2 describes the robots and their environment, chapter 3 contains the description of the architecture and the spatial database. Chapter 4 gives some results and chapter 5 concludes the paper.

2. THE ROBOTS

The robot team consists of three mobile robots (Fig. 1). Each is about 80 cm × 60 cm large and 90 cm high. The mobile platform can carry a payload of 200 kg at speeds of up to 0.8 m/s (about half the speed of a pedestrian). The right and left driving wheels are mounted on a suspension on the center line of the mobile platform. Passive castors on each corner of the chassis ensure stability. The major application areas are administrative buildings, like insurances, banks, hospitals, and offices. The variety of tasks is very large: from mail, lunch, or coffee distribution on schedule or on demand via intra-/internet during the day to patrolling at night.

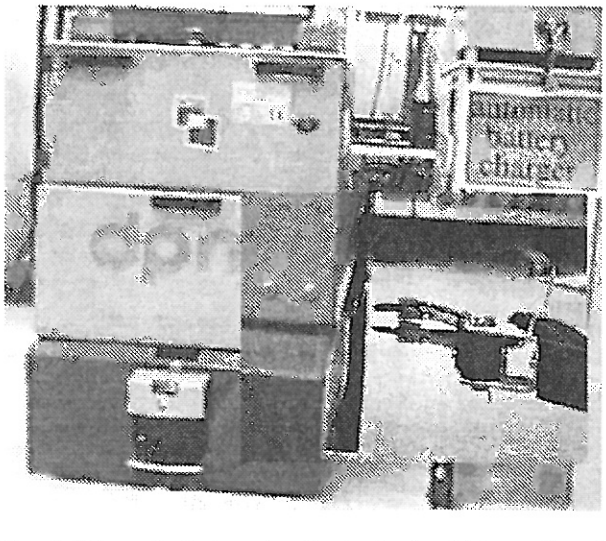

Fig. 1. The robot charging its batteries. It is equipped with two drawers on each side

The core of every robot is a Pentium PC 166 MHz with 16 MB RAM and real-time Linux. Two microcontrollers are used. One controls the internal states, display, keyboard and radio link of the robot. The other one manages the motors and the optical line-tracking. Besides autonomous navigation, which is done by using fuzzy-logic (Surmann *et al.* 1995), the robot has the ability to drive along an optical line. This navigation is preferred in narrow passages, e.g. while entering an elevator or during docking maneuvers. The robots can change from autonomous navigation to line-tracking and back by themselves.

Every platform is rigged with two laser scanners, one on the front and one on the rear. Each laser scans 180° of the environment (Pauly *et al.* 1998).

The 250 kg robots can operate for about 8 hours with one battery charge. When the power drains the robot visits an automatic power recharging station, connects itself to it and recharges its batteries. A mission server will be informed about the time period for which the robot is unavailable and another robot takes over. Current information about the robot, e.g. robot monitoring, robot jobs, schedules, battery loading, laser scans etc., is represented at http://lamu.gmd.de:8080.

The robots currently move in the GMD-ROBOBENCH, an H-shaped building of about 1600 m^2 .The building is departed into 84 office rooms and 15 corridors on 3 levels reachable by elevators. All elevators and 17 doors can be controlled automatically by every robot with Internet radio link. Office room doors can be opened by humans only. Web cameras on some floors allow live observation of robot actions.

3. THE ARCHITECTURE WITH THE SPATIAL DATABASE

3.1 *The architecture*

The key idea of the system architecture is based on the principle of distribution of intelligence. The intelligence needed by the mobile robots to fulfill their tasks is distributed between all five layers and their evaluation modules. The spatial database serves as central knowledge base for the higher layers. The entire amount of intelligence is higher, though each component has only average intelligence.

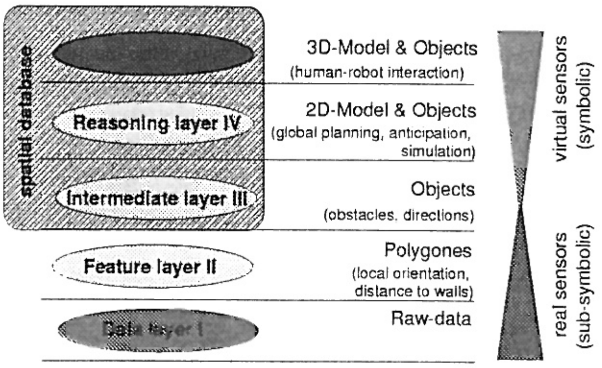

Fig. 2. Five layer sensor architecture for guiding service robots in office environments.

3.2 *The spatial database*

The spatial database is the base for the layers III - V. On one hand it is influenced by the different sensors and the possible features extracted from the sensors as well as the robot mission. On the other hand the spatial database influences the selection of the sensors and the feature extraction algorithms.

The spatial database introduced here contains a topological map to schedule jobs for robots as well as to plan robot behaviors. The planner is a graph planner (Kautz and Selman 1996). In addition a geometrical and feature map for the landmark recognition and robot relocalization is implemented. The codes necessary for the handling of the elevators and doors are likewise stored in the spatial database. Furthermore references to SQL databases are included in the model. In the SQL database lists of persons are stored with their surname, first name, email address, telephone number and door designator. Thus all information needed for the interaction of robots and humans is available and it is possible to implement a user friendly interface (Surmann and Theißinger 1999).

It is important that the spatial database is a general prototype for multistory buildings. It can be simply adapted to other buildings and is easily extended to deal with dynamic objects. Figure 3 shows a segment of our spatial database for multistory buildings. Different sections are parenthetical by <name> ... </name> like the XML specification. Therefore, robots or other clients or servers can exchange and update there spatial database over the Internet.

The "corridor"structure is a part of the topological map and used at reasoning layer IV. Each corridor has an unambiguous number (0), a name (T0-Roboter), a corridor type (0: office corridor, 1: junction corridor, 2: elevator ...), dimensions (width, length, height, orientation), a global position (x,y,z) and a list of branches (x,y,z). The "Branch" section expresses the spatial relation between the different corridors and defines that the first branch (0) of the current corridor is connected to the first branch (0) of the corridor (2) e.g. (0 2 0). The second branch (1) is connected to the third branch (2) of corridor with the number 3. The branch definition is the basic description for the graph planner.

The section <RightWall> and <LeftWall> describes the geometric dimensions of the right/left wall respectively. The format is the following: type (1-10 walls, 11-15 doors, 16-20 protrusion, 21-25 niche, 31 fire extinguisher, 36-40 door signs ...), color (rgb), width, local position, thickness, depth, material (glass, wood, metal,...), height, reference to the telephone list, local height, a free entry. This part can either be generated automatically from a floor plan or by a ride of the robot in the corridor. The corridor interconnections as well as the control codes for the elevator and fire doors have to be added by hand. Additional parts e.g. <Office> ... </Office> and <Codes> ... </Codes> describe the dimensions of offices and the codes for opening the fire doors or elevators.

```
<Corridor>
0 T0-Roboter 0 206 2190 300 -90
0 0 0   0 -103   0
</Corridor>

<Branch>
0  2  0
1  3  2
</Branch>

<RightWall>
 1 ffffff 20   0  74   0 0   0   0      0 0
31 ff0000 16  74  53 -16 3 120   0      0 0
 1 ffffff 20 127 117   0 0   0   0      0 0
16 ffffff 25 244 160  -5 3 200   0      0 0
 1 ffffff 20 404  15   0 0   0   0      0 0
36 00ffff  1 419  15  -1 4   9 C2-T11 157 0
..
</RightWall>
```

Fig. 3. Segment of the spatial database.

One major point is that different internal descriptions are generated on the base of the spatial database e.g. VRML for 3D robot tracking, 2D floor plans for gnuplot and a java applet. The floor plans are also used for 2D tracking and simulations which is important for the design of the behavior modules as well as the prediction of the near future.

3.3 *The data layer*

The **data layer** extracts raw data, e.g. up to 500 local distance values (Schmersal, Sick: 361) every 30 ms in the range from 0-180° (Schmersal EOT 1999). This layer has to implement the physical protocols for the laser scanner according to the technical description. The predefined warning and protection zones are evaluated. The sensor has a hardware connections to the motor. Therefore a hit within the zones directly influences the speed without any calculation. Usually it is checked for a semi-circular or rectangular zone whether a point P(x, y) of the current laser scan is situated within this zone. Let n = number of distance values. Then it can be checked in O(n) by means of a bit test whether the warning or protection field is violated. Several different or permanently changing warning and protection zones can also be checked in O(n).

Besides the information "zone violated" or "zone free" the length of an obstacle can be measured by the euclidian distance: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2)}$. This distance can also be used for speed reduction. For indoor environments we reduce the number of distance values of the laser scanner from 500 (12 Bit resolution) to 126. So the processing time for a complete scan is 60 ms yielding 16 - 17 scans per second. The distance values of the reflection points and the related angles are converted to local x, y coordinates (Fig. 4). On the lowest levels of the model particularly robust and real time capable algorithms are implemented. Other sensors are added to this level.

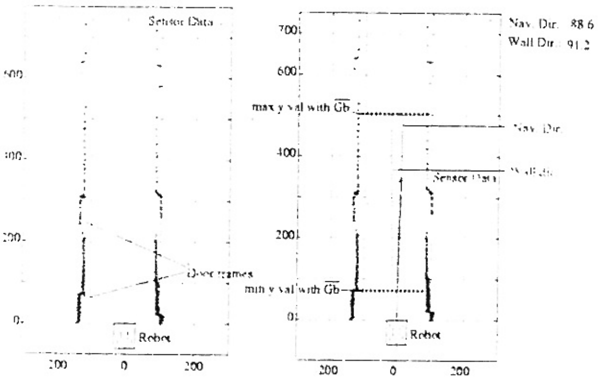

Fig. 4. Laser scan of a corridor. Left: Raw data. Right: Evaluated data. $\bar{G}\bar{b}$= average corridor width, Nav.= navigation orientation, Wall= wall orientation.

3.4 *The feature layer*

Real time feature detection algorithms based upon the sensor data derived from layer I have to determine several features. Here real time means in less than 60 ms. For example the orientation of walls or the position of obstacles has to be extracted as precisely as possible. Mainly time consuming line tracking algorithm are used to structure the environment (McLaughlin and Alder 1998, Arras *et al.* 1996).

3.4.1. *The ROCO algorithm* Now the ROCO (Rapid Orientation Calculation in Office environments) algorithm is described. Corridors in office environments usually are limited by two parallel walls (Figure 4). The walls are not necessarily straight, but can also run circularly. From the parallelism of the walls it follows that the distance between the two walls is constant. This information can be used in order to implement a fast and particularly robust algorithm. The intention is to calculate the width of a corridor or office from a laser scan with horizontally facing points. For this two discrete lists of integers, named $wall_{left}$ and $wall_{right}$ are required and have to be initialized with suitable values. The resolution of the lists may be for example 10 cm in y-direction, so 10 meters can be represented with 100 values. The two lists are needed for the collection of the smallest or largest x-values of the points P(x_i, y_i), $i = 1..n$ respectively. The index of the list corresponds to the y coordinate. The effort for this operation is only O(n). Now, the two lists mainly contain discreet points of the right and left walls. In the next step for each list index (y-position) the difference between $wall_{left}$ and $wall_{right}$ is computed and a frequency distribution of the differences is calculated.

The most frequent difference corresponds to the average corridor width $\bar{G}\bar{b}$, i.e. to the demanded width of the corridor. It can be compared with the value for the width of the corridor in the world model. The door frames (niches) form a local maximum in the frequency distribution. The effort at this point is still O(n), because all the operations are computed in the main loop over the laser scan points n.

By means of the lists $wall_{left}$ and $wall_{right}$ and the width of the corridor $\bar{G}\bar{b}$ it is possible to calculate also the orientation of the two walls: Therefore, we search for the largest and the smallest list entry (y-values), for which the difference of the two horizontal list entries corresponds to the width of corridor $\bar{G}\bar{b}$ (Fig. 4 right).

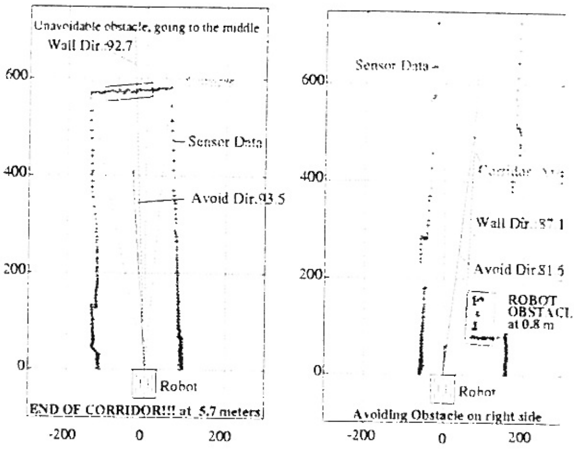

Fig. 5. Data of the ROCO algorithm (layer II). Left: End of corridor. Right: Avoiding a second moving robot.

The figure shows also that in contrast to standard polygon fitting approaches larger distances between the points y_{max} and y_{min} are found since door frames, niches and protrusions do not limit the fitting through y_{max} and y_{min}. If the distance between both points increases to more than five meters, the maximal error decreases to 0.6^o and the average error to 0.2^o. Thus, this local orientation can be used also for relocalization of the autonomous robots (Borenstein and Feng 1996). The distance between the y_{max} and y_{min} is also quality index for the correctness of the orientation. If the raw laser data is transformed with orientation of the previous scan then the calculation is again improved. After the transformation of x,y coordinates the walls are represented almost vertically (Figure 4).

The local orientation of the robot is calculated from the orientation of the wall with 180^o- orientation of wall. If for instance the mobile vehicle has to drive centered within the corridor, then a navigation orientation is calculated for the point $P((x_{right} - x_{left})/2 + x_{left}, y_{max})$. Here x_{left}, x_{right} are the values from the two lists $Wall_{left}$ and $Wall_{right}$ at y_{max}.

A simple PID controller is sufficient to follow this orientation. There is no need to consider the distance from the wall by the controller. Of course it is also possible to use arbitrary points between $P(x_{left}, y_{max})$ and $P(x_{right}, y_{max})$. Further informations can be gained in a simple manner with the ROCO algorithm. These informations can be useful for the robot navigation (Porrill 1988) e.g. if any deviations from the average width of the corridor occur while passing through

20

the two lists $Wall_{left}$ and $Wall_{right}$, this indicates niches, protrusions, end of corridor or obstacles in the passage. Niches, protrusions and ends of corridors may be used directly as natural landmarks for the robot relocalization by means of the world model. If obstacles are detected an alternate orientation can immediately be calculated from the knowledge of the position of the right and left gap (e. g. through the center of one gap) respectively. The distance to the obstacle is obtained simultaneously from the index of the list entry (Y-coordinate), thus the warning and protection zones from the raw data layer I can be verified.

With the size of the obstacle the robot can estimate a hypothesis about the object itself (Kreuziger and Wenzel 1994). Human obstacles (Fig. 6) and robot obstacles (Fig. 5 right) differ obviously. An end of a corridor is an unavoidable obstacle (Fig. 5 left), so the ARIADNE robots are led to the center of the corridor to pass the fire doors or to turn around. Otherwise our vehicles keep to the right, which is customary for Germany. This means they have a distance of 50 cm from the wall.[1]

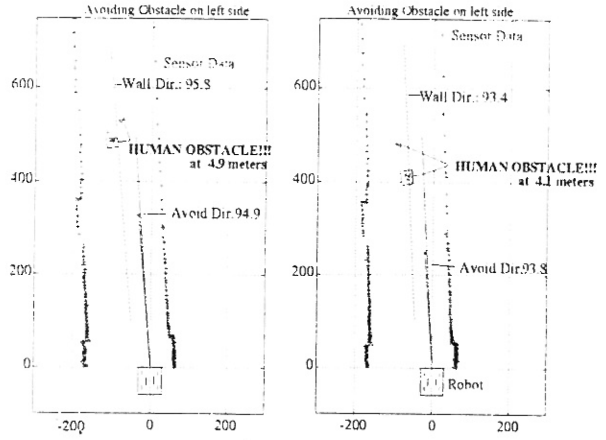

Fig. 6. Human obstacle in the scan.

3.5 *The intermediate layer*

The **intermediate layer** is the interface between data from the real sensors and information from the virtual sensors of the 2D model. Also different features from different sensors a combined at this layer (sensor fusion). It contains both, symbolic and subsymbolic information. These are combined using a fuzzy-matcher. Therefore the features from layer II are compared with 2D objects of layer IV. Especially for indoor environments the number of important objects is limited (Kreuziger and Wenzel 1994).

3.6 *The reasoning layer*

The main task of **reasoning layer IV** is planning (Kautz and Selman 1996). We distinguish three different levels of planning.

- At the highest level a large number of jobs has to be scheduled to the different robots. This NP-hard problem is known as the pickup and delivery problem with time windows (PDPTW). We implemented an approximation strategy called "shortest remain" which has a time consumption O(n*log n), n = 2*number of jobs. The effort is independent for preemptive and non-preemptive strategies respectively, i.e. for strategies which can interrupt a current job or not. The topological part of the world model together with estimated time values, e.g. waiting for an elevator, is the basis for the calculation of a good robot path[2].
- At the next level one determined job, which consists of two events (start point, end point), has to be planned. A graph latitude search of the topological part of the world model generates the path in O(n), n = number of corridors. This graph search is known as the standard planning approach, global or wide area planning.
- At the lowest level a list of actions and codes, e.g. for opening a door, is generated for each corridor on the base of the geometric description of the corridor in the world model. The effort depends upon the number of objects in a corridor which usually is equal to the number of office doors. This type of planning is known as local planning.

The interconnection between the global/local planning and the fuzzy navigator is done by fuzzy state variables (Surmann *et al.* 1996). A fuzzy state variable expresses a navigation advise in a corridor. The variables are fuzzy because they express the strength of the advice.

3.7 *The visualization layer*

The **visualization layer V** is important for the interaction with users (human machine interface) and for the design and maintenance of the robots by skilled operators. Therefore a 2D and 3D representation is generated from the world model. The 2D representation of the building, e.g. as gnuplot data, is used for simulation and design of the robot behaviors. In addition to the representation data, an apache web server and a java applet implement an interface for a position tracking of the robots in the WWW. The 3D representation (VRML) is used for a more realistic position tracking, for tele-operating the robots in case of unforeseen situations and for overcoming system errors (Surmann and Theißinger 1999). A VRML browser with EAI interface (e.g. cosmo player) visualizes the

---

[1] At http://lamu.gmd.de:8080/robobld several examples of laser scans are recorded while the robots navigated through our corridors. The feature detection is also included.

[2] For more details please visit http://lamu.gmd.de:8080 and enter jobs in the transport form

behavior of the robots and increases thereby the user friendliness of the system. An administrator can check this representation and the live camera pictures to detect, whether a robot reaches a critical area, e.g. an area crowded by persons or an area of construction work.

4. RESULTS

The whole concept of distributed intelligence presented in this paper was implemented and tested with the three robots drive that in the GMD-ROBOBENCH, a test environment of three floors connected by an elevator at Schloss Birlinghoven, GMD. The test data was collected during a 7 day demonstration at the CeBit (Hannover Fair) and during various tests at GMD.

In layer I a data rate of 16-17 Hz (60ms, 126 distance values) is achieved. The processing of one scan takes 2 ms, including the reliability and zone tests. In the feature extraction layer II the orientations and positions of obstacles could be updated with a data rate of 200 Hz, so they are only limited by layer I. The average length of the distance which is used to calculate the orientation in our environment at GMD is 5 m and the robustness of the feature extraction varies from 92.8% - 98.7%. The accuracy of detected feature positions is up to 10 cm. The recognition time for more complex features e.g. doors is included in the ROCO algorithm and needs no further computation time.

The position of the robot is updated every driven meter only if no contradicting information has been generated. It is always exact up to 15 centimeters. The accuracy of the wheel encoders is 5 cm on a 30 m corridor. The 3D-model and information in level V are only computed on demand. The position update in the 3D model is limited by the Internet access. The default is 1 hz.

5. CONCLUSION

In this paper we first presented a multi layer sensor architecture for autonomous mobile service robots in indoor environments. The layer architecture integrates modules of average or limited intelligence and a spatial data base for indoor environments. The world model as well as the evaluation modules of the sensor data are based on natural landmarks and could be evaluated in real-time. Secondly, a very fast and robust basic feature extraction algorithm was presented. The robots get all needed data at different levels of abstraction in various layers. According to the sensor layers different levels of control as well as different types of data exchange are realized in order to meet real-time constraints. The key features of the sensor architecture are reusability, modularity and portability to other multistory buildings as well as expendability with different sensors (sensor fusion). Needless to say a lot of work remains to be done to achieve further intelligence. Future work will improve the long term robustness of the total system.

6. REFERENCES

Arkin, Ronald C. (1998). *Behavior-based Robotics*. The MIT Press, Cambridge, Massachusetts, London, England.

Arras, Kai, S.J. Vestli and N. Tschichold-Guerman (1996). *Echtzeitfaehige Merkmalsextraktion und Situationsinterpretation aus Laserscannerdaten*. pp. 57–66. Schmidt G., Freyberger F. (eds.).

Borenstein, J. and L. Feng (1996). Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation* (7), 869–880.

Engelberger, Joseph F. (1993). Health-care robotics goes commercial: the 'HelpMate' experience. *Robotica* **11**, 517–523.

Graf, R. and P. Weckesser (1998). Roomservice in a hotel. In: *IFAC Symposium on Intelligent Autonomous Vehicle, Madrid*. pp. 641–647.

Kautz, H. and B. Selman (1996). Pushing the envelope: planning, propositional logic, and stochastic search. In: *Proc. of the 13th National Conference on Artificial Intelligence (AAAi-96). Portland Or.*

Konolige, K. and K. Myers (2000). *The Saphira Architecture for Autonomous Mobile Robots*. MIT press.

Kreuziger, J. and W. Wenzel (1994). Learning generic object descriptions for autonomous robot systems. In: *Proceedings of the Fifth International Symposium on Robotics and Manufacturing, ISRAM'94, Maui.*

McLaughlin, Robert A. and Michael D. Alder (1998). The hough transform versus the upwrite. *IEEE. Transactions on Pattern Analysis and Machine Intelligence.*

Pauly, Michael, Hartmut Surmann, Marion Finke and Nang Liang (1998). Real-time object detection for autonomous robots. *Informatik Aktuell. Autonome Mobile Systeme, 14. Fachgespräch, Springer-Verlag* pp. 57–64.

Pfeifer, Rolf and Christian Scheier (1999). *Understanding Intelligence*. The MIT Press, Cambridge, Massachusetts, London, England.

Porrill, J. (1988). Optimal combination and constraints for geometrical sensor data. *The International Journal of Robotics Research* (6), 66–77.

PSLT (1998). *FTS-Anlagen-Analyse Europa: Technischer Bericht*. PSLT Hannover.

Schmersal EOT (1999). Lss 300: Telegramm zwischen benutzerschnittstelle und sensorsystem v.1.14. In: *Schmersal EOT, Freibug, Germany.*

Schraft, R.D. and H. V[illegible] (1996). *Serviceroboter, Innovative Technik in Dienstleistung und Versorgung*. Springer-Verlag, Berlin, Heidelberg.

Surmann, Hartmut and Markus Theißinger (1999). Robodis: A dispatching system for multiple autonomous service robots. In: *Proceedings of the FSR'99, Robotics Applications for the Next Millenium, Pittsburgh, PE.* pp. 168 – 173.

Surmann, Hartmut, Jörg Huser and Jens Wehking (1996). Path planning for a fuzzy controlled autonomous mobile robot. In: *Fivth International Conference on Fuzzy Systems, FUZZ-IEEE 96, New Orleans, USA*. pp. 1660 – 1665.

Surmann, Hartmut, Jörg Huser and Liliane Peters (1995). A fuzzy system for indoor mobile robot navigation. In: *Fourth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 95, Yokohama, Japan.* pp. 83–88. Distinguished with *Robot Intelligence Award.*

Thrun et al., Sebastion (1999). Experiences with two deplyed interactive tour-guide robots. In: *Proceedings of the FSR'99, Robotics Applications for the Next Millenium, Pittsburgh, PE.* pp. 37 – 42.

Vestli, S.J. (1996). Mops - a system for mail distribution in office-type buildings. *Service Robot: An International Journal* (2), 29–35.