A five layer sensor architecture for autonomous robots in indoor environments

Hartmut Surmann and Antonio Morales

Abstract—**Autonomous mobile service robots for transportation tasks in indoor environments e.g. multistory buildings, have to act in normal dynamic environments but with a huge number of components. The robots total repertoire of skills is high according to the complexity of the building and its respective task. Difficult tasks can only be achieved on the base by immediate sensing of the environment. This paper describes a five layer sensor architecture with an integrated world model for multistory buildings. In contrast to grid based approaches we use a feature based approach. The sensor architecture as well as the evaluation modules of the sensor data are based on natural landmarks. The key features of the sensor architecture are reuseability, modularity and portability to other multistory buildings as well as extendibility with different sensors.**

Keywords—**robot architectures, sensor architectures, collision avoidance and sensor-based control, robot sensing and data fusion, behavior-based robotics**

I. INTRODUCTION

Prognoses at the beginning of the nineties expected in the year 2000 a number of about 50.000 independently operating autonomous service robots in different areas of production and in the service sector [1]. The reality looks different. In industrial environments Guided Vehicles (GV) i.e. vehicles guided by a magnetic or optical line are standard [2]. However, in practice autonomously acting mobile service systems - i.e. systems not restricted to follow a line - are used extremely rarely. Although many research groups work on autonomous mobile robots for several years – particularly on mobile systems for transportation tasks e.g. [3], [4], [5], [6]. The references contain only real robots not toy robots or scientific prototypes which in practice could not transport something. One main reason for the gap between prognoses and reality is the complexity of the environment in which the robots have to act, particularly for multistory buildings. Current robot control architectures are not able to handle the complexity of multistory buildings [7], [8]. The robots can only operate in one level like the museum robots [9]. Multistory buildings are dynamic environment with elevators and fire doors which leads to special demands for the mobile units.

Todays robot control architectures like saphira [8] handle only the low level driving instruction but not high level sensor information processing with different sensors e.g. for handling elevators or fire doors. However in multistory buildings an integrated sensor layer architecture is demanded for the mobile units. So, this paper presents the integrated layered sensor architecture with a world model that considers all this boundary conditions. The described sensor architecture is only a (very important) part of a robot control architecture but not a control architecture by itself. Some layers are well known in literature and only briefly described. The layers of the sensor architecture are connected to different robot modules e.g. to the behavior based module and planning module, modules for the design of robot control software, the simulation of robot behaviors and robot behavior anticipation which are not part of this paper.

Due to the size of office buildings a feature based approach is selected. Grid based approaches have been successfully used for museum guides [9], [10] on one level, but lead to space and/or computing time problems for multistory buildings. Other typical aspects of this class of applications such as planning and mutirobot cooperation are implemented but not described here [11].

Autonomous mobile service robots that have to execute transportation tasks in multistory buildings have to react flexibly to various boundary conditions:

- The spacious environment contains a lot of components, that usually are only of a few different types. So complexity is originated by the large number of components (offices).
- In the corridors that lead to the offices usually a few humans and other robots are moving, i.e. the dynamical complexity is small or medium.
- The infrastructure of the buildings including e.g. elevators, fire doors or cameras has to be used to reach a given target, since these components represent natural landmarks for repositioning.
- Permanently changing requests of the users have to be fulfilled fast and comfortably. In particular the user wants to use his known working environment for his interactions, e.g. his Internet browser [12].
- The robots have to be save and reliable. i.e. a large system with numerous robots must support treatment of unforeseen situations and overcome system errors.

The paper is structured as follows: The second and third section give a brief introduction to our robots and the experimental environment. In the fourth section we present the world model which is the base of layer III - V. The five layer sensor architecture and the different layers are described in the fifth section. We conclude the paper by giving a prospect of our future research activities.

II. THE ARIADNE SERVICE ROBOT TEAM

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

GMD AiS - German National Research Center for Information Technology, Schloß Birlinghoven, D-53754 Sankt Augustin, Germany, surmann@gmd.de

Jaume-I University, Robotic Intelligence Laboratory Campus Riu Sec, Edificio TI, E-12071 Castellon, Spain, morales@inf.uji.es

Published at: International symposium on robotics and automation ISRA'2000, Monterrey, N.L., Mxico, Nov. 2000, pp. 533–538

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 micro-controllers 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 [13], 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 [14].

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.

A. The experimental environment GMD-ROBOBENCH

The robots currently move in the GMD-ROBOBENCH, a typical multistory H-shaped building of about 1600 m^2 (Fig. 2) which serves as a general prototype for other multistory buildings. The building is departed into 84 office rooms and 15 corridors on 3 levels reachable by elevators. All elevators and 17 doors can be automatically controlled 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.

B. World model

The world model i.e. 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 by the robot mission. On the other hand the spatial database influences the selection of the sensors and the feature extraction algorithms.

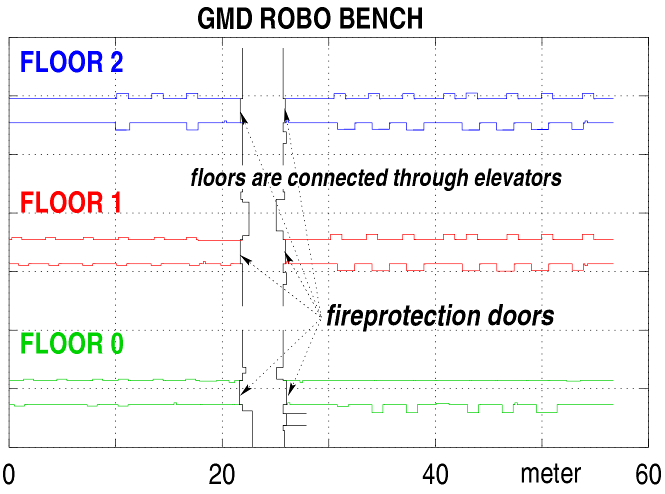


Fig. 2. Layout of the prototype environment.

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 [15]. 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 [12].

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 in the XML specification. Therefore robots or other clients or servers can exchange and update the spatial database via the Internet.

The “corridor” structure is 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 base 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

```
<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.

...), 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.

One major point is that different internal descriptions are generated on 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.

III. THE FIVE LAYER SENSOR ARCHITECTURE

Figure 4 shows the five layers of the architecture. The navigation behaviors of the robots can be connected to each layer of the sensor architecture. The lower layers are mainly used for local navigation and the upper layers for global planning and the human machine interface (HMI).

The **data layer** extracts the raw data, e.g. up to 500 local distance values (vendor Schmersal or Sick) every 30 ms in the range from 0-180^o [16], [17]. 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 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).

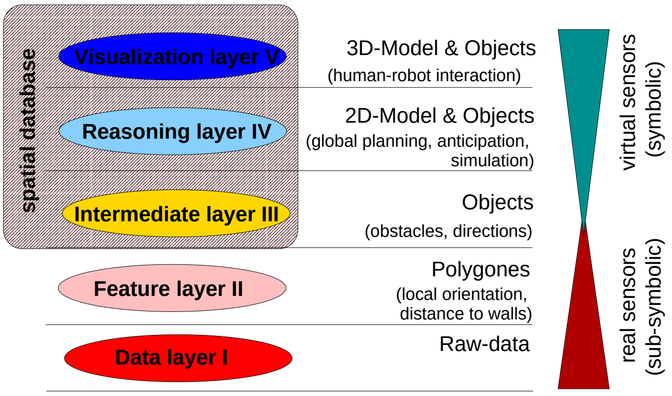


Fig. 4. Five layer sensor architecture for guiding service robots in office environments.

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.

On the lowest levels of the model particularly robust and real time capable algorithms are implemented. Other sensors are added to this level.

The **feature layer** uses the raw data to detect polygons, local orientations of walls and rectangular obstacles [18]. Therefore, the ROCO algorithm (Rapid Orientation Calculation in Office environments) is used. Real time feature detection algorithms based upon the sensor data derived from layer I have to compute 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.

The key idea is that corridors in office environments usually are limited by one or two walls (Figure 5). 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 discrete 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

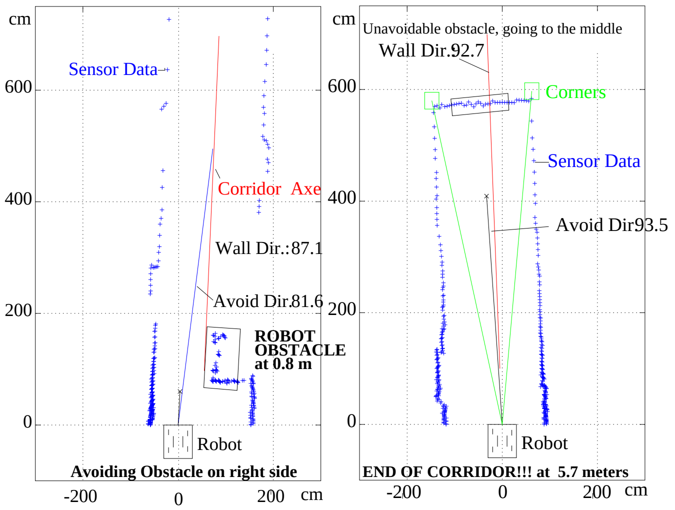


Fig. 5. Data of the ROCO algorithm (layer II): left) Avoiding a second moving robot, right) End of corridor.

The most frequent difference corresponds to the average corridor width $\overline{gb}$, 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 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 $\overline{gb}$ 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 $\overline{gb}$.

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 by 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 [19]. 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.

The local orientation of the robot is calculated from the orientation of the wall with 180$^o -$ orientation of wall. If f.i. the mobile vehicle has to drive in the center of 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}.

Further information can be gained in a simple manner with the ROCO algorithm. This information can be useful for the robot navigation [20] e.g. if any deviations from the average width of the corridor occur while passing through 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.

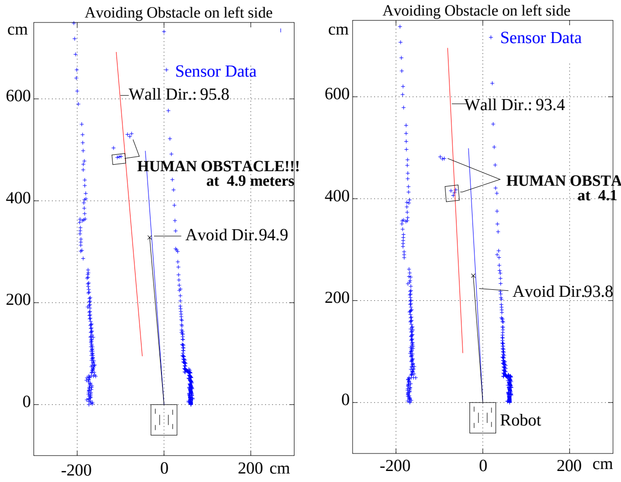


Fig. 6. Human obstacle in the scan.

Knowing the size of the obstacle the robot can estimate a hypothesis about the object itself [21]. Human obstacles (Fig. 6) and robot obstacles (Fig. 5 left) differ obviously. An end of a corridor or a closed fire door is an unavoidable obstacle (Fig. 5 right), 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]

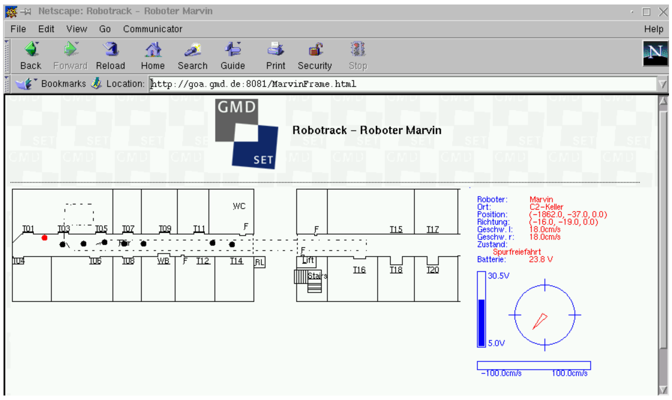
The **intermediate layer** is the interface between the data from the real sensors and the information from the virtual sensors of the 2D model. Also different features from different sensors a combined at this layer (sensor fusion). It contains both, symbolic and sub-symbolic information. These are 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 [21].

The main task of **reasoning layer IV** is planning [15]. 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 form preemptive and non-preemptive strategies respectively, i.e. 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 base for the calculation of a good robot path.
- At the next level one selected job, which consists of two

---

[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.

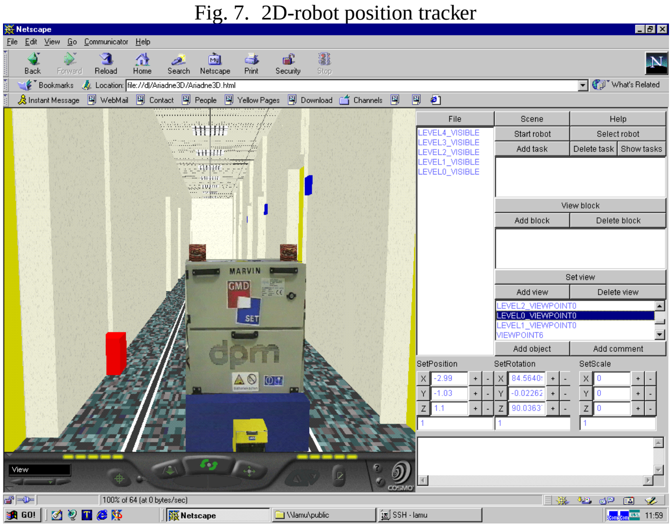


http://goa.gmd.de:8081/MarvinFrame.html

Robotrack – Roboter Marvin

Roboter: Marvin
Ort: C2–Keller
Position: (−1862.0, −37.0, 0.0)
Richtung: (−16.0, −19.0, 0.0)
Geschw. l: 18.0cm/s
Geschw. r: 18.0cm/s
Zustand: Spurfreiefahrt
Batterie: 23.8 V

Fig. 7. 2D-robot position tracker



file://d|/Ariadne3D/Ariadne3D.html

Fig. 8. 3D view of the robot while driving through the building.

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 in the literature 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 [22]. A fuzzy state variable expresses a navigation advice in a corridor. The variables are fuzzy because they express the strength of the advice.

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 (Fig. 8 and 7)

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 [12]. A VRML browser with EAI interface (e.g. cosmo player) visualizes the 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.

IV. RESULTS

The concept was implemented and tested with the robots of the ARIADNE team. The three robots drive in the GMD-ROBOBENCH, a test environment of 3 floors connected through 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 the 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 were generated with a data rate of 200 Hz, so they are only limited by layer I. The average length of a line which can be 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 a detected features 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 relocalization of the robot is updated every driven meter only if no contradicting information was generated. The position 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 any 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.

V. CONCLUSION

In this paper we have 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. The robots get all needed data at different levels of abstraction at the different 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 reuseability, modularity and portability to other multistory buildings as well as the extendibility with different sensors (sensor fusion).

The concept has been validated during various tests at GMD using the three robots of the ARIADNE team. Future work will improve the long term robustness of the total system.

VI. ACKNOWLEDGE

The authors would like to thank Daum & Partner GmbH, Schmersal EOT, the members of the ARIADNE team and Claudia Hirsch for supporting this work.

REFERENCES

[1] R.D. Schraft and H. Volz, *Serviceroboter, Innovative Technik in Dienstleistung und Versorgung*, Springer-Verlag, Berlin, Heidelberg, 1996.

[2] PSLT, *FTS-Anlagen-Analyse Europa: Technischer Bericht*, PSLT Hannover, 1998.

[3] Joseph F. Engelberger, “Health-care robotics goes commercial: the ‘HelpMate’ experience,” *Robotica*, vol. 11, pp. 517–523, March 1993.

[4] P. Weckesser R. Graf, “Roomservice in a hotel,” in *IFAC Symposium on Intelligent Autonomous Vehicle, Madrid*, 1998, pp. 641–647.

[5] und H.-B. Kuntze (Hrgs.) E. Prassler, R. Dillmann, *Robotik in Deutschland – Lehre, Forschung und Entwicklung*, Shaker-Verlag, Aachen, 1998.

[6] S.J. Vestli, “Mops - a system for mail distribution in office-type buildings,” *Service Robot: An International Journal*, , no. 2, pp. 29–35, 1996.

[7] Ronald C. Arkin, *Behavior-based Robotics*, The MIT Press, Cambridge, Massachusetts, London, England, 1998.

[8] K. Konolige and K. Myers, *The Saphira Architecture for Autonomous Mobile Robots*, MIT press, 2000.

[9] Sebastion Thrun et al., “Experiences with two deplyed interactive tour-guide robots,” in *Proceedings of the FSR’99, Robotics Applications for the Next Millenium, Pittsburgh, PE*, 29.8. - 31.8.1999, pp. 37 – 42.

[10] D. Fox J-S. Gutmann, W. Burgard and K. Konolige, “Experimental comparison of localization methods,” in *International Conference on Intelligent Robots and Systems, Victoria, B.C.*, Oct. 1998.

[11] Hartmut Surmann and Antonio Morales, “Scheduling service robot missions,” in *ECAI 2000, European Conference on Artifical Intelligence, Workshop 5: Current Issues in Spatio-Temporal Reasoning*, 2000, pp. 25–34.

[12] Hartmut Surmann and Markus Theißinger, “Robodis: A dispatching system for multiple autonomous service robots,” in *Proceedings of the FSR’99, Robotics Applications for the Next Millenium, Pittsburgh, PE*, 29.8. - 31.8.1999, pp. 168 – 173.

[13] Hartmut Surmann, Jörg Huser, and Liliane Peters, “A fuzzy system for indoor mobile robot navigation,” in *Fourth[13] IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 95, Yokohama, Japan*, 20–24 Mar. 1995, pp. 83–88, Distinguished with *Robot Intelligence Award*.

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

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

[16] Schmersal EOT, “Lss 300: Telegramm zwischen benutzerschnittstelle und sensorsystem v.1.14,” in *Schmersal EOT, Freibug, Germany*, 1999.

[17] Sick optic electronic, “Pls: Definition der telegramme zwischen benutzerschnittstelle und sensorsystem über rs 422 / rs 232,” in *http://www.sickoptic.com/laser.htm*, 1996.

[18] Jean-Claude Latome Kenneth Y. Goldberg, Dan Halperin and Randall H. Wilson, *Algorithmic Foundations of Robotics*, AK Peters, Wellesley, MA, 1995.

[19] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Transactions on Robotics and Automation*, , no. 7, pp. 869–880, 1996.

[20] J. Porrill, “Optimal combination and constraints for geometrical sensor data,” *The International Journal of Robotics Research*, , no. 6, pp. 66–77, 1988.

[21] J. Kreuziger and W. Wenzel, “Learning generic object descriptions for autonomous robot systems,” in *Proceedings of the Fifth International Symposium on Robotics and Manufacturing, ISRAM’94, Maui*, 1994.

[22] Hartmut Surmann, Jörg Huser, and Jens Wehking, “Path planning for a fuzzy controlled autonomous mobile robot,” in *Fivth International Conference on Fuzzy Systems, FUZZ-IEEE 96, New Orleans, USA*, Sept. 1996, pp. 1660 – 1665.

Hartmut Surmann received his diploma in computer science from the Department of Computer Science, University of Dortmund and his Ph.D. in electrical engineering from the Faculty of Electrical Engineering, University of Dortmund, Germany, in 1989 and 1994, respectively. He is a senior researcher at GMD Autonomous intelligent System Institute in St. Augustin, Germany. His primary research interests are robotics and computational intelligence. He has published several papers in the area of design, application and hardware for robotics and fuzzy logic controllers. He is a member of the GI and VDE.

Dr. Surmann received the FUZZ-IEEE/IFES’95 robot intelligence award for a fuzzy system for indoor mobile robot navigation and the Ph.D. award for his thesis from the German AI institutes in 1996.

Antonio Morales received a B.Sc. in computer Engineering from the Jaume I University (Spain) in 1996. He is currently a graduate student at the Robotic Intelligence Laboratory of the Computer Science Department in the Jaume I University, where he holds “SEUI - Programa Sectorial de Formación de Profesorado Universitario y Personal Investigador en España” studentship. His current research interests are in the field of robotics, control architectures, sensor and vision guided grasping.