Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 86831, 10 pages
doi:10.1155/2007/86831

Research Article

Autonomous Robot Navigation in Human-Centered Environments Based on 3D Data Fusion

Peter Steinhaus, Marcus Strand, and Rüdiger Dillmann

Institute for Computer Science and Engineering (CSE), University of Karlsruhe (TH), Haid-und-Neu-Straße 7, 76131 Karlsruhe, Germany

Received 1 December 2005; Revised 14 July 2006; Accepted 17 December 2006

Recommended by Ching-Yung Lin

Efficient navigation of mobile platforms in dynamic human-centered environments is still an open research topic. We have already proposed an architecture (MEPHISTO) for a navigation system that is able to fulfill the main requirements of efficient navigation: fast and reliable sensor processing, extensive global world modeling, and distributed path planning. Our architecture uses a distributed system of sensor processing, world modeling, and path planning units. In this arcticle, we present implemented methods in the context of data fusion algorithms for 3D world modeling and real-time path planning. We also show results of the prototypic application of the system at the museum ZKM (center for art and media) in Karlsruhe.

1. INTRODUCTION

The problem of navigating mobile systems in dynamic indoor environments is one of the basic problems in the area of mobile robots. Starting not long ago, as a first trend, service robots and especially humanoid robots address more and more human-centered working spaces, so the problem of efficient navigation in such dynamic scenarios seems to be very important. As a second trend, sensor and computing technologies become cheaper, faster, and smaller, enabling the design and implementation of huge sensor networks in the so-called "intelligent buildings" (smart houses). As mobile robots can also be seen as actuators of these intelligent buildings, it seems almost intuitive to combine both techniques, mobile robots and sensor networks, to solve the problem of efficient navigation in dynamic human-centered indoor environments.

Looking at the enormous amount of previous works already done in the field of navigation system research, almost all approaches can be categorized with respect to their architectures in one of the following categories.

Autonomous onboard navigation

The problem of autonomous navigation without any help of external sensor systems is addressed since the beginning of mobile robotics. The used approaches can be divided into the classes of functional/cybernetic, behavioristic, and hybrid approaches. A typical functional approach can, for example, be found in [1, 2], where global and local planning modules work on a 2D geometrical and topological map to plan subgoals for the mobile systems, using current sensor data to adapt paths while targeting the subgoals. A behavior-based approach is, for example, given in [3–5], where a set of logical and physical sensor systems is situation-dependent activated to search for edges or obstacles. A hybrid approach which combines the functional deliberative aspects of path planning with the reactive and behavioristic concepts of path control can, for example, be found in [6]. Here, a coordination instance is used to activate the different behaviors of the CAIR-2 robot system on an originally planned path. The characteristic problem of all these approaches is that the amount of environment information a mobile system can acquire, at a specific point of time, is limited by the perspectives of the sensor systems, the sensor systems characteristics, and situation-specific occlusions.

Autonomous multi-robot navigation

In multi robot environments (decoupled multi robot systems), it is possible to see every mobile system as a part of a distributed sensor network consisting of several mobile sensor agents. If these sensor agents are able to communicate (e.g., if they are near enough together), they share their current sensory data to achieve more complex environment descriptions by sensor integration or fusion. Every robot is working autonomously on its own environment model. There is no central coordination unit. A typical implementation of this basic idea can be found in the CyberScout project (see [7–9]), where 2D polygonal environment information is distributed among the partners during environment exploration processes.

There are some similar approaches that all share the same main problem: in dynamic scenarios the mobile agents are not able to generate a complete observation of the environment at every point of time, as they are moving around to their own specific targets.

Smart-house navigation

The main idea of these approaches is to use an intelligent environment consisting of embedded distributed sensor networks in combination with centralized or also distributed computing power to guide mobile systems in dynamic indoor environments. The smart house can be responsible for almost all navigation functionalities like localization, sensor data acquisition, environment modeling, path planning or collision avoidance. For example, in [10] a distributed camera network is used in combination with a centralized computing instance to build 2D obstacle maps for use by mobile systems using the freespace approach. The scalability of this approach to wide scenarios is limited due to the monolithic system design. A similar approach using distributed intelligent networked devices (DINDs) in a warehouse environment is examined in the Intelligent Space project (see [11–13]), where a 2D environment/obstacle model is acquired by cameras. The mobile robots are only equipped with odometric sensors. Most of the reviewed approaches use only 2D models for environment representation instead of more realistic 3D models. Mobile sensor systems, as additional sensor data sources, are not included.

Smart-house multi robot navigation

In this category, additional mobile sensor systems are used as an extension of the smart-house navigation concept to improve the environment model degree of detail and help in cases of occlusion. Here, stationary and mobile sensor systems have to be fused to result in a global environment model. There are just a few works in this area. One is in the field of robot soccer (see [14–17]), where the soccer playing robots share their sensor data with each other and a central external instance. Another is in the approach of the intelligent data carrier (IDC) (see [18–21]), where the external distributed components store regional environment information that is gathered by mobile systems. Even here, only 2D models are used for environment modeling.

We propose a navigation system approach belonging to the smart-house multi robot navigation category, combining stationary and mobile sensor sources to achieve a complete 3D environment model by fusing heterogeneous sensor data. This environment model is used for real-time path planning and path adaptation. In this article, we concentrate on the homogeneous camera network processing, the data fusion approach, and the resulting path planning method. Some results on data fusion of stationary and mobile sensors can be found in [22].

Section 2 gives a short overview about the navigation system architecture, Section 3 describes the methods used for the stationary camera sensor network, Section 4 gives an introduction to the data fusion algorithm, Section 5 shows the theoretical background of the path planning and path adaption processes and Section 6 demonstrates some experimental results.

2. NAVIGATION SYSTEM ARCHITECTURE

The navigation system architecture has the following characteristics:

(i) use of a heterogeneous distributed sensor network consisting of stationary and mobile components (scalability, completeness);
(ii) stationary sensors build up a homogeneous color camera network;
(iii) use of 3D laser range sensors as mobile sensors;
(iv) distributed sensing, environment modeling, data fusion and, path planning;
(v) data fusion to 3D environment models;
(vi) hierarchical real-time path planning and path adaptation approach.

We propose the use of specialized local-area processing units (LAPUs). They perform the observation of their local-area, the update of their local environment model, the short-time prediction, and the short-way planning. Every LAPU is connected to a robot control unit (RCU) that contacts the robots via wireless ethernet to send commands and receive status messages and preprocessed 2D and 3D sensor data. The combination of the LAPUs with their dedicated RCUs is called a global area processing unit (GAPU). Figure 1 shows an example of a GAPU with 6 LAPUs and 2 RCUs.

For global and local path planning, environment model enhancement, and sensor data fusion a neighborhood graph of the observed areas gives the basic network topology of our sensor network. Fast ethernet connections between LAPUs themselves and also to RCUs ensure the validity of interchanged sensor, path, and administrative data.

Figure 2 shows the complex architecture of a LAPU. Beginning in the sensor layer, the local sensor module (LSM) tracks the positions of obstacles and robots during their way through the sensor area. The resulting data is given to the local environment modeling module (LEMM) and the local environment prediction module (LEPM) in the environment layer. These build up the current- and short-time future environment representations for the local area planning module (LAPM) and the global planning module component (GPMC) in the planning layer. The LEPM is responsible for choosing a good path inside the local sensor area while

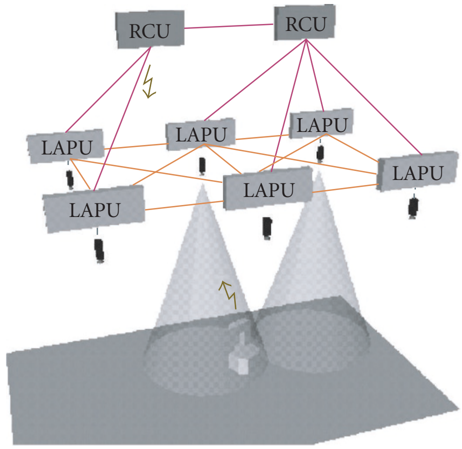


FIGURE 1: Distributed structure of the navigation system.

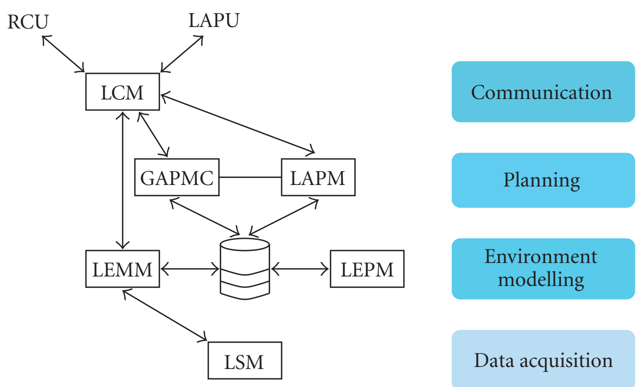


FIGURE 2: Components of a local area processing unit (LAPU).

the GPMC is planning a good path to the destination LAPU. The local communication module (LCM) is responsible for handling incoming and outgoing sensors, planning and administrative data to the other LAPUs and the dedicated RCU.

More details about the architecture, their components and the communication between modules are given in [23].

Figure 3 gives a simplified description of the data flow between the different components of the navigation system architecture. It can be seen that every sensor source results in a single unique view on the global environment and that these views are fused to the current dynamic 3D environment model.

3. STATIONARY ENVIRONMENT VIEWS

In our setup, the LAPUs build a homogeneous sensor network which is equipped with standard CCD color cameras. These cameras are fixed to the ceiling of our laboratory and result in different perspective views of the environment. As the camera positions are stationary, difference image analysis can be used to find obstacles in the image and to compute the corresponding world-model representation. Color cameras are used because the neon lights in the laboratory generate shadows that cannot be suppressed in gray images.

The difference-image algorithm works as follows: the digitized color image is acquired in RGB format. The first step of the algorithm is to transform this RGB color space format image into HLS (hue, lightning, saturation) format. The reference image is stored already in HLS. To suppress shadows, the heuristic is that a current image pixel representing the shadow of an object on the unchanged background has approximately the same hue and saturation values as the corresponding pixel in the reference image. The lightning component differs significantly. So the lightning component is ignored and two difference images (hue, saturation) are generated. Every difference image has to be processed by morphologic opening and closing operations to fill gaps and eliminate isolated pixels. Therefore, for every difference image two binarized images are generated (with a high and a low threshold). The high threshold binary image does only hold pixels that have to be in the resulting binary image, but has many gaps in it. Therefore, it is processed by closing operations to fill these gaps. As not only the gaps get filled but the blob area grows in all directions, an AND-operation with the lower threshold binary image is computed to cut these areas. At the end, we have binary images representing the differences in hue and saturation. These binary images are combined by an OR-operation to get the resulting binary image.

On the binary image, a blob search is performed that results in the pixel contours of all obstacles in the camera area. These pixel contours are transformed into polygons by an iterative end point fit algorithm to reduce the complexity of the obstacle description. Using a special interpolation-based camera calibration, the mapping from pixel coordinates to world coordinates is done by a table look up in one step. The latency introduced through is very low due to the distributed approach. In world model coordinates, the contour polygons of the obstacles can be used to construct a boundary surface representation of every obstacle volume by knowing the position of the camera for every LAPU and building a view polyhedron for every obstacle. These obstacle volume estimations are transferred to the local environment model and to the LAPUs topologic neighbors. A description of the maximal possible view polyhedron of every camera sensor is also given to every LAPU after the camera calibration process. Details on the stationary environment views can be found in [22].

4. FUSION OF VIEWS

The environment model is organized as a 3D surface model. All objects are represented as set of triangles with corresponding information about object number, center of gravity, object trajectory, and so forth. The environment model module receives the obstacle polyhedrons from the local sensor layer, possibly from neighboring LAPUs sensor layers and all mobile sensors in their area and has to integrate and fuse these data to object representations. These polyhedrons are in general not convex but as we ignore the floor area

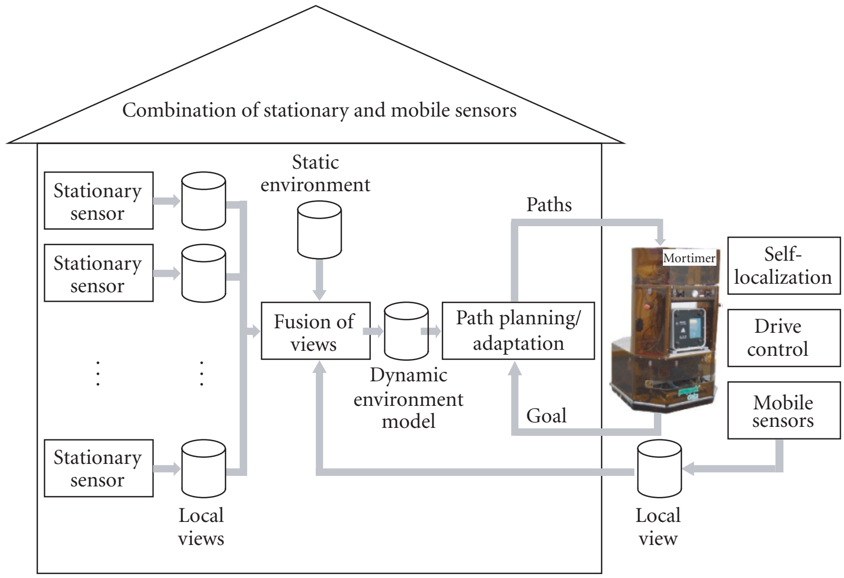


Figure 3: Simplified dataflow diagram.

every facet of the polyhedron is convex. Given overlapping sensor areas of different LAPUs, it is possible to reconstruct an approximation of the 3D scene by intersecting the different view polyhedrons. This is done by converting the triangle representation of the environment to a BSP tree (binary space partitioning tree). At the beginning, the first object of the first camera builds the root tree, then other objects of the first camera extend this tree. This tree is used to intersect the polyhedrons of another camera with the first camera. As a result, a more complex BSP tree is again the starting point for the next camera data fusion. When all polyhedrons of all cameras belong to the tree, the remaining triangles of the intersection can be read from the tree and give the corresponding environment model. As the fusion algorithm can only work in the intersection volume of the view areas of all cameras, an extension to the union volume has been implemented. It is possible to extend the local object polyhedron representation by a so-called neutral volume representation. This neutral volume represents the volume given by the difference of the union of all view volumes and the view volume of that specific camera.

The resulting triangle structure of the environment description is then analyzed with respect to triangle clusters to identify objects, to track them permanently, and to give the possibility of a short-time prediction of object paths.

The environment model data structure is called DU with

$$DU = \{(f_i, N_i) \mid i = 1, \ldots, n\}. \quad (1)$$

Here every convex facet f_i denotes a boundary representation of the occupied space, N_i is the corresponding normal vector denoting the inside/outside relation of the facet. Details on the fusion of views can be found in [22].

5. PATH PLANNING

The path planning module supervises the platforms movement to a specified goal point. Here, real time planning is necessary to ensure the validity of the path at any time. Additionaly, predictable narrow passages and dead ends should be identified and avoided. Our proposed path planner fulfills these requirements due to two properties.

(i) At every time step, the system provides either an efficient path from the current robot's pose to the goal or the information that there is no valid path in the current obstacle configuration.

(ii) The hierarchical system approach differentiates between global and local path planning tasks. Hereby, the path planning task can be distributed over several LAPUs and parallel calculations are possible.

A LAPU can have several different tasks, depending on the platforms position and goal.

Traversal

when traversing a LAPU a trajectory between two transfer points, specified by the global path planner, has to be generated.

Entry

a path has to be generated from the transfer point to the goal within the LAPU.

Exit

a path has to be generated from current location to the transfer point of a neighboring LAPU.

Internal drive

the local path planner generates a path between two points within the LAPU.

In every case, a path between two specified points has to be generated and maintained. The following theoretical notes refer to path planning and adaption from any start to any goal configuration.

First, the complex 3D environment model is reduced for high-speed processing of the path planner. In a second step, two parallel processes are initiated. In the main process, an initial path planner determines a valid path within the current environment configuration. This path is constantly modified due to moving obstacles by means of a dynamic path adaption. In parallel, another process tries permanently to find an alternative path, which is significantly more efficient than the current modified path.

5.1. Reduction of the environment model

The environment model consists of a 3D-occupied space description DU with

$$DU = \{(f_i, N_i) \mid i = 1, \dots, n\}. \tag{2}$$

To transform the model for realtime path planning, several steps are carried out.

(1) The robots geometry is removed from the environment representation. For planning a collision free path, every obstacle for a mobile platform r has to be considered. This means that, every robot geometry, except the robot r, has to be added to DU so that

$$DUR^r = DU \bigcup \left(\bigcup_{i, i \neq r} GM^i \right). \tag{3}$$

(2) In a next step, the environment model will be reduced to the information, which is really necessary for platform navigation. Therefore the platform r is approximated by a cylindric bounding box, whose parameters are determined in the platform geometries GM^r. All available information with z-coordinates bigger than the height h of the bounding box can be discarded, so that the resulting model is $DUR'^r(h)$.

(3) For the detection of a collision between a facet of the model and the bounding box of the platform, the resulting 3D model is projected onto the plane $z = 0$, so that

$$PDUR^r = \{(f_i, N_i) \mid i = 1, \dots, n,\ f_i \in \mathbb{R}^2,\ N_i \in \mathbb{R}^2\}. \tag{4}$$

(4) With a discretization of the LAPU area, collisions between the environment facets and the platform can be calculated very fast. This late discretization helps to keep the memory requirements low, since the discretization space is 2D and the discretization is restricted to the local LAPU area. The environment grid $UG(x, y)$ has a value $UG(x, y) = 1$ if the position is occupied, so that

$$\begin{aligned} UG(x, y) = 1 \Longleftrightarrow \exists f \in PDUR^r &: \text{INSIDE}((x, y), f) \\ = 1, \quad &\text{else } UG(x, y) = 0. \end{aligned} \tag{5}$$

The function *INSIDE* tests, if a coordinate lies within a convex polygon.

5.2. Initial path planning

An initial path is determined with help of a wave propagation algorithm. In order to simplify calculations, the geometry of the platform is ignored and the robot is viewed as one single movable point. Therefore, the complete reduced environment is distended with the amount of the platform diameter, plus some extra space for safety reasons, so that the results in both kinds of views coincide. On the beginning of the quest for a feasible path, a simulated wave is sent from the goal point, spreading in all directions. Only free space is treated as propagation medium, so that waves are not allowed to travel though obstacles. As the wave reaches the robots position, the propagation finally stops and the existence of a path is proved. Throughout the propagation, collision free distance to the goal point is stored. This distance information is considered as potential and an artificial potential field is generated. By means of a gradient descent, a path from the robot position to the goal point which bypasses obstacles is determined (see Figure 4).

5.3. Dynamic path adaption

The generated path is permanently modified and adapted due to moving obstacles. For this task, methods of the elastic band developed by O. Khatib and S. Quinlan [24] are used. This approach supposes the path can behave in changing environments like an elastic band. This means that the path is repelled from approaching obstacles but contracts to remove slack as an obstacle withdraws from the current path. The elastic band is represented by bubbles of free space. The radius of a bubble is equivalent to the distance to the closest object. This means that the platform always moves collision free as long as it remains within a bubble. The path is covered with subsequently overlapping bubbles with the center on the path. Thus the collision free path is simply represented by the corresponding set of bubbles with center and radius information (see Figure 5). The path is modified by the adaption of every bubble to the current environment configuration. Internal and external forces apply to the center of each bubble. The internal force removes originated slack and tightens the band. The amount and direction of the internal force is only depending on the position of the two neighboring bubbles, which try to locate the enclosed bubble centric. The external force repels the bubble from obstacles. The distance to the obstacle determines the value of the external force so that close objects result in a strong force. Both forces are weighed and summarized. Each bubble obtains a new location and thus the original path is modified. Finally, the new set of bubbles is checked concerning the complete coverage of the

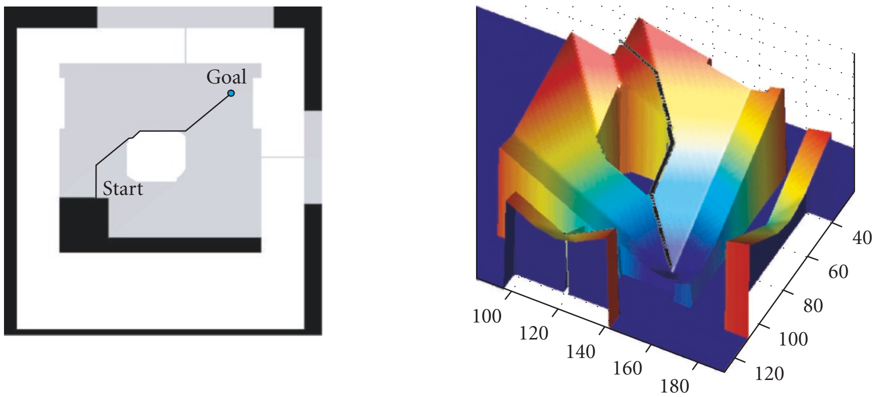


Figure 4: Environment and start/goal configuration, generated artificial potential field, and resulting path.

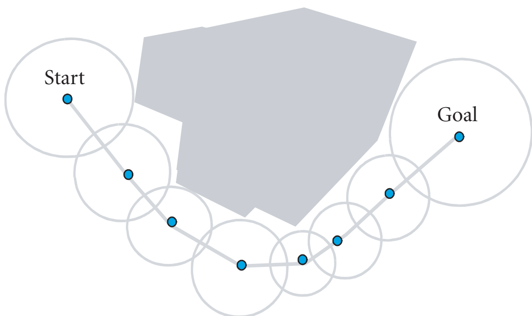


Figure 5: Overlapping bubbles represent the path.

Figure 6: Mobile service robot MORTIMER.

path. If two bubbles do not overlap, an intermediate bubble is generated. Bubbles which are redundant are removed from the bubble set. The resulting bubble set describes the adapted path and will be modified itself in the next time step.

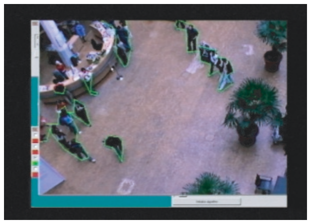
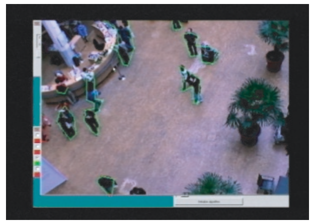
5.4. Alternative paths

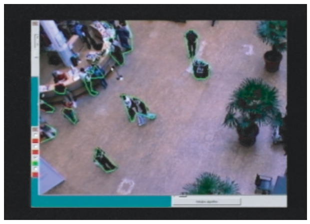
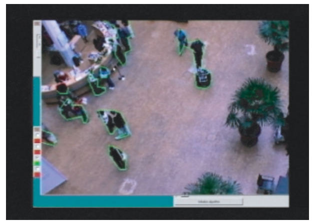
Situations may occur in which the current bubble set does not describe the shortest path to a goal point. Imagine a person walking into a tightened band amid the start and goal points. The elastic band is repelled as long as the person keeps on walking. At some point a path behind the person would be much more reasonable but the bubbles still repel from the moving obstacle and do not describe the shortest path to the goal any more. To handle such cases, the initial path planning module calculates in a parallel thread about every two seconds a path from the current robot position to the goal. If this path is significantly more efficient than the current path, this alternative path is used by the dynamic path adaption module for further path modification.

6. EXPERIMENTS

Some data fusion and path planning experiments have been performed in the entrance hall of the museum of art and media technology (ZKM) in Karlsruhe. In our experimental implementation, there were four LAPUs equipped with sensors and one combined LAPU/RCU for robot control, fusion, path planning, and display. The pure LAPUs are implemented as Matrox 4-Sight-II embedded PCs with 600 MHz Pentium III processors, firewire interface, and 100 Mbit network cards for communication. As sensors, Sony firewire color cameras were used. The four-sensor systems observed an area of approximately 120 qm. The combined LAPU/RCU consisted of a dual processor PC with two Intel Pentium III (933 MHz) processors, 100 Mbit network connection, and additionally fast access to the wireless ethernet for robot control. All components of the advisory system run Windows NT as operating system. As an experimental robot system the service robot Mortimer (see Figure 6) has been used. Basic skills of Mortimer were odometric drive control, collision avoidance, and laser-based repositioning. As the robots tasks are real-time critical, we are using VxWorks as an operating system to ensure the fulfillment of the real-time conditions. The maximal velocity is about 1 m/s.

Figure 7 shows an image sequence of the entrance hall. The detected object contours can be seen at about 10 frames per second. In Figure 8, the integration of the computed view polyhedrons of four LAPUs is demonstrated.


(a) $t = 0$ s

(b) $t = 2$ s

(c) $t = 4$ s

(d) $t = 6$ s

(e) $t = 8$ s

(f) $t = 10$ s

(g) $t = 12$ s

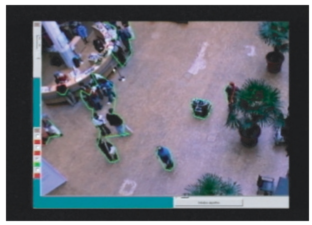
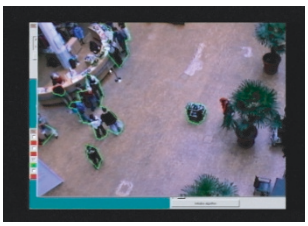
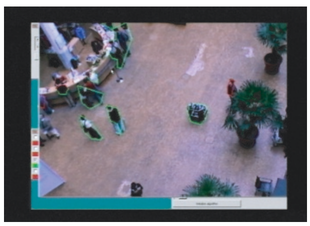
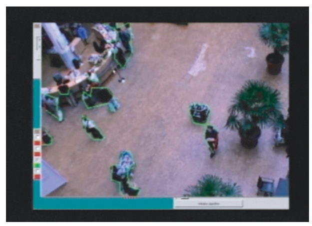
(h) $t = 14$ s

(i) $t = 16$ s

(j) $t = 18$ s

(k) $t = 20$ s

(l) $t = 22$ s

Figure 7: Object detection and contour polygons.

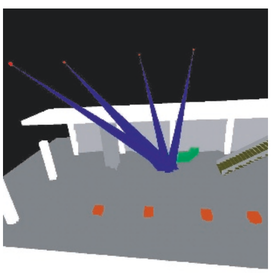
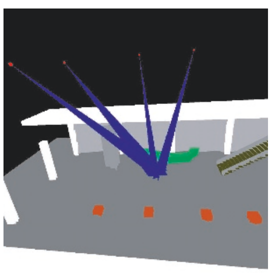
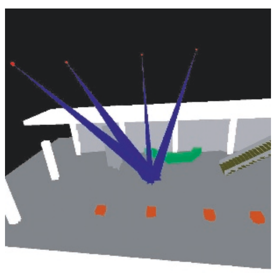
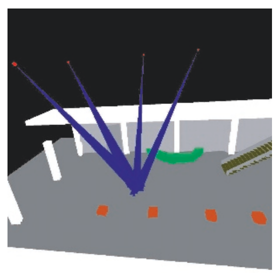
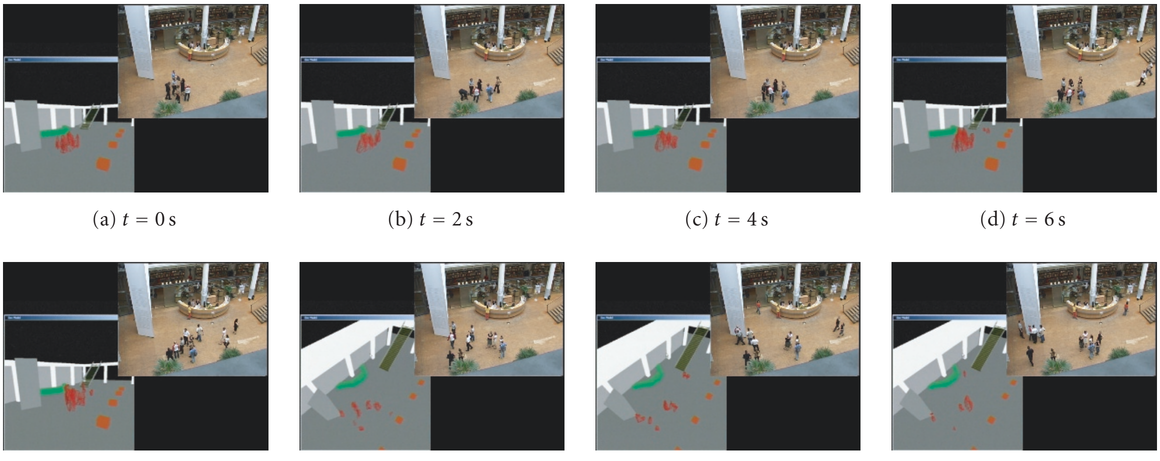

(a) $t = 0$ s

(b) $t = 1$ s

(c) $t = 2$ s

(d) $t = 3$ s

Figure 8: Object polyeder integration.

Figure 9 shows a sequence of generated 3D models and the corresponding situations in the entrance hall. This sequence is also reconstructed by four LAPUs with about 5 frames per second.

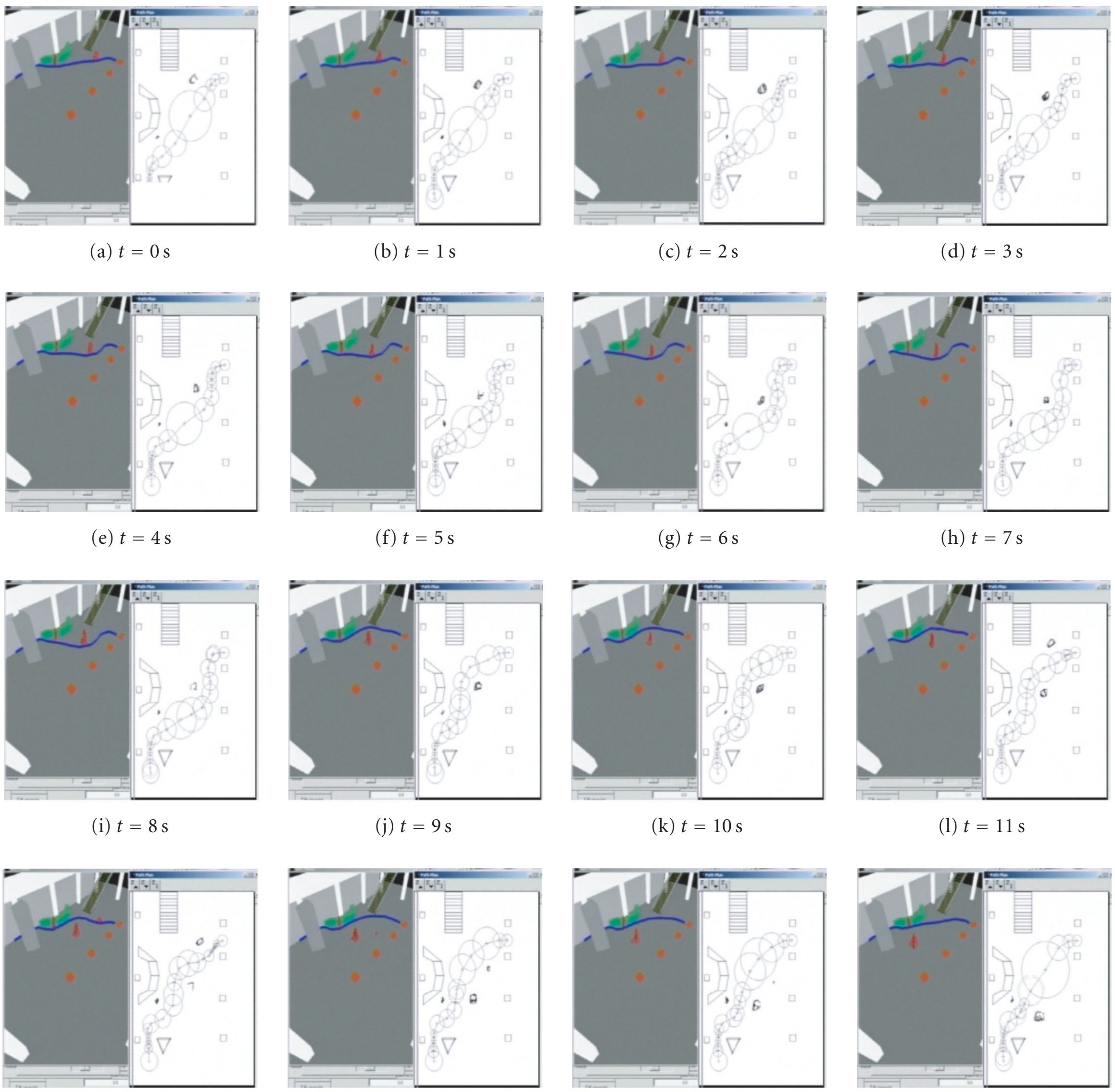
Figure 10 refers to a path planning experiment where a path adaption process was performed between two fixed positions on the floor plane. The initial path planning method gives a first solution to the problem and further on this



(a) $t = 0$ s (b) $t = 2$ s (c) $t = 4$ s (d) $t = 6$ s

(e) $t = 8$ s (f) $t = 10$ s (g) $t = 12$ s (h) $t = 14$ s

Figure 9: Real scenes and corresponding 3D models.



(a) $t = 0$ s (b) $t = 1$ s (c) $t = 2$ s (d) $t = 3$ s

(e) $t = 4$ s (f) $t = 5$ s (g) $t = 6$ s (h) $t = 7$ s

(i) $t = 8$ s (j) $t = 9$ s (k) $t = 10$ s (l) $t = 11$ s

(m) $t = 12$ s (n) $t = 13$ s (o) $t = 14$ s (p) $t = 15$ s

Figure 10: 3D model and corresponding path planning experiment.

solution is adapted by the elastic-band method. System performance depends mainly on the number of bubbles and hence on the number and distance of surrounding static and dynamic obstacles. In the shown experiments, a path update rate of about 8 Hz could be achieved.

7. CONCLUSION

In this article, we have given an overview about our navigation system approach. It consists of a distributed sensor network in combination with distributed data fusion, environment modeling, and path planning. Some first results of mobile and stationary environment views fusion have been shown. A first navigation experiment under real conditions (museum hall) has been demonstrated. Further work will be done in the fields of improving the image processing algorithms, doing environment model analysis, predicting obstacle behavior, and adapting path planning to the results of the prediction process.

ACKNOWLEDGMENTS

This work has been partly supported by the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Center 588 on Humanoid Robots and by the BMBF project MORPHA (anthropomorphic assistance systems). The authors would like to thank both institutions for their support.

REFERENCES

[1] L. Kiss, A. R. Várkonyi-Kóczy, and P. Baranyi, "Autonomous navigation in a known dynamic environment," in *Proceedings of the 12th IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 266–271, St. Louis, Mo, USA, May 2003.

[2] L. Kiss and A. R. Várkonyi-Kóczy, "A hybrid autonomous robot navigation method based on artificial intelligence and soft computing techniques," in *Proceedings of the IFAC International Conference on Intelligent Control Systems and Signal Processing (ICONS '03)*, pp. 251–256, Faro, Portugal, April 2003.

[3] H. Hu and J. M. Brady, "A bayesian approach to real-time obstacle avoidance for an intelligent mobile robot," *Autonomous Robots*, vol. 1, no. 1, pp. 69–92, 1994.

[4] H. Hu, J. M. Brady, F. Du, and P. Probert, "Distributed real-time control of a mobile robot," *The International Journal of Intelligent Automation and Soft Computing*, vol. 1, no. 1, pp. 68–83, 1995.

[5] H. Hu, D. Gu, and M. Brady, "Navigation and guidance of an intelligent mobile robot," in *Proceedings of the 2nd Euromicro Workshop on Advanced Mobile Robots (EUROBOT '97)*, pp. 104–111, Brescia, Italy, October 1997.

[6] B.-S. Ryu and H. S. Yang, "Integration of reactive behaviors and enhanced topological map for robust mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 29, no. 5, pp. 474–485, 1999.

[7] M. Saptharishi, C. S. Oliver, C. P. Diehl, et al., "Distributed surveillance and reconnaissance using multiple autonomous ATVs: CyberScout," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 826–836, 2002.

[8] A. Soto, M. Saptharishi, J. Dolan, A. Trebi-Ollennu, and P. Khoshla, "Cyberatvs: dynamic and distributed reconnaissance and surveillance using all-terrain UGVs," in *Proceedings of International Conference on Field and Service Robotics*, pp. 329–334, Pittsburgh, Pa, USA, August 1999.

[9] C. P. Diehl, M. Saptharishi, J. B. Hampshire II, and P. K. Khosla, "Collaborative surveillance using both fixed and mobile unattended ground sensor platforms," in *Unattended Ground Sensor Technologies and Applications*, vol. 3713 of *Proceedings of SPIE*, pp. 178–185, Orlando, Fla, USA, April 1999.

[10] A. Hoover and B. D. Olsen, "Real-time occupancy map from multiple video streams," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '99)*, vol. 3, pp. 2261–2266, Detroit, Mich, USA, May 1999.

[11] J.-H. Lee, N. Ando, T. Yakushi, K. Nakajima, T. Kagoshima, and H. Hashimoto, "Applying intelligent space to warehouse—the first step of intelligent space project," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '01)*, vol. 1, pp. 290–295, Como, Italy, July 2001.

[12] J.-H. Lee and H. Hashimoto, "Controlling mobile robots in distributed intelligent sensor network," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 890–902, 2003.

[13] K. Morioka, J.-H. Lee, and H. Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 1, pp. 229–237, 2004.

[14] T. Weigel, "Roboter-fußball: selbstlokalisierung, weltmodellierung, pfadplanung und verhaltensbasierte kontrolle," M.S. thesis, Institut für Informatik, Albert-Ludwigs-Universitt Freiburg, Freiburg, Germany, 1999.

[15] T. Weigel, J.-S. Gutmann, M. Dietl, A. Kleiner, and B. Nebel, "CS Freiburg: coordinating robots for successful soccer playing," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 685–699, 2002.

[16] J.-S. Gutmann, W. Hatzack, I. Herrmann, et al., "The CS Freiburg robotic soccer team: reliable self-localization, multirobot sensor integration and basic soccer skills," in *RoboCup-98: Root Soccer World Cup II*, M. Asada and H. Kitano, Eds., Lecture Notes in Artificial Intelligence, pp. 93–108, Springer, Berlin, Germany, 1999.

[17] B. Nebel, J.-S. Guttmann, and W. Hatzack, "The CS Freiburg '99 team," in *RoboCup-99: Robot Soccer World Cup III*, Lecture Notes in Artificial Intelligence, pp. 703–706, Springer, Berlin, Germany, 2000.

[18] T. Fujii, H. Asama, T. Fujita, et al., "Knowledge sharing among multiple autonomous mobile robots through indirect communication using intelligent data carriers," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)*, vol. 3, pp. 1466–1471, Osaka, Japan, November 1996.

[19] T. Fujii, H. Asama, T. Fujita, H. Kaetsu, A. Matsumoto, and I. Endo, "Intelligent data carrier system for cooperative behaviors emerged among collective robots," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 299–304, Orlando, Fla, USA, October 1997.

[20] Y. Arai, T. Fujii, H. Asama, H. Kaetsu, and I. Endo, "Realization of autonomous navigation in multirobot environment," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98)*, vol. 3, pp. 1999–2004, Victoria, Canada, October 1998.

[21] D. Kurabayashi and H. Asama, "Knowledge sharing and cooperation of autonomous robots by intelligent data carrier system," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 1, pp. 464–469, San Francisco, Calif, USA, April 2000.

[22] P. Steinhaus, M. Walther, B. Giesler, and R. Dillmann, "3D global and mobile sensor data fusion for mobile platform navigation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 4, pp. 3325–3330, New Orleans, La, USA, May 2004.

[23] P. Steinhaus, M. Ehrenmann, and R. Dillmann, "Eine skalierbare, verteilte Architektur zur Navigation mobiler Systeme in dynamischen Umgebungen," in *Autonome Mobile Systeme*, pp. 11–18, Karlsruhe, Germany, November-December 1998.

[24] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 802–807, GA, USA, May 1993.

Peter Steinhaus received his Dipl.-Inform. degree in computer science from the University of Karlsruhe (TH), Germany, in 1997. He is currently a Research Assistant at the Institute for Computer Science and Engineering at University of Karlsruhe (TH), Germany, and he coordinates the German Collaborative Research Center on Humanoid Robots.

Marcus Strand received his Dipl.-Ing. degree in electrical engineering from the University of Karlsruhe (TH), Germany, in 2002. He is currently a Research Assistant at the Institute for Computer Science and Engineering at University of Karlsruhe (TH), Germany, and he is currently working in the field of autonomous 3D mapping and sensor fusion.

Rüdiger Dillmann received his Dipl.-Ing. degree in electrical engineering and his Ph.D. degree from the University of Karlsruhe (TH), Germany, in 1976 and 1980, respectively. He is currently a Full Professor in the Department of Computer Science at the University of Karlsruhe (TH), Germany. His research interests include humanoid robotics, programming by demonstration and medical simulation systems. He is the head of the German Collaborative Research Center on Humanoid Robots in Karlsruhe.