Müller, R. Röfer, T., Lankenau, A., Musto, A., Stein, K., Eisenkolb, A. (2000). Coarse Qualitative Descriptions in Robot Navigation. In: Freksa, Ch., Brauer, W., Habel, Ch., Wender, K. F. (Eds.): Spatial Cognition II. Lecture Notes in Artificial Intelligence 1849. Springer. 265-276.

Coarse Qualitative Descriptions in Robot Navigation

Rolf Müller[1], Thomas Röfer[1], Axel Lankenau[1], Alexandra Musto[2], Klaus Stein[2], and Andreas Eisenkolb[3]

[1] Universität Bremen, Postfach 330440, 28334 Bremen, Germany
{rmueller,roefer,alone}@tzi.de
[2] Technische Universität München, 80290 München, Germany
{musto,steink}@in.tum.de
[3] Ludwig-Maximilians-Universität München, 80336 München, Germany
amadeus@imp.med.uni-muenchen.de

Abstract. This work is about the integration of the skills *robot control*, *landmark recognition*, and *qualitative reasoning* in a single autonomous mobile system. It deals with the transfer of coarse qualitative route descriptions usually given by humans into the domain of mobile robot navigation. An approach is proposed that enables the Bremen Autonomous Wheelchair to follow a route in a building, based on a description such as "Follow this corridor, take the second corridor branching off on the right-hand side and stop at its end." The landmark recognition uses a new method taken from the field of image processing for detecting significant places along a route.

1 Introduction

When designing human-computer interfaces for computer systems that solve configuration tasks (e. g. layout-manager), move in space on their own (e. g. semi-autonomous robots), or help humans to move in space (e. g. navigation systems), a good understanding of humans' spatial mental models is important.

It is well known [15] that in these spatial mental models the relations between the elements are coarse and include no metrical information. Typical spatial expressions are "next to", "left of", "east of", etc. Humans' spatial models typically do not only include no metric information, there also occur systematic distortions that influence judgements on distances and directions, e. g. the distance from a landmark to an ordinary building is judged smaller than the other way round, which leads to asymmetric distances [12].

Considering these findings, we see that computer systems dealing with motion in space should be able to understand coarse, *qualitative* relations and should be robust against errors humans make due to systematic distortions, e. g., they should not rely too much on metric information.

In the work described in this paper, qualitative route descriptions are used in an application from the robotics domain, i. e. controlling a semi-autonomous wheelchair along a route. In the field of spatial reasoning, qualitative relations are established between complex objects such as a *refrigerator* or a *person*, e. g. in [2]. In contrast, the items in the qualitative descriptions presented here are very simple, because the autonomous mobile system must be able to recognize them with its limited sensory equipment.

The intended scenario is as follows: In a hospital, a patient should visit a certain room, e. g. for having a medical examination. He or she is handicapped, so a wheelchair is used to travel to the examination room. Normally, the patient would be guided by a nurse. But, considering all the medical examination performed each day in a hospital, this costs a lot of the staff's time. Therefore, the hospital is equipped with intelligent power wheelchairs, enabling the nurse to instruct the wheelchair where to go. Then, the patient is automatically transported to the examination room. Currently, the experiments are carried out in an office building in the University of Bremen. Even though this building is accessible for wheelchairs, navigating there is more complex than in the hospital environment because the corridors and the doors are comparatively narrow.

The *Bremen Autonomous Wheelchair "Rolland"* serves as the experimental platform. It is based on the commercial power wheelchair *Genius 1.522* manufactured by the German company Meyra. The wheelchair is a non-holonomic vehicle that is driven by its front axle and steered by its rear axle. The human operator controls the system with a joystick. In addition, an external keyboard can be used by the service staff, e. g. to type in some instructions. The wheelchair is equipped with a standard PC (Pentium 233, 64 MB RAM) and a ring of sonar sensors to perceive its environment. Furthermore, the system is able to perform dead reckoning by measuring its speed and steering angle.

2 System Architecture

To navigate through an environment along a specified route requires a variety of skills: measuring locomotion (dead reckoning), perceiving the surroundings (obstacle and landmark detection), self-localization (mapping from reality to route description), planning (choosing an appropriate action in each situation), and moving as such.

In a technical system, the navigation skills can be implemented as asynchronously communicating hardware and software components that run in parallel. In the Bremen Autonomous Wheelchair, the communication is done via a real-time capable network (for more information cf. [5]). Since the implementation details do not matter here, Fig. 1 shows a schematic overview of the architecture and the information flow in the system described in this paper.

The wheelchair runs in a control loop of 32 ms cycles and provides three components relevant here: a *state monitor* that supplies information about the current state of the actuators, 27 *sonar sensors* that measure the distance to objects in the surroundings, and the *motor* which accepts driving commands.

In order to hide the specific properties of the hardware, an abstraction of a safe wheelchair had been introduced (cf. [5, 9]). It is called *SAM* (short for Sensor/Actuator-Module). SAM runs in real-time, i. e. its main loop must not

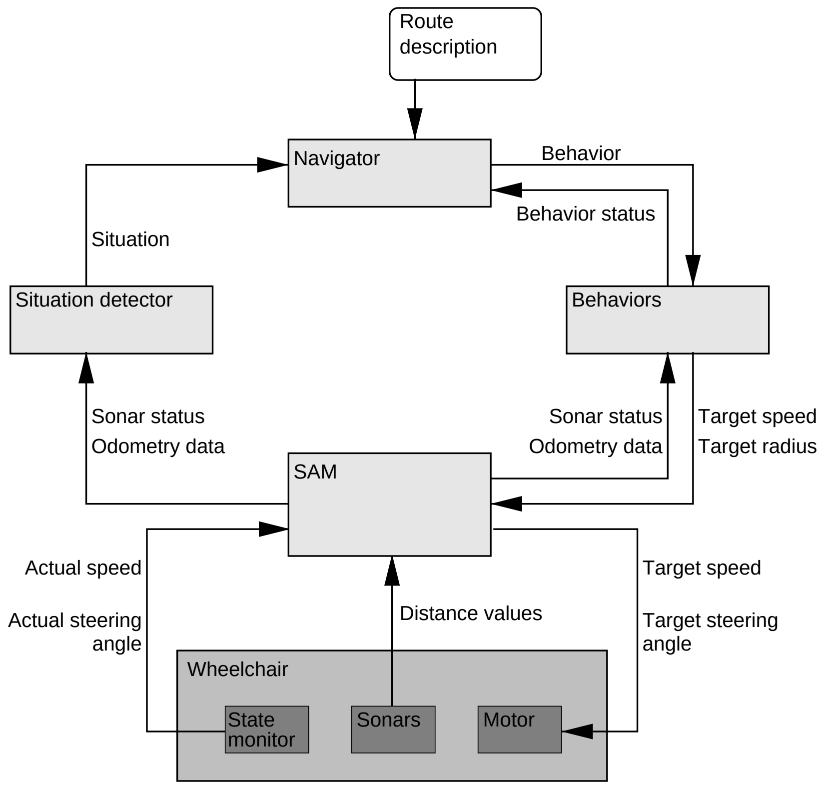

Fig. 1. System architecture (schematic overview; details in the text).

take longer than the 32ms frame time prescribed by the wheelchair. It processes the data produced by the wheelchair and the sensors, waits for driving commands in a $(speed, radius)$-format from other modules, and sends safe target motor commands to the vehicle. Furthermore, SAM provides the higher-level modules *Situation detector* and *Behaviors* with odometry data it derived from the actual speed and the actual steering angle by dead reckoning. In addition, it delivers the current status of the sonar sensors, i. e. the distance value measured by each sensor, the global position where the wheelchair was located when taking that specific measurement, and the absolute point in time of the measurement.

Concurrent to SAM, three high-level modules are responsible for the skills of environment perception and self-localization (*Situation detector*, cf. section 3), choosing adequate motion behaviors (*Navigator*, cf. section 4) and executing these behaviors (*Behaviors*, cf. [4]).

The situation detector extracts features from the sonar image that are mapped to certain landmark types, such as “CorridorRight”. In the current implementation, this module produces a data record every 10 cm travel distance.

It contains the actual position of the wheelchair relative to its starting-point, and a 5-bit feature vector that indicates the detected landmark type.

This information is matched by the navigator with the initial route description which specifies the target track. According to the current situation, the navigator processes a behavior the execution of which allows the wheelchair to follow the target route.

This behavior (e. g. "FollowRightWall") is passed to the third high-level module. It simply converts the active behavior in a target motor command that consists of a speed and a radius component. This command is sent to SAM.

In the sequel, the situation detector and the navigator are described in detail.

3 Perceiving the Environment: Situation Detector

With its limited sensory equipment, the wheelchair can only perceive a small part of its environment. The data is analyzed to detect landmarks, which are used as points of reference during navigation.

In order to recognize them, the situation detector module (cf. section 2) processes the readings of the wheelchair's sonar sensors in several steps. At first, corridor walls are determined with a line detection algorithm. Then a coarse grid of relative positions in the wheelchair's immediate surroundings is searched for prominent properties, such as "there is a wall aligned with the corridor direction on the right side in front". Finally, typical patterns in the coarse grid are recognized yielding landmarks, e. g., "the corridor turns left".

3.1 Probabilistic Obstacle Map

The sonar sensors' measurements are accumulated in a probabilistic obstacle map. This is a local grid map covering an area of $4\,\mathrm{m} \times 4\,\mathrm{m}$. The center of the wheelchair's front axle is located in the center of the map. For each cell in this map, two values are kept track of: the number of measurements that covered this cell and the number of measurements that detected an obstacle in this cell. The ratio of the latter to the former is taken as the cell's probability to contain an obstacle (cf. Fig. 2b).

The map is shifted in the opposite direction of the wheelchair's movement in space, thus it moves in the same way as the real environment observed from the wheelchair.

3.2 Line Segment Detection

In the next step, walls are to be extracted from the probabilistic obstacle map. This is achieved by a method from image processing: edge detection using a Hough transform.

Firstly, the structure matrix [1] is determined. It contains information on the orientation and the contrast of an edge in the local environment of each map cell. This information can be regarded as an orientation vector which serves

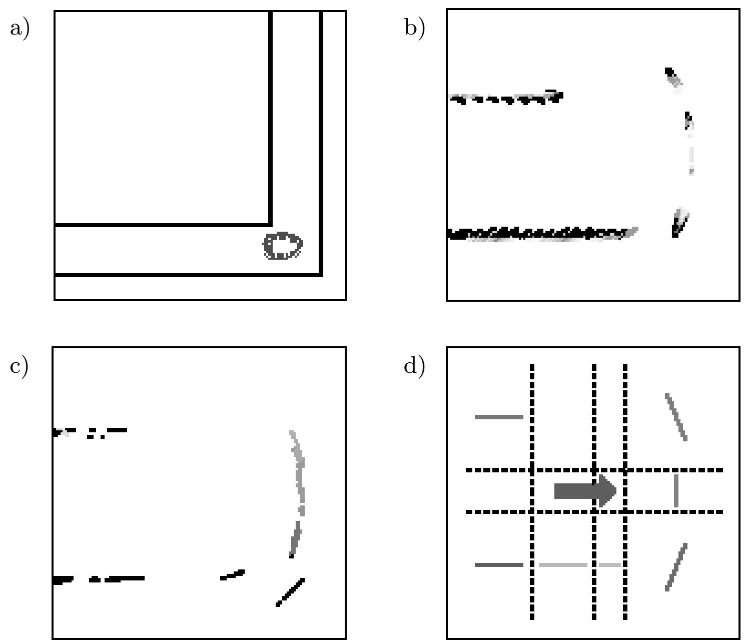

Fig. 2. Processing steps. a) In a simulation, the wheelchair is located in a corner of a corridor facing a wall. b) The probabilistic obstacle map. Darker entries indicate greater occupation probabilities of the map cells. c) The orientation map. The gray values encode the angles of the lines entered. Darker pixels mark clockwise increasing angles. The lightest shade of gray stands for a horizontal line. d) The orientation grid. The arrow indicates the corridor direction. The orientation of the line in each grid cell represents the main orientation angle in the corresponding area of the orientation map. The gray value of a line indicates the frequency of pixels of that orientation. A darker line means more pixels.

for speeding up the Hough transform. The next subgoal is to find maxima of intensity in the Hough space. These correspond to straight lines in the obstacle map which in turn are likely to represent walls in the wheelchair's environment. To find out the exact coordinates of the wall candidates, both dimensions of the Hough space which represent the angle and the distance from the origin of the coordinate system are searched consecutively for relative extrema. A dynamic threshold value is used to reduce the influence of noise.

The result of this procedure is a set of coordinate pairs corresponding to lines of infinite length in the coordinate system of the probabilistic obstacle map. Each of these lines is traced through the obstacle map yielding line segments where the line covers entries in the map.

These solid line segments are inserted into another grid map. In this map, the line segments consist of pixels which encode the orientation angle of the line. The size and resolution of this orientation map are the same as those of the probabilistic obstacle map, and it is shifted according to the wheelchair's movements as well (cf. Fig. 2c).

3.3 Orientation Grid

As mentioned, the wheelchair currently operates in an office environment and will be used in a hospital in the future. Thus, the chief navigation task is to find routes along corridors. To accomplish this, the orientation of a corridor has to be determined first.

The walls of the corridor are expected to be the most frequent entries in the local obstacle map. Therefore, the orientation map is searched for the most prominent orientation angle. This is done by computing a histogram of the angles of the cells in the orientation map. The mean angle of the most frequented class yields the orientation of the corridor.

The orientation angle of the corridor is defined in an interval of length π, since an orientation, e. g., from north to south is equivalent to an orientation from south to north. To obtain a direction, the orientation angle is combined with the heading direction of the wheelchair. There are two directions which comply with the corridor orientation. The direction which differs less from the heading direction of the wheelchair is the direction of the corridor.

In the next step, the orientation map is divided into twelve areas relative to the position of the wheelchair, facing in the corridor direction. These categories of relative positions mark areas of interest for assessing the features of the wheelchair's surrounding. They make up a coarse grid in the orientation map (cf. Fig. 2d). In each of these areas, the main orientation is computed with a histogram in the same way as described above.

3.4 Landmark Detection

The orientation grid is searched for typical patterns that indicate prominent features of the wheelchair's surroundings. These landmark categories are: wall in front, corridor left, corridor right, door left and door right. A Boolean variable corresponds to each of these landmark categories, the state of which is determined according to the presence of the feature. For instance, a "wall in front" is detected if the main orientation angle of the two center grid cells in front of the wheelchair is perpendicular to the corridor direction, and if the sum of the numbers of the wall pixels of that orientation in these grid cells is greater than a threshold value.

Finally, the Boolean vector, the components of which represent the results of the detection of the five categories, is mapped to a specific type of landmark. In the example given in Fig. 2, the landmark category detection yields a "true" value for the categories "wall in front" and "corridor left". The combination of these is the landmark "LeftHandBend."

4 The Navigator

The navigator matches the landmark information computed by the situation detector with the initial route description which specifies the target track. Depending on the current situation, the navigator determines a behavior allowing the wheelchair to follow the route.

4.1 Coarse Route Descriptions

According to [13], humans are used to give route descriptions that can be segmented into pieces mainly belonging to four categories: starting-point, reorientation (direction), path/progression, and goal. However, it was also found that people often give additional information such as extra landmarks (not only at turning points), cardinal directions, and the shape of the path between landmarks. "This information, while not essential, may be important for keeping the traveler confidently on track" [13], p. 169.

Humans typically give route descriptions as a sequence of elementary pieces which consist at least of some of the following items:

- starting-point
- reorientation
- path/progression (additional landmarks, approximate distances, ...)
- goal

In order to ease the communication of a human operator with the wheelchair, it should understand route descriptions that consist only of these elements, and thus enable the vehicle to find its way in a building based on such a description.

The starting-point is always the current position of the robot. The goal is the end of the route. A route description is specified by a sequence of tuples of the following kind:

$$< [\{ \textit{controlmarks} \} \; \textit{router} \;] \; \textit{reorientation} >$$

A *reorientation* is some directional instruction that humans often use in route descriptions [14], such as "TurnLeft", "EnterRightDoor", or "FollowCorridor". A *router* is a landmark where a directional change can take place. The last router is the goal of the route. *Controlmarks* support following routes over longer distances without directional changes; they are especially useful to describe locations where no turn should take place. Depending on the situation, landmarks found by the situation detector are interpreted as controlmarks and routers, respectively.

The coarse qualitative route description A

$$\begin{array}{l} < \text{RightHandBend TurnRight} > \\ < \text{CorridorRight CorridorLeft TurnLeft} > \\ < \text{CorridorRight DeadEnd Stop} > \end{array}$$

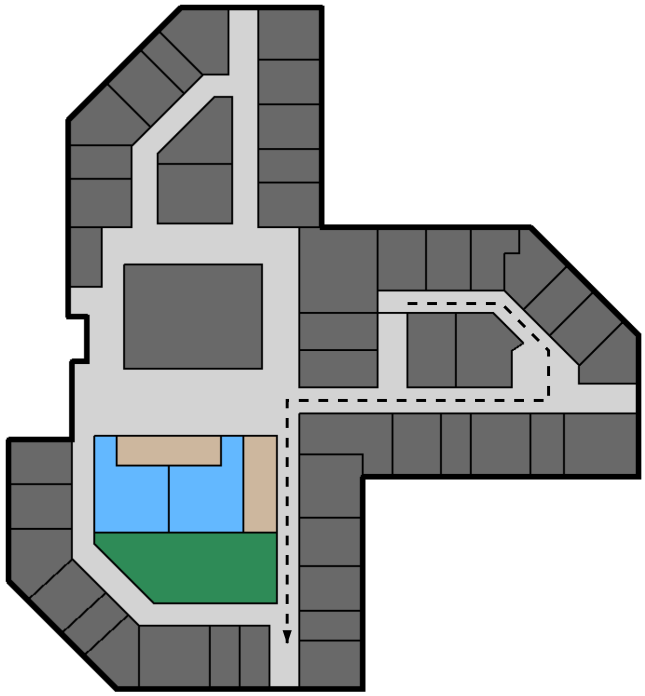

Fig. 3. Plan of the second floor of the MZH-building of Bremen University and the route *A* (dashed line, approx. 65 m in reality)

corresponds to a route depicted as a dashed line in Fig. 3. However, the wheelchair is not able to directly perform operations such as “TurnRight” because it cannot determine how far it has to turn. Instead, basic behaviors such as “FollowRightWall” are employed. They are started before arriving at the router and may end after it has successfully been passed.

In a first step, the route description is converted into a representation that takes such demands into account:

$<$ FollowRightWall RightHandBend$>$
$<$ FollowLeftWall CorridorRight CorridorLeft $>$
$<$ FollowLeftWall CorridorRight DeadEnd $>$
$<$ Stop $>$

This representation does not yet prevent the wheelchair from turning into a corridor that is part of a controlmark and therefore should not be entered. Instead, this is achieved when following the route.

4.2 Generation of Driving Commands

The navigator generates driving commands, i. e. it processes the route representation and derives basic behaviors that are adequate in the specific situations. The basic navigation algorithm works as follows: the navigator always selects the first elementary piece of the route representation, the tuple T. Depending on the contents of T, one of the following three cases can occur:

***T* consists only of a reorientation.** Then, the behavior must have an intrinsic end, and it is able to detect when it is has reached this state, as e. g. “Stop” or “TurnRound”. The navigator waits until the behavior module announces that this action has been terminated by setting the behavior status accordingly. The navigator will then remove T and switch to the next tuple.

There are controlmarks in *T*. These controlmarks have to be straightly passed in order to reach the router where a reorientation has to take place. Therefore, a default behavior is associated with each controlmark type that is necessary to avoid entering, e. g., a branching corridor which is part of the mark. After a controlmark has successfully been passed, it is removed from T and the navigator continues with the rest of the current tuple.

There are no controlmarks or they all have successfully been passed, i. e., they were removed from T. Then, it has to be searched for the router. The major difference between a controlmark and a router is that the wheelchair is not prevented from turning off at the router, so if the router is, e. g., a “Crossing” and the behavior is “FollowLeftWall”, the wheelchair is allowed to—and in fact should—turn into the left branch of the crossing. Again, if the router has successfully been passed, T is deleted from the route representation and the next tuple becomes the actual one.

When the route description is empty, the navigator stops the wheelchair, and it is assumed that the goal has been reached.

5 Experimental Results

Experiments with the wheelchair robot prove that the approach described in this article does work in practice. In Fig. 4 a successful trial to follow route A (for details cf. above) is depicted.

However, relying on sonar sensors for perceiving the environment has its drawbacks. As the angular resolution of these sensors is relatively weak and their distance information is prone to be erroneous (cf. [10]), extracting features of the environment reliably from sonar data was not always successful. For instance, an open door of an office was occasionally interpreted as a branching corridor by the situation detector. Moreover, a closed door could hardly be distinguished

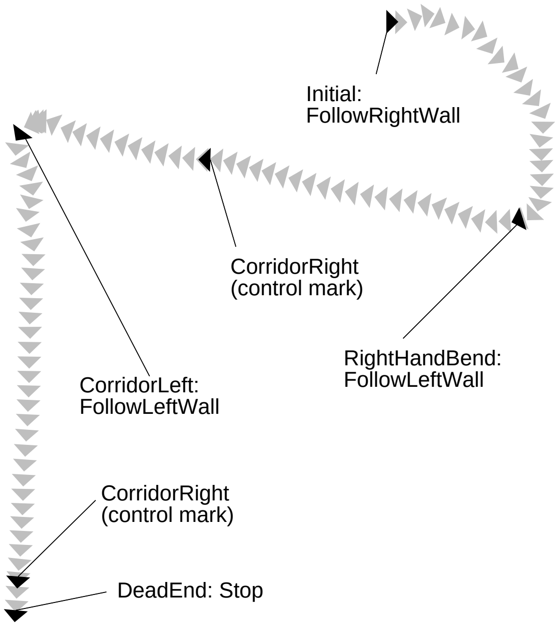

Fig. 4. Path of the wheelchair successfully following route A. The triangles indicate the position and orientation of the robot in intervals of about 0.25 s. The distortions of the depicted path, particularly the turns not appearing as right angles, are due to the non-ideal dead reckoning system of the wheelchair. For a map of the setting of this experiment cf. Fig. 3.

from a wall. Therefore, doors have not been used as controlmarks or routers for these experiments.

6 Conclusion and Outlook

In order to improve the acceptance of service and rehabilitation robots such as the Bremen Autonomous Wheelchair, the problem of human-computer interaction has to be tackled. Especially in the context of elder and handicapped users, these robots have to satisfy some conditions: Firstly, they should be *safe* in the sense that they work as they are intended to do and do no harm to people or objects in their environment. Secondly, they should only *assist* the human operator and provide skills he or she lost due to age, illness or impairments, but they should not replace the user's remaining capabilities. And thirdly, the robots should be easy to control.

This paper deals with the latter in that it presents an approach to feed a mobile robot with a route description that makes use of coarse and qualitative expressions such as "left of". This approach integrates the topics *control of mo-*

bile robots, *landmark recognition* and *qualitative reasoning* in a single service robotics application.

While traveling, the wheelchair perceives its environment by sonar sensors and extracts information about special landmarks, the *controlmarks* and the *routers*. This is done by an algorithm based on line detection which is borrowed from the field of image processing. In accordance with the current situation, the robot chooses one of the available basic behaviors, e. g. "FollowCorridor".

The first experimental results show that the approach presented here is a promising method to guide mobile systems through buildings.

In order to improve the robustness of the algorithm, the recorded odometry data should be generalized (cf. [6, 11] and [8] for incremental generalization of routes) and matched with the route descriptions augmented by qualitative distances.

When using qualitative distances in route descriptions, the performance of the landmark detection can significantly be increased, because the likelihood of the existence of a particular landmark varies with its position along the route. By taking this fact into account, the misinterpretations of sonar images can be minimized.

The integration of the segmentation and classification algorithm presented in [7] will result in a further improvement by extracting motion shapes from the route descriptions. The algorithm exploits the fact that the shape of the path between two landmarks can be considered as a landmark itself, e. g. in a description such as "The street *makes a sharp turn to the left*, which you follow. After this turn, take the next street on the right." The wheelchair can make use of this additional information without further sensor equipment.

Future work will also deal with supplying route descriptions in natural language rather than in the formal language used in this article.

Acknowledgements

The authors are supported by the Deutsche Forschungsgemeinschaft (DFG) within the Priority Program "Spatial Cognition".

References

1. J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *Proc. of the First International Conference on Computer Vision*, pages 433–438. IEEE Computer Society Press, 1987.
2. B. Claus, K. Eyferth, C. Gips, R. Hörnig, U. Schmid, S. Wiebrock, and F. Wysotzki. Reference frames for spatial inference in text understanding. In *Spatial Cognition - An interdisciplinary approach to representing and processing spatial knowledge*, number 1404 in Lecture Notes in Artificial Intelligence. Springer, 1998.
3. C. Freksa and D. M. Mark, editors. *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science. International Conference COSIT'99*, volume 1661 of *Lecture Notes in Computer Science*, Berlin,Heidelberg,New York, August 1999. Springer.

4. B. Krieg-Brückner, Th. Röfer, H.-O. Carmesin, and R. Müller. A taxonomy of spatial knowledge and its application to the Bremen Autonomous Wheelchair. In Ch. Freksa, Ch. Habel, and K. F. Wender, editors, *Spatial Cognition*, volume 1404 of *Lecture Notes in Artificial Intelligence*, pages 373–397, Berlin, Heidelberg, New York, 1998. Springer.
5. A. Lankenau, O. Meyer, and B. Krieg-Brückner. Safety in robotics: The Bremen Autonomous Wheelchair. In *Proceedings of AMC'98, 5th Int. Workshop on Advanced Motion Control*, pages 524–529, Coimbra, Portugal, 1998.
6. A. Musto, K. Stein, A. Eisenkolb, and Th. Röfer. Qualitative and quantitative representations of locomotion and their application in robot navigation. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, August 1999.
7. A. Musto, K. Stein, A. Eisenkolb, K. Schill, and W. Brauer. Generalization, segmentation and classification of qualitative motion data. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 180–184. John Wiley & Sons, 1998.
8. A. Musto, K. Stein, A. Eisenkolb, K. Schill, Th. Röfer, and W. Brauer. From motion observation to qualitative motion representation. In C. Freksa, C. Habel, and K. Wender, editors, *Spatial Cognition II*, Lecture Notes in Artificial Intelligence. Springer, 2000.
9. Th. Röfer and A. Lankenau. Architecture and applications of the Bremen Autonomous Wheelchair. In P. P. Wang, editor, *Proc. of the Fourth Joint Conference on Information Systems*, volume 1, pages 365–368. Association for Intelligent Machinery, 1998.
10. Th. Röfer and A. Lankenau. Ensuring safe obstacle avoidance in a shared-control system. In J. M. Fuertes, editor, *Proc. of the 7th International Conference on Emergent Technologies and Factory Automation*, pages 1405–1414, 1999.
11. Th. Röfer. Route navigation using motion analysis. In Freksa and Mark [3], pages 21–36.
12. E.K. Sadalla, W.J. Burroughs, and L.J Staplin. Reference points in spatial cognition. *Journal of Experimental Psychology: Human Learning and Memory*, 6:516–528, 1980.
13. B. Tversky and P. U. Lee. How space structures language. In C. Freksa, C. Habel, and K.-F. Wender, editors, *Spatial Cognition. An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, volume 1404 of *Lecture Notes in Artificial Intelligence*. Springer, 1998.
14. B. Tversky and P. U. Lee. Pictorial and verbal tools for conveying routes. In Freksa and Mark [3], pages 51–64.
15. B. Tversky. Cognitive maps, cognitive collages, and spatial mental models. In Andrew U. Frank and Irene Campari, editors, *Spatial Information Theory. A Theoretical Basis for GIS. European Conference, COSIT'93*, volume 716 of *Lecture Notes in Computer Science*, pages 14–24, Berlin, Heidelberg, New York, 1993. Springer.