Key Technologies in Robot Assistants: Motion Coordination Between a Human and a Mobile Robot

Erwin Prassler, Dirk Bank, and Boris Kluge

Abstract: In this paper we describe an approach to coordinating the motion of a human with a mobile robot moving in a populated, continuously changing, natural environment. Our test application is a wheelchair accompanying a person through the concourse of a railway station moving side by side with the person. Our approach is based on a method for motion planning amongst moving obstacles known as Velocity Obstacle approach. We extend this method by a method for tracking a virtual target which allows us to vary the robot's heading and velocity with the locomotion of the accompanied person and the state of the surrounding environment.

Keywords: robot motion planning, motion coordination, dynamic environments, human-robot interaction

I. Introduction

The interaction, cooperation, and coordination between a human and a robot system is a research topic which has attracted much attention recently [1][2][5][6]. Under captions such as *human-friendly robotics* or *human-robot co-existence* this field covers a large variety of aspects. These aspects reach from a human-robot communication based on "human-friendly" communication channels such as natural language, gestures, mimics, over an understanding of the context of a task or an understanding of situations where human and robot have to interact to achieve a common task, to physical interaction (robot touches human, human touches robot) and finally to coordinating the motion and action of the human and the robot.

In this paper we study the problem of coordinating the motion of a mobile robot and a human through a populated, continuously changing natural environment. This problem has a very useful application in the transportation of disabled or elderly people or the transportation of patients in a hospital. Transportation services are usually carried out by nursing personnel which pushes the patient or disabled person sitting or lying in some type of carriage, for example a wheelchair or a hospital bed. Since pushing and maneuvering a heavy carriage exposes the back of the pushing person to significant strain, these people often suffer severe long-term back problems. Using a robotic wheelchair, which is able to follow the nurse side by side like a heeling dog, through arbitrarily populated, continuously changing natural environments would certainly allow the reduction of this problem or even avoid it.

Accompanying an object or a person side by side involves the control of the relative position with respect to a target position which has to be inferred from a predicted position of the accompanying person. Besides this, there are further constraints which affect the heeling of a person. Ideally, the robot and the person should move at the same velocity. Only if this constraint is satisfied a true accompanying behavior can be achieved. So, accompanying a person side by side is not only a position control problem but at the same time a velocity control problem. As a final constraint, the target position, inferred from the predicted position of the accompanying person may be perturbated through obstacles in the environment.

A side by side locomotion or heeling behavior differs from following a person as it is described in the literature (e.g. [9][10]) in a very crucial way. A simple example which makes this difference obvious is passing a door by the compound robot human. If the robot simply follows the person there is no problem at all. The problem obviously arises, when the robot is requested to accompany the person side by side. In this situation the robot has to get behind the person while passing through the door and catch up once the door is passed.

One might argue that this situation suggests to design several distinct behaviors, one for *following* and one for *accompanying*, and arbitrate between them appropriately. In the following section, we describe a different approach. This approach allows the implementation of an accompanying behavior for various situations in a uniform manner and does not require to arbitrate between two or more distinct behaviors.

II. Coordinating the motion of a human and a mobile robot

For coordinating the motion between a mobile robot and a human we use a three layer control architecture. At the bottom of this architecture is the layer for elementary motion control. This layer offers two control modes, a mode for velocity control[1] and a mode for position control.

The layer for elementary motion control receives its input from the layer for *tactical navigation*, which computes a collision-free course to a target position in an environment with stationary as well as moving objects. This layer is based on the *Velocity Obstacle* approach [4][8] which is summarized in the following section.

The Velocity Obstacle approach assumes a target velocity and heading, which in the case of an unobstructed path leads to the

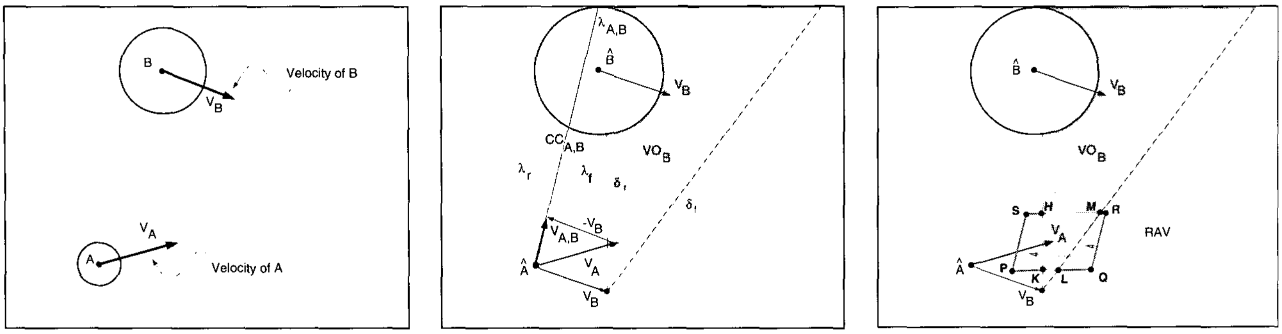
---

Manuscript received: Aug. 23, 2001., Accepted: Nov. 16, 2001.

Erwin Prassler, Dirk Bank, Boris Kluge: Research Institute for Applied Knowledge Processing at the University of Ulm (prassler,bank,kluge @faw.uni-ulm.de)

† This work was supported by the German Department for Education and Research (BMB+F) under grant no. 01 IL 902 F6 as part of the project MORPHA.

[1] More specifically, velocity control applies to both the translational and rotational velocity (v, ω)



a) two objects on collision course b) velocity obstacle *VO* c) reachable avoidance velocities *RAV*

Fig. 1. The *velocity obstacle* approach.

target location. In the case of obstacles the approach computes an actual velocity and heading, which ensure a collision-free locomotion. These actual values may deviate from the target values. They are selected, however, such that they minimize this deviation from the target velocity and heading.

The tactical navigation layer is fed by the top layer which we called the *Virtual Target Tracking*. By defining a relative position of the robot with respect to the person which is to be accompanied and by deriving proper target velocities and headings in order to maintain this relative position while both human and robot are moving, this top layer creates the desired accompanying behavior.

1. Velocity obstacles

To introduce the Velocity Obstacle (VO) concept, we consider the two circular objects, A and B, shown in Fig. 1a at time t_0, with velocities $\mathbf{v}_A$ and $\mathbf{v}_B$. Let circle A represent the mobile robot, and circle B represent an obstacle. To compute the VO, we first map B into the *Configuration Space* of A, by reducing A to the point $\widehat{A}$ and enlarging B by the radius of A to $\widehat{B}$, and represent the state of the moving object by its position and a velocity vector attached to its center. Then, the set of colliding *relative* velocities between $\widehat{A}$ and $\widehat{B}$, called the *Collision Cone*, $CC_{A,B}$, is defined as $CC_{A,B} = \{\mathbf{v}_{A,B} \mid \lambda_{A,B} \cap \widehat{B} \neq \emptyset\}$, where $\mathbf{v}_{A,B}$ is the relative velocity of $\widehat{A}$ with respect to $\widehat{B}$, $\mathbf{v}_{A,B} = \mathbf{v}_A - \mathbf{v}_B$, and $\lambda_{A,B}$ is the line of $\mathbf{v}_{A,B}$. This cone is the light grey sector with apex in $\widehat{A}$, bounded by the two tangents λ_f and λ_r from $\widehat{A}$ to $\widehat{B}$, shown in Fig. 1b. Any relative velocity that lies between the two tangents to $\widehat{B}$, λ_f and λ_r, will cause a collision between A and B. Clearly, any relative velocity outside $CC_{A,B}$ is guaranteed to be collision-free, provided that the obstacle $\widehat{B}$ maintains its current shape and speed.

The collision cone is specific to a particular pair of robot/obstacle. To consider multiple obstacles, it is useful to establish an equivalent condition on the *absolute* velocities of A. This is done simply by adding the velocity of B, $\mathbf{v}_B$, to each velocity in $CC_{A,B}$ and forming the *Velocity Obstacle VO*, $VO = CC_{A,B} \oplus \mathbf{v}_B$ where $\oplus$ is the Minkowski vector sum operator, as shown in Fig. 1b by the dark grey sector. The VO partitions the absolute velocities of A into *avoiding* and *colliding* velocities. Selecting $\mathbf{v}_A$ outside of VO would avoid collision with B. Velocities on the boundaries of VO would result in A grazing B.

To avoid multiple obstacles, we consider the union of the individual velocity obstacles, $VO = \cup_{i=1}^{m} VO_{B_i}$, where m is the number of obstacles. The avoidance velocities, then, consist of those velocities $\mathbf{v}_A$, that are outside all the VO's.

An *avoidance maneuver* consists of a one-step change in velocity to avoid a future collision within a given time horizon. The new velocity must be achievable by the moving robot, thus the set of avoidance velocities is limited to those velocities that are physically reachable by robot A at a given state over a given interval. This set of *reachable velocities* is represented schematically by the polygon $PQRS$ shown in Fig. 1c. The set of *reachable avoidance velocities*, RAV, is defined as the difference between the reachable velocities and the velocity obstacle. A maneuver avoiding obstacle B can then be computed by selecting any velocity in RAV. Fig. 1c shows schematically the set RAV consisting of two disjoint subsets. For multiple obstacles, the RAV may consist of multiple disjoint subsets.

It is possible then to choose the type of an avoidance maneuver, by selecting on which side of the obstacle the mobile robot will pass. As discussed earlier, the boundary of the velocity obstacle VO, $\{\delta_f, \delta_r\}$, represents all absolute velocities generating trajectories tangent to $\widehat{B}$, since their corresponding relative velocities lay on λ_f and λ_r. For example, the only tangent velocities in Fig. 1c are represented by the segments KH and LM of the reachable avoidance velocity set RAV. By choosing velocities in the set $PKHS$ or $MLQR$, we ensure that the corresponding avoidance maneuver will avoid the obstacle from the rear, or the front, respectively.

2. Motion coordination by tracking a virtual moving target

Let us consider two moving objects H and R. To coordinate the motion of both objects such that one object "accompanies" the other requires to control the relative position of the accompanying object R with respect to the accompanied object H during locomotion. As mentioned above, in our application the accompanied object is a human, for example, a nurse, and the accompanying object is a robotic wheelchair (robot in short) transporting a disabled person or a patient. We describe this desired relative position of the accompanying robot with respect to the accompanied human by a vector $p_H - p_R$, where p_H and p_R are the positions of the human and the robot, respectively. Alternatively the desired relative position can be described by a lateral distance d_L and an advance distance d_A.

What does maintaining the relative position between the accompanying and the accompanied object now mean in detail? As the robot's perception takes place only at discrete time steps, let us consider the problem at such a discrete time step t (see Fig. 2). At time t the robot observes the human at a position $p_H(t)$. From its previous observations the robot knows, that the human was at position $p_H(t-1)$ at time $t-1$. This allows the robot to infer the human's velocity $v_H(t)$ at time t with $v_H(t) = (\, p_H(t) - p_H(t-1)\,)/(\Delta t)$. Note that in our definition velocity is a vector which comprises magnitude and direction of the velocity. Informally we speak about the human's velocity and heading.

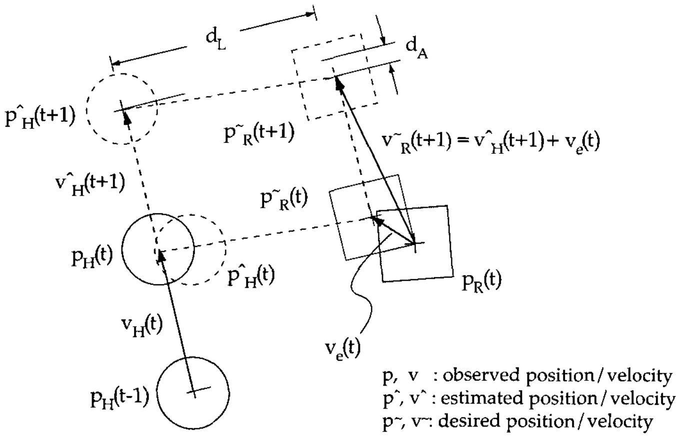
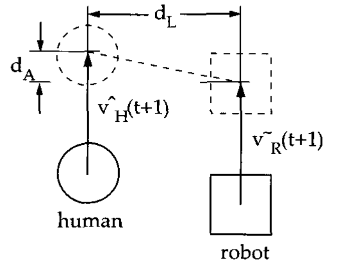


Fig. 2. Accompanying behavior based on tracking a virtual moving target.

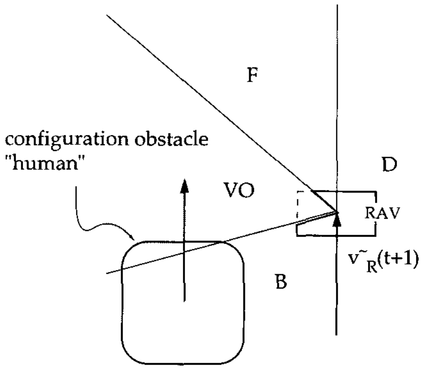
Let us now assume for a moment that the robot at time t has reached the desired relative position with respect to the actual position of the accompanied person. We denote this position as $\tilde{p}_R(t)$. In order to maintain this desired relative position during its locomotion the robot needs to know the human's future velocity and heading, from which it can compute the velocity and heading taking it to this desired relative position in the future. We view this desired future relative position $\tilde{p}_R(t+1)$ as *virtual target* which moves with the accompanied person. Hence the term *virtual moving target*.

The apparent problem is to estimate this future velocity and heading of the accompanied person. Sophisticated methods for estimating the velocity and heading of a moving object are described in [3][7]. However, as the accurate prediction of the motion of a person through a populated, natural environment, which underlies continuous as well as spontaneous changes, is a difficult and often vain venture in practice, we content ourselves with a very coarse estimate. We simply extrapolate the person's past velocity $v_H(t)$ and take it as an estimate of the person's future velocity $\hat{v}_H(t+1)$. With $\hat{v}_H(t+1)$ we also obtain an estimate of the person's position at time $t+1$, $\hat{p}_H(t+1)$. This makes it straightforward to determine the desired position of the robot, $\tilde{p}_R(t+1)$, at time $t+1$ (see Fig. 2).

At the beginning of the above considerations we assumed that at time t the robot has reached the desired relative position $\tilde{p}_R(t)$ with respect to the accompanied person. Apparently this assumption describes an ideal situation. For a number of reasons it is very unlikely, however, that the robot has really reached the desired position. Firstly, the robot's locomotion towards its virtual target may have been perturbated by stationary or moving obstacles in the environment. So the robot may have deviateted from the desired velocity $\tilde{v}_R(t)$. We discuss this issue in the following section. Secondly, the estimate of the person's future velocity and position, which we used to determine the robot's desired velocity, could have be inaccurate. The robot will not notice this before it makes its observation at time t. Finally, the robot's velocity control is afflicted with a certain delay, which also results in a positional error.



a) work space



b) configuration space

Fig. 3. Combining the Velocity Obstacle Approach with virtual target tracking

All these reasons together make it very likely that the robot, when it senses its own position and the position of the accompanying person at time t, will discover that its true position $p_R(t)$ deviates noticeably from the desired position $\tilde{p}_R(t)$, which is determined by the person's true position $p_H(t)$ and the offset by the lateral and advance distance, d_L and d_A, respectively. To compensate for this positional error we have to add a velocity vector $v_e(t)$ to the vector $\hat{v}_H(t+1) = v_H(t)$, which would be the robot's desired velocity in the ideal case. Then the corrected desired velocity for the robot becomes $\tilde{v}_R(t+1) = \hat{v}_H(t+1) + v_e(t)$. Note that we implicitly assume that the intervals between the instants $t-1, t, t+1, \ldots$ are of constant length Δt.

3. Combining the velocity obstacle approach with virtual target tracking

3.1 Accompanying a person in an unobstructed environment

The velocity $\tilde{v}_R(t+1)$ is the input to the control layer which computes the Velocity Obstacles and determines a collision-free course for the robot. This velocity in the ideal case of an unobstructed environment takes the robot to the desired relative position with respect to the accompanying person and thereby causes desired accompanying behavior. In the following we briefly discuss this combination of the Velocity Obstacle approach with the Virtual Target Tracking approach and its effects in an unobstructed environment.

Fig. 3a shows the robot and the human moving in an ideal unobstructed environment. As a result of the prediction of the human's future motion and the computation of the desired future relative position of the robot we obtain $\tilde{v}_R(t+1)$ as described above. At this point we do not care about the conditions of the environment, which might effect this tracking of a virtual target. We do not account for any objects neither stationary nor moving.

For the robot's desired velocity $\tilde{v}_R(t+1)$ we then compute the velocity obstacles for the objects surrounding the robot. In the ideal circumstances which we assumed for the moment, there is just one velocity obstacle, namely that caused by the accompanied person. In Fig. 3b we see, that the robot's desired velocity $\tilde{v}_R(t+1)$ coincides with the apex of the velocity obstacle caused by the accompanied person but does not lie within VO. Accordingly, $\tilde{v}_R(t+1)$ is a *reachable avoidance velocity*, i.e. not a colliding velocity, and the robot can in fact move towards the desired relative position at t. Of course, this is what we expect, but it is still worth mentioning, that in the case of an unobstructed environment the computation of a reachable avoidance velocity does not "overwrite" the desired velocity $\tilde{v}_R(t+1)$. This will be different in the case of an obstructed environment as we will see in the following section.

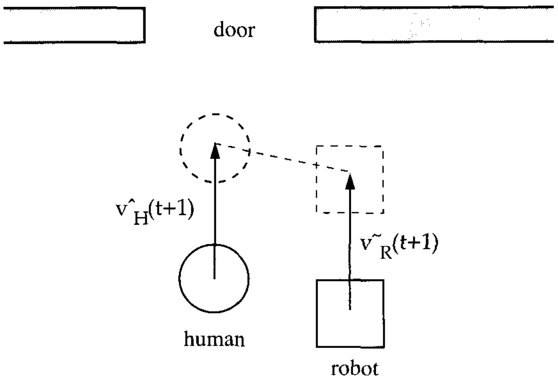
A final remark about the robot's desired $\tilde{v}_R(t+1)$: As $\tilde{v}_R(t+1)$ lies outside of VO, it is therefore a reachable avoidance which does not even require a velocity change. So, it is most natural that the robot maintains this direction and velocity. Now let us ignore for a moment that the robot is supposed to track a virtual target and thus should follow $\tilde{v}_R(t+1)$. According to the Velocity Obstacle approach any velocity in the set of reachable avoidance velocities can be chosen in order to avoid a collision with the person. Of course, the robot's relative position with respect to the accompanied person varies depending on which velocity the robots selects. In Fig. 3b we see three marked regions in addition to the area representing the velocity obstacle VO, labeled with D, F, and B, respectively. If the robot selects a reachable avoidance velocity, which lies in F, the robot will speed up, reduce the distance to the person temporarily and pass to the person's left in front of the person. If it selects a reachable avoidance velocity in B, the robot will fall behind, still approach the person, but pass it in its rear. If the robot selects a velocity in D it will steadily increase the distance to and depart from the person.

3.2 Motion coordination with perturbation - passing through a door

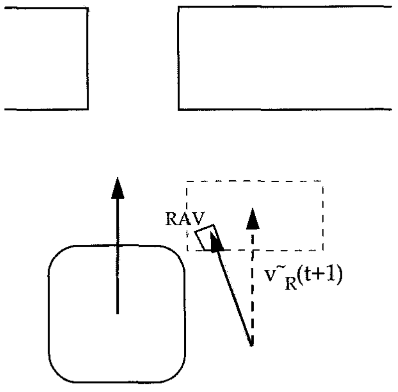
Somewhat more difficult than moving side by side through an unobstructed environment is the situation, when human and robot have to pass through a narrow door or a narrow hallway. Apparently the robot will have to adapt its relative position with respect to the accompanied person depending on and accounting for the environmental conditions while it still tries to accompany the person.

An instance of such a situation is shown in Fig. 4a). Human and robot approach a door, which is too narrow to allow the robot accompanying the human as it would in an unobstructed environment. How does our approach handle this situation?

Like in an unobstructed environment we determine the relative position which the robot should maintain in order to accompany the person and compute the desired velocity $\tilde{v}_R(t+1)$ which is requested for such a behavior. Note, that in the example the virtual target $\tilde{p}_R(t+1)$ may lie behind the wall or even coincide with the wall.



a) work space



b) configuration space

Fig. 4. Human and robot passing through a door.

We then compute the velocity obstacles for all the obstacles in the scene based on the robot's requested velocity $\tilde{v}_R(t+1)$ and determine the set of reachable avoidance velocities (RAV). Both the velocity obstacles and the RAV set are shown in Fig. 4b. We can see in the figure that the requested velocity $\tilde{v}_R(t+1)$ (dashed arrow) in this scenario is not in the set. The Velocity Obstacle approach "overwrites" $\tilde{v}_R(t+1)$ and yields a velocity (bold arrow) which allows to avoid any collisions and minimizes the deviation with respect to $\tilde{v}_R(t+1)$.

This commanded velocity lets the robot slow down, reduce the distance, and fall behind the person. Further iterations of this procedure will slow down the robot even further until it gets entirely behind the accompanied person. The robot will pass the door behind the accompanied person and then speed up again in order to reach the desire relative position again.

This behavior automatically results from the combination of the Virtual Target Tracking approach with the Velocity Obstacle approach. It does not require any additional mechanisms.

III. Experimental results

The above function has been extensively and successfully tested in the concourse of the central station in Ulm, Germany, during regular business hours. The mission was for the robot to accompany a person side by side in a lateral distance of 60 cm through the concourse. The concourse has a size of about 20×40 m^2, with several rows of seats, an information booth and several ticket machines. During the experiments typically between 50 and 100 people were constantly staying and moving in the concourse. The total mission time adds up to 8–10 hours distributed over several days. The distance traveled during that time adds up to around three kilometers. Due to visibility problems such as occlusion, the robot several times lost the person which was to accompany and then stopped. But there was no collision between the robot and a passenger.

In Fig. 5 an experiment is shown, where the formation human and robot, move through a door. The top picture shows the robot which is leaving its ideal accompanying position and getting back behind the person. In the second picture the robot has completely taken a following position and passes the door behind the person as shown in the bottom picture.

IV. Conclusion

In this paper we described an approach for coordinating the motion between a human and a mobile robot in a populated, continuously changing, natural environment. Our test application is a robotic wheelchair accompanying a person through the concourse of a railway station moving side by side with the person. During several experiments the robot successfully managed to accompany a person through a populated concourse over a total distance of around three kilometers with a total mission time of 8-10 hours.

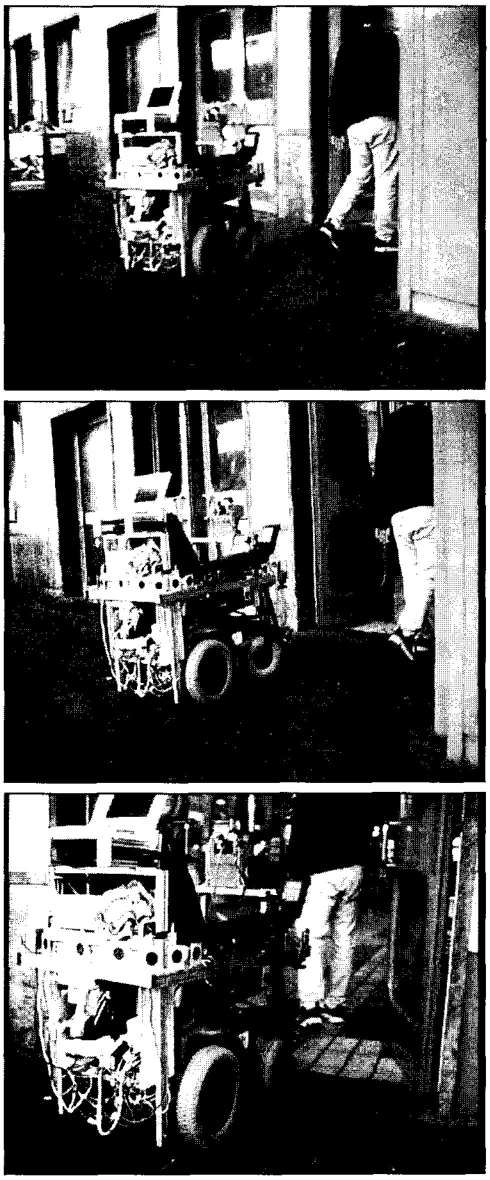


Fig. 5. A robotic wheelchair accompanying a person through a railway station.

References

[1] IEEE workshop on robot and human interaction, since 1992.

[2] First Europe-Japan symposium on human-friendly robotics, Tokyo, Japan, 1999.

[3] Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*. Academic Press, 1988.

[4] P. Fiorini, Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *Int. Journal on Robotics Research*, 17(7), pp. 711–727, 1998.

[5] R. Ikeura, H. Monden, and H. Inooka. Cooperative motion control of a robot and a human. In *Proc. IEEE Int. Workshop on Robot and Human Communication*, pp. 112–117, 1990.

[6] J. Y. S. Luh and S. Hu. Interactions and motions in human-robot coordination. In *Proc. 1999 IEEE Int. Conf. on Robotics and Automation ICRA'99*, Detroit, MI, pp. 3171-3176, 1999.

[7] P. S. Maybeck *Stochastic models, estimation, and control*. Academic Press, 1979.

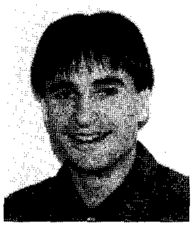
[8] E. Prassler, J. Scholz, and P. Fiorini. Navigating a robotic wheelchair in a railway station during rush hour. *Int. Journal on Robotics Research*, 18(7), pp. 760–772, 1999.

[9] C. Schlegel, J. Illmann, H. Jaberg, and R. Wörz. Vision based person tracking with a mobile robot. In *Proc. British Machine Vision Conf.*, vol. 2, pp. 418–427, 1998.

[10] H. Sidenbladh, D. Kragic, and H. I. Christensen. A person following behavior for a mobile robot. In *Proc. IEEE Int. Conf. on Robotics and Automation ICRA'99*, Detroit, MI, pp. 670–675, 1999.

Erwin Prassler

Dr. Erwin Prassler received a master's degree in Computer Science from the Technical University of Munich in 1985 and a Ph.D. in Computer Science from the University of Ulm in March 1996. Between 1986 and 1989, he held positions as a member of the scientific staff at the Technical University of Munich and as a guest researcher in the Computer Science Department at the University of Toronto. In fall 1989, Dr. Prassler joined the Research Institute for Applied Knowledge Processing (FAW) in Ulm, Germany, as an associate researcher. Since 1994, Dr. Prassler has been heading a research group working in the field of mobile robots and service robots. He has published more than 50 papers and articles in the fields of path planning and motion planning, robot navigation, service robotics, and robotics in computer integrated manufacturing. He is the editor of a German research guide for robotics and automation. Erwin Prassler is a member of the Institute of Electrical and Electronic Engineers (IEEE), the Association for Computing (ACM), the German Association for Computer Science (GI), and the Association of German Engineers (VDI/VDE). He is the chair of the IEEE Technical Committee for Service Robots and a co-chair of the GI Committee for Robot Systems.



Dirk Bank

He received his M. Sc. degree in electronic engineering from the University of Wales, Cardiff, U.K., in 1996. He is employed as a scientist at the Research Institute for Applied Knowledge Processing in Ulm, Germany. His research interests include autonomous mobile systems, sensor data fusion, ultrasonic sensing, automatic error detection and error recovery.

Boris Kluge

He received his M. Sc. degree in computer science from the University of Würzburg, Germany, and is currently a scientist within the autonomous systems division at the Research Institute for Applied Knowledge Processing at the University of Ulm, Germany. Besides robotics and autonomous systems, he has an interest in applied graph theory and computational geometry.