Tele-presence in Populated Exhibitions through Web-operated Mobile Robots

Wolfram Burgard[1] Panos Trahanias[2] Dirk Hähnel[1] Mark Moors[3]
Dirk Schulz[3] Haris Baltzakis[2] Antonis Argyros[2]

[1]University of Freiburg, Computer Science Department, Germany

[2]Foundation for Res. and Technology - Hellas (FORTH) and University of Crete, Greece

[3]University of Bonn, Department of Computer Science, Germany

Abstract

This paper presents techniques that facilitate mobile robots to be deployed as interactive agents in populated environments such as museum exhibitions or trade shows. The mobile robots can be tele-operated over the Internet and, this way, provide remote access to distant users. Throughout this paper we describe several key techniques that have been developed in this context. To support safe and reliable robot navigation, techniques for environment mapping, robot localization, obstacle detection and people-tracking have been developed. To support the interaction of both web and on-site visitors with the robot and its environment, appropriate software and hardware interfaces have been employed. By using advanced navigation capabilities and appropriate authoring tools, the time required for installing a robotic tour-guide in a museum or a trade fair has been drastically reduced. The developed robotic systems have been thoroughly tested and validated in the real-world conditions offered in the premises of various sites. Such demonstrations ascertain the functionality of the employed techniques, establish the reliability of the complete systems, and provide useful evidence regarding the acceptance of tele-operated robotic tour-guides by the broader public.

1 Introduction

Mobile robotic technology and its application in various sectors is currently an area of high interest. Research in this field promises advanced developments and novelties in many aspects. Over the last decade, a variety of service robots were developed that are designed to operate in populated environments. Example cases are robots that are deployed in hospitals [23], museums [6, 27, 38], trade-fairs [31], office buildings [2, 34, 1], and department stores [12]. In these environments the mobile robots perform various services, e.g., deliver, educate, entertain or assist people. Moreover, robots may offer alternative ways for interactive tele-presence in exhibition spaces, a topic that we informally term "robots in exhibitions."

The deployment of robotic systems able to operate in populated environments is heavily based on advances in a number of enabling technologies that facilitate safe, reliable and effective operation of mobile robots and the execution of assigned tasks. The set of enabling technologies needed to pursue a specific application varies according to the tasks implied by the application as well as the environment in which the robot(s) operate. Still, there exists a set of generic technologies that are essential to most of the applications. Examples are technologies that give rise to standard navigation competences, such as mapping, localization, path planning, and obstacle avoidance [9, 24, 19, 37].

In this paper, we describe a number of techniques that cover various aspects of robots that are deployed in populated environments and hence have to interact with people therein. Among them are techniques for mapping large environments, an obstacle-avoidance technique that relies on laser-vision fusion to detect objects that are invisible to the laser scanner, a method for tracking people with a moving mobile robot, and an approach to filter out range measurements coming from moving persons in the process of map construction. We also describe new aspects of the user interfaces, such as a speech interface for on-site users and a flexible web-interface with enhanced visualization capabilities for remote users.

For robots operated over the web, video streaming and enhanced visualizations may be used for improving the user's perception of the environment. Furthermore, robots physically interacting with people may employ enhanced interaction interfaces like speech recognition or text-to-speech synthesis. A variety of Web-based tele-operation interfaces for robots has been developed over the last years. Three of the earlier systems are the Mercury Project, the "Telerobot

on the Web," and the Tele-Garden [16, 17, 36]. These systems allow people to perform simple tasks with a robot arm via the Web. Since the manipulators operate in prepared workspaces without any unforeseen obstacles, all movement commands issued by a Web user can be carried out in a deterministic manner. Additionally, it suffices to provide still images from a camera mounted on the robot arm after a requested movement task has been completed. The mobile robotic platforms Xavier, Rhino and Minerva [34, 6, 38] can also be operated over the Web. Their interfaces relied on client-pull and server-push techniques to provide visual feedback of the robot's movements; this includes images taken by the robot as well as a java-animated map indicating the robot's current position. However, these interfaces do not include any techniques to reflect changes of the environment. 3D graphics visualizations for Internet-based robot control have already been suggested by Hirukawa et al. [22]. Their interface allows Web users to carry out manipulation tasks with a mobile robot, by controlling a 3D graphics simulation of the robot contained in the Web browser. The robots described in this paper use video streams to convey visual information to the user. Additionally, they provide online visualizations in a virtual 3D environment. This allows the users to choose arbitrary viewpoints and leads to significant reductions of the required communication bandwidth. Furthermore, our interface provides accurate visualization of the persons in the vicinity of the robot which makes the navigation behavior of the robot easier to understand for remote users.

In this work we deal with robots that operate in exhibition spaces, serving at the same time web- as well as on-site visitors. This endeavor has evolved as a pure research and development activity in the involved laboratories, as well as in the formal framework of two European Commission funded projects, namely TOURBOT [40] and WebFAIR [43]. TOURBOT dealt with the development of an interactive tour-guide robot able to provide individual access to museums' exhibits over the Internet. The successful course of TOURBOT and the vision to introduce corresponding services to the more demanding case of trade fairs, resulted in launching WebFAIR. Additionally, WebFAIR introduces tele-conferencing between the remote user and on-site attendants and employs a multi-robot platform, facilitating thus simultaneous robot control by multiple users. In this paper, we also report on the demonstration events that took place in the framework of TOURBOT and argue on the drastic reduction of the system set-up time that was achieved.

2 Mapping

In order to navigate safely and reliably, mobile robots must be able to create suitable representations of the environment. Maps are also necessary for human-robot interaction since they allow users to direct the robot to places in the environment. Our current system uses two different mapping techniques. The first approach is an incremental technique for simultaneous localization and mapping that is highly efficient and uses grid maps to deal with environments of arbitrary shape but lacks the capability of global optimization especially when larger cycles have to be closed. The second approach relies on line-features and corner-points. It uses a combination of a discrete (Hidden Markov) model and a continuous (Kalman-filter) model [4] and applies the EM-algorithm to learn globally consistent maps. Both methods offer very promising alternatives, each one with its own merits. In environments with no clearly defined structure (walls, corridors, corners, etc), the former method is more adequate, at the price of slightly decreased loop-closing capabilities. When the environment structure becomes evident, the latter method can be employed, to render more robust loop closing. Both mapping approaches are described in the remainder of this section.

2.1 Incremental Mapping using Grid Maps

This first mapping technique is incremental and employs occupancy grid maps [26]. To learn such occupancy grids we use an incremental mapping scheme that has been previously employed with great success [39, 19]. Mathematically, we calculate a sequence of robot poses $\hat{l}_1, \hat{l}_2, \ldots$ and corresponding maps by maximizing the marginal likelihood of the t-th pose and map relative to the $(t-1)$-th pose and map:

$$\hat{l}_t \;=\; \operatorname*{argmax}_{l_t} \{ p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) \cdot p(l_t \mid u_{t-1}, \hat{l}_{t-1}) \} \tag{1}$$

The term $p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$ is the probability of the most recent measurement s_t given the pose l_t and the map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$ constructed so far. The term $p(l_t \mid u_{t-1}, \hat{l}_{t-1})$ represents the probability that the robot is at location l_t provided that the robot was previously at position $\hat{l}_{t-1}$ and has carried out (or measured) the motion u_{t-1}. The resulting pose $\hat{l}_t$ is then used to generate a new map $\hat{m}$ via the standard incremental map-updating function presented in [26]:

$$\hat{m}(\hat{l}^t, s^t) \;=\; \operatorname*{argmax}_{m} p(m \mid \hat{l}^t, s^t) \tag{2}$$

The overall approach can be summarized as follows. At any point $t-1$ in time the robot is given an estimate of its pose $\hat{l}_{t-1}$ and a map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$. After the robot moved further on and after taking a new measurement s_t, the robot determines the most likely new pose $\hat{l}_t$. It does this by trading off the consistency of the measurement with the map (first term on the right-hand side in (1)) and the consistency of the new pose with the control action and the previous pose (second term on the right-hand side in (1)). The map is then extended by the new measurement s_t, using the pose $\hat{l}_t$ as the pose at which this measurement was taken.

Our algorithm to 2D scan matching is an extension of the approach presented in [39]. To align a scan relative to the map constructed so far, we compute an occupancy grid map $\hat{m}(\hat{l}^{t-1}, s^{t-1})$ [26, 39] out of the sensor scans obtained so far. Additionally to previous approaches, we integrate over small Gaussian errors in the robot pose when computing the maps. This increases the smoothness of the map and of the likelihood function to be optimized and, thus, facilitates range registration. To maximize the likelihood of a scan with respect to this map, we apply a hill climbing strategy.

2.2 Feature-based Mapping

In the case of structured environments, localization accuracy can be increased by constructing and employing feature based maps of the environment. Our feature-based mapping algorithm utilizes line segments and corner points which are extracted out of laser range measurements. We apply a variant of the Iterative-End-Point-Fit algorithm [25] to cluster the end-points of a range scan into sets of collinear points. Corner points are then computed at the intersections of directly adjacent line segments [5]. Data association (feature matching) during mapping is performed based on a dynamic programming string-search algorithm [4]. This algorithm exploits information contained in the spatial ordering of the features. Simultaneously, the dynamic programming implementation furnishes it with computational efficiency.

Existing approaches to simultaneous mapping and localization usually store all map features and the pose of the robot in a single complex random variable [35]. This approach is computationally demanding since its space and time complexity grows quadratically in the number of features [15]. In contrast, our approach treats map features as parameters of the dynamical system according to which the robot's state evolves. Therefore, the problem is reformulated

as to simultaneously determine the state and the parameters of a dynamical system; that is, a learning problem, that is solved via a variant of the EM-algorithm [11, 7, 39].

In the mapping context, the E-step is responsible for calculating the state of the robot at each point t in time. Since the approach described here is an off-line mapping technique, all past and future observations are available and can be used to estimate the state at time t. The problem of estimating variables given both past and future observations is denoted as "smoothing." A very popular method for performing smoothing is the Raugh-Tung-Striebel smoother [30]. This algorithm consists of two steps. The first step (forward step) is the Extended Kalman Filter (EKF) forward recursions. For each time instant t, it estimates the mean μ_{x_t} and covariance Σ_{x_t} using an EKF. The second step is a backward recursion. It starts with the final measurement and recursively estimates maximum a-posteriori estimates for the previous states μ_{x_t} and Σ_{x_t} as:

$$\Sigma_{x_t} = \Sigma_{x_t} + S[\Sigma_{x_{t+1}} - \Sigma_{x_{t+1}^-}]S^T \tag{3}$$

$$\mu_{x_t} = \mu_{x_t} + S[\mu_{x_{t+1}} - \mu_{x_{t+1}^-}] \tag{4}$$

where

$$S = \Sigma_{x_t} \bigtriangledown {F_x}^T \Sigma^{-1}_{x_{t+1}^-}. \tag{5}$$

Here $\bigtriangledown F_x$ denotes the Jacobian of the transition function F with respect to μ_{x_t}.

To detect and close loops during mapping, our algorithm relies on the global localization capabilities of a hybrid method based on a switching state-space model [4]. This approach applies multiple Kalman trackers assigned to multiple hypotheses about the robot's state. It handles the probabilistic relations among these hypotheses using discrete Markovian dynamics. Hypotheses are dynamically generated by matching corner points extracted from measurements with corner points contained in the map. Hypotheses that cannot be verified by observations or sequences of observations become less likely and usually disappear quickly.

Our algorithm (see Figure 1) iterates the E- and the M-step until the overall process has converged or until a certain number of iterations has been carried out. Our current system always starts with a single hypothesis about the state of the system. Whenever a corner point appears in the robot's measurements, new hypotheses will be created at corresponding positions. On the other hand, hypotheses that cannot be confirmed for a sequence of measurements typically vanish. The resulting map always corresponds to the most likely hypothesis.

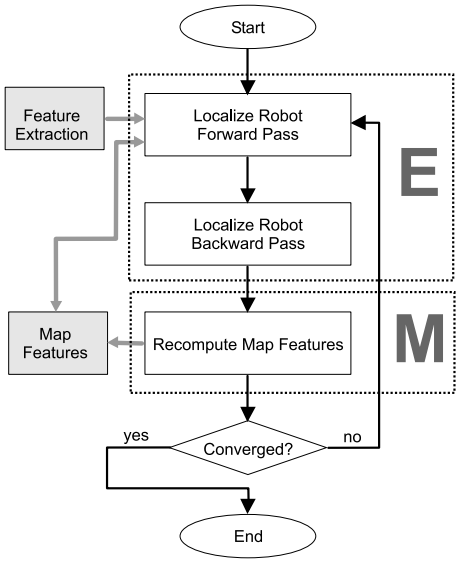

Figure 1. Flow-gram of the Iterative Mapping Algorithm.

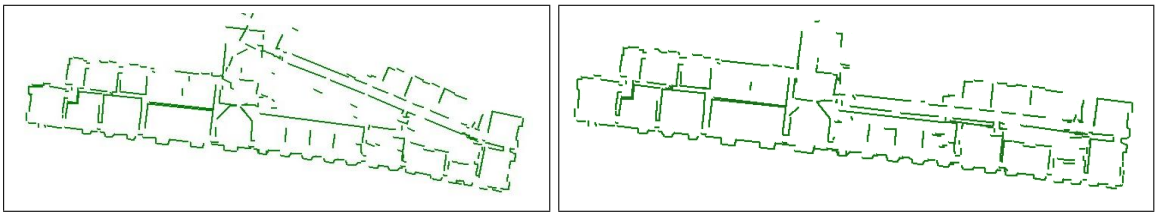

Figure 2. Line feature maps of an exhibition site: Original data (left image) and map generated by our algorithm (right image).

The left image of Figure 2 shows a typical map of an exhibition site computed based on laser range information and the pure odometry data gathered with a mobile robot. As can be seen from the image the odometry error becomes too high to generate a consistent map. The right image of Figure 2 shows the map obtained with the approach described above. As the figure illustrates, our algorithm was able to correctly eliminate the odometry error although it had to close serval cycles. The overall computation time to compute this map was one hour on a standard Pentium IV personal computer.

3 Fusion of Laser and Visual Data

Laser scanners have proven to be very reliable sensors for navigation tasks, since they provide accurate range measurements in large angular fields and at very fast rates. However, laser range scans are 2D representations of a 3D world. Thus, the underlying assumption is that laser profiles accurately model the shape of the environment also along the vertical dimension which is invisible to the laser. Although this assumption is justified in many indoor environments, various objects such as chairs, tables, and shelves usually have a geometry that is not uniform over their height. Accordingly, not all aspects of such objects are visible in laser-range scans. The capability to correctly identify objects in the vicinity of the robot, however, is particularly important for robots in real-world environments such as exhibitions, since it is a basic precondition for the safety of the robot and the exhibits.

One potential solution to this problem could be to exploit 3D laser scanners, which unfortunately are very expensive. The alternative solution is to utilize additional sources of information, such as vision, to infer 3D information. Our system exactly follows this approach. It uses the 2D structure acquired with the laser-range scanner and, based on this, it computes a $2\frac{1}{2}$D representation by introducing vertical planar walls for the obstacles in the 2D map. We then exploit camera information to (a) validate the correctness of the constructed model and (b) qualitatively and quantitatively characterize inconsistencies between laser and visual data wherever such inconsistencies are detected.

The employed method [3] operates as follows. At time $(t-1)$ the robot acquires a laser range scan s_{t-1} and an image i_{t-1}. Based on s_{t-1} the robot builds a $2\frac{1}{2}$D model of the environment. The same process is applied at time t resulting in i_t and s_t. Using the world model derived at

Figure 3. Images captured at time $t-1$ (left image) and time t (center image) and results of the evaluation process projected on the image captured at time t (right image).

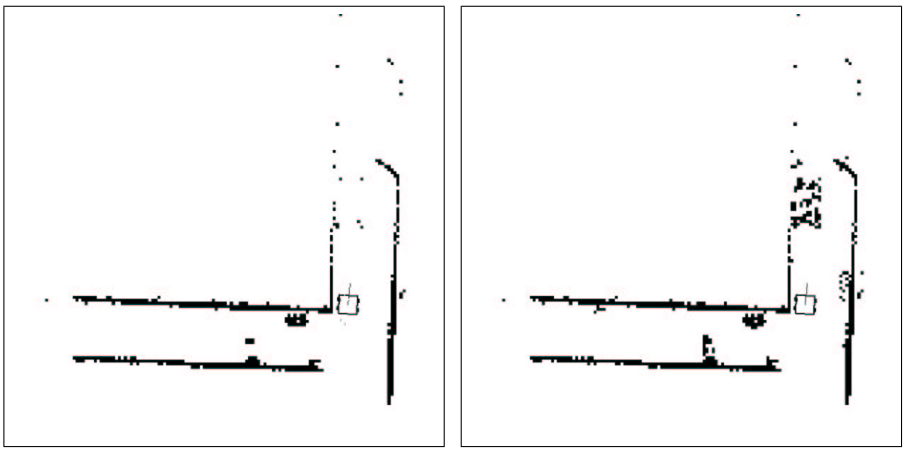

Figure 4. Occupancy grid maps computed based on the fusion of vision and laser data. The left image shows the map computed from the laser-range data. The right image shows the resulting map after combining vision and laser data.

time $(t-1)$ and the motion of the robot u_{t-1}, the image i_{t-1} is back-projected to the reference frame of image i_t, resulting in the image $\hat{i}_t$. Images $\hat{i}_t$ and i_t are identical in areas where the $2\frac{1}{2}$D model is valid but differ in areas where this model is invalid. To identify obstacles not detected by the 2D range scans, we perform a local correlation of the intensity values in $\hat{i}_t$ and i_t. The inconsistencies between the laser and visual data are then converted to real word coordinates along epipolar lines and then are accumulated in a 2D occupancy map.

Figures 3 and 4 illustrate a typical application example in the corridor environment at ICS/FORTH. In this case the robot travels along a corridor with several objects that are invisible to the laser scanner, such as tables, fire extinguishers, and a wall cabinet. The left and center pictures of Figure 3 show the images grabbed by the robot at time $t-1$ and at time

t. The right image of Figure 3 shows the results of the obstacle detection technique. Regions with inconsistencies are marked with crosses. As the figure illustrates, the objects that cannot be sensed with the laser range finder are successfully detected by our algorithm. Finally, Figure 4 shows the occupancy grid maps obtained without considering visual information (left image) and obtained after integrating vision and laser data. The map generated by our algorithm provides a more accurate representation of the environment since it allows the robot to avoid collisions with obstacles that cannot be sensed by the laser scanner.

4 People Tracking

Tour-guide robots, by definition, operate in populated environments. Knowledge about the position and the velocities of moving people can be utilized in various ways to improve the behavior of tour-guide robots. For example, it can enable a robot to adapt its velocity to the speed of the people in the environment, it can be used by the robot to improve its collision avoidance behavior in situations where the trajectory of the robot crosses the path of a human, etc. Moreover, being able to keep track of moving people is an important prerequisite for human-robot interaction.

Our robots apply sample-based joint probabilistic data association filters (SJPDAFs) [32] to estimate the positions of people in the vicinity of the robot. In contrast to standard data association filters (JPDAFs) [10], which rely on Gaussian distributions to represent the beliefs about the position of the objects being tracked, SJPDAFs apply particle filters [18, 29] for that purpose. In particular they use a set of particle filters to keep track of the individual persons in the vicinity of the robot. The particle filters are updated according to the sensory input, using a model of typical motions of persons. The approach computes a Bayesian estimate of the correspondence between features detected in the data and the different objects to be tracked. It then uses this estimate to update the individual particle filters with the observed features.

Suppose there are K persons and let $\mathbf{X}^t = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_K^t\}$ be the states of these persons at time t. Note that each $\mathbf{x}_i^t$ is a random variable ranging over the state space of a single person. Furthermore, let $\mathbf{Z}(t) = \{\mathbf{z}_1(t), \ldots, \mathbf{z}_{m_t}(t)\}$ denote a feature set observed at time t, where $\mathbf{z}_j(t)$ is one feature of such a set. $\mathbf{Z}^t$ is the sequence of all feature sets, up to time t. The key question when tracking multiple persons is how to assign the observed features to the individual objects.

In the JPDAF framework, a joint association event θ is a set of pairs $(j, i) \in \{0, \ldots, m_t\} \times \{1, \ldots, K\}$. Each θ uniquely determines which feature is assigned to which object. Please note, that in the JPDAF framework, the feature $\mathbf{z}_0(t)$ is used to model situations in which an object has not been detected, i.e. no feature has been found for object i. Let Θ_{ji} denote the set of all valid joint association events which assign feature j to the object i. At time t, the JPDAF considers the posterior probability that feature j is caused by object i:

$$\beta_{ji} \quad = \quad \sum_{\theta \in \Theta_{ji}} P(\theta \mid \mathbf{Z}^t). \tag{6}$$

According to [32], we can compute the β_{ji} as

$$\beta_{ji} \quad = \quad \sum_{\theta \in \Theta_{ji}} \alpha \, \gamma^{(m_t - |\theta|)} \prod_{(j,i) \in \theta} p(\mathbf{z}_j(t) \mid \mathbf{x}_i^t). \tag{7}$$

It remains to describe how the beliefs $p(\mathbf{x}_i^t)$ about the states of the individual objects are represented and updated. In our approach [32], we use sample-based representations of the individual beliefs. The key idea is to represent the density $p(\mathbf{x}_i^t \mid \mathbf{Z}^t)$ by a set $\mathbf{S}_i^t$ of N weighted, random samples or *particles* $s_{i,n}^t (n = 1 \ldots N)$. A sample set constitutes a discrete approximation of a probability distribution. Each sample is a tuple $(x_{i,n}^t, w_{i,n}^t)$ consisting of state $x_{i,n}^t$ and an importance factor $w_{i,n}^t$. The *prediction* step is realized by drawing samples from the set computed in the previous iteration and by updating their state according to the prediction model $p(\mathbf{x}_i^t \mid \mathbf{x}_i^{t-1}, \delta t)$. In the *correction* step, a feature set $\mathbf{Z}(t)$ is integrated into the samples obtained in the prediction step. Thereby we consider the assignment probabilities β_{ji}. In the sample-based variant, these quantities are obtained by integrating over all samples:

$$p(\mathbf{z}_j(t) \mid \mathbf{x}_i^t) \quad = \quad \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{z}_j(t) \mid x_{i,n}^t). \tag{8}$$

Given the assignment probabilities we may then compute the weights of the samples

$$w_{i,n}^t = \alpha \, \sum_{j=0}^{m_t} \beta_{ji} p(\mathbf{z}_j(t) \mid x_{i,n}^t), \tag{9}$$

where α is a normalizer ensuring that the weights sum up to one over all samples. Finally, we obtain N new samples from the current samples by bootstrap resampling. For this purpose we select every sample $x_{i,n}^t$ with probability $w_{i,n}^t$.

In our system we apply the SJPDAF to estimate the trajectories of persons in range scans. Since the laser range scanners mounted on our platforms are at a height of approximately 40

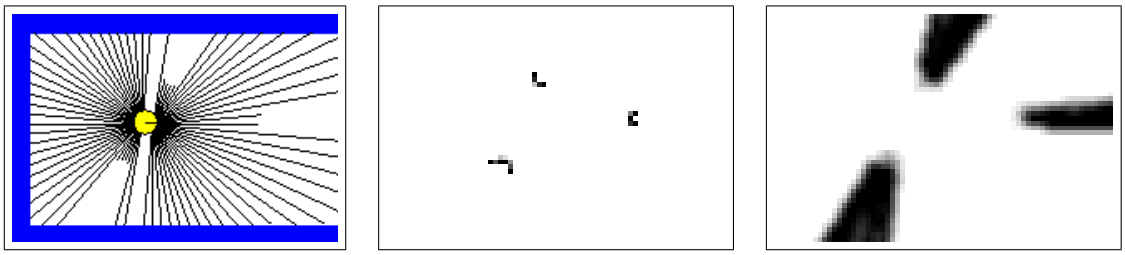

Figure 5. Typical laser range finder scan. Two of the local minima are caused by people walking by the robot (left image). Features extracted from the scan, the grey-level represents the probability that a person's legs are at the position (center). Occlusion grid, the grey-level represents the probability that the position is occluded (right image).

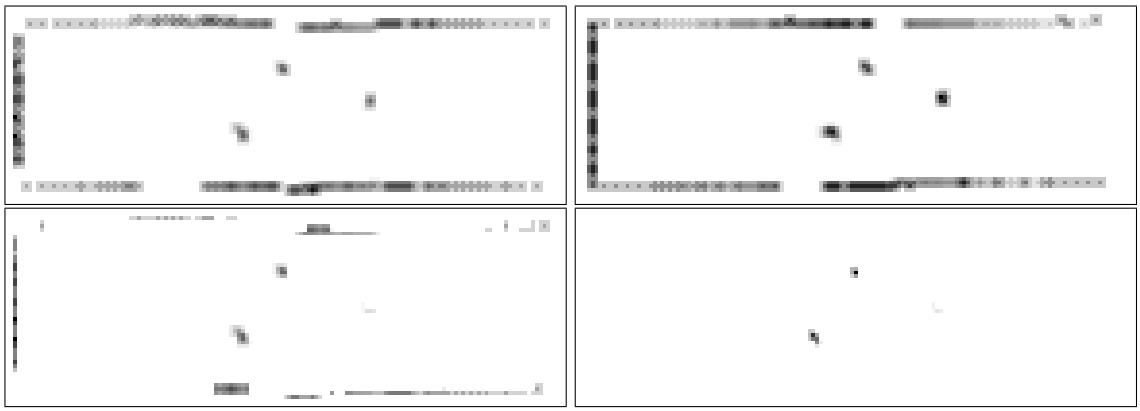

Figure 6. From left to right, top-down: the occupancy map for the current scan, the occupancy map for the previous scan, the resulting difference map, and the fusion of the difference map with the feature maps for the scan depicted in Figure 5

cm, the beams are reflected by the legs of the people which typically appear as local minima in the scans. These local minima are used as the features of the SJPDAF (see left and middle part of Figure 5). Unfortunately, there are other objects which produce patterns similar to people. To distinguish these static objects from moving people our system additionally considers the differences between occupancy probability grids built from consecutive scans. Static features are filtered out. This is illustrated in Figure 6.

Finally, we have to deal with possible occlusions. We therefore compute a so-called "occlusion map" containing, for each position in the vicinity of the robot, the probability that the corresponding position is not visible given the current range scan (see right part of Figure 5). The whole process is described in detail in [32].

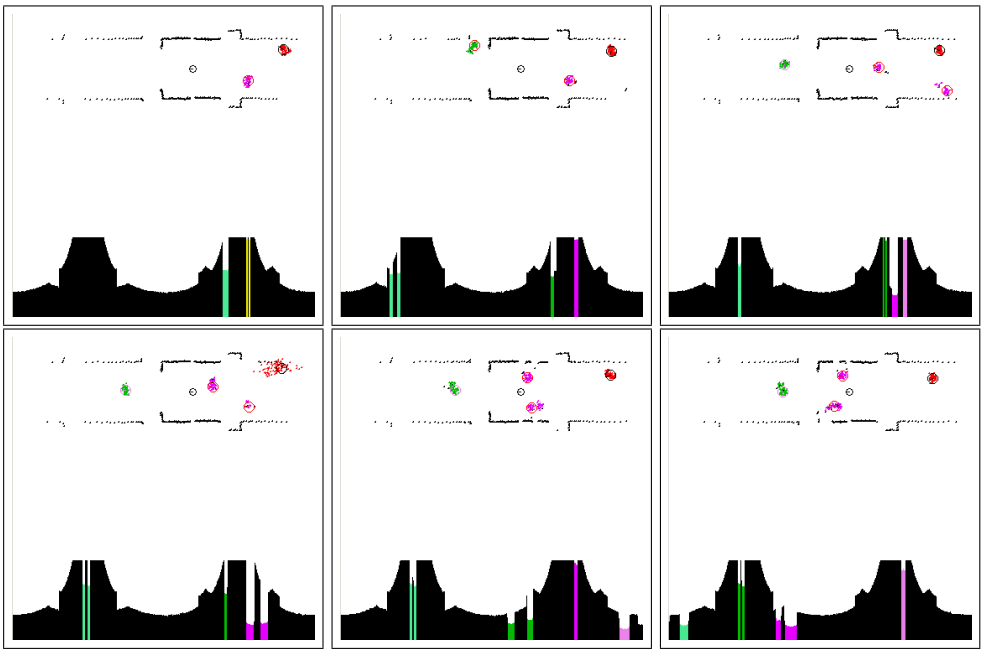

Figure 7. Tracking people using laser range-finder data.

Figure 7 shows a typical situation, in which the robot is tracking up to four persons in a corridor environment. The upper part of each image depicts the end-points of the distance scans obtained with the two SICK PLS laser range scanners covering the whole surrounding of the robot, the pose of the robot, the particles of the individual particle filter, and the estimated positions of the persons corresponding to the means of the distributions represented by the individual particle filters. The lower part of each image illustrates the polar histograms of the corresponding range scans. As can be seen in the figure, our approach is robust against occlusions and can quickly adapt to changing situations in which additional persons enter the scene. For example, in the lower left image the upper right person is not visible in the range scan, since it is occluded by a person that is closer to the robot. The knowledge that the samples lie in an occluded area prevents the robot from deleting the corresponding sample set. Instead, the samples only spread out, which correctly represents the growing uncertainty of the robot about the position of the person.

5 Mapping in Populated Environments

Mapping approaches generally deal with static environments. In populated environments, however, people in the vicinity of the robots may appear as objects in the resulting maps and

therefore make the maps not usable for navigation tasks. The previously described people tracking technique can be combined with our mapping techniques, leading to several advantages. First, by incorporating the results of the people tracker, the localization becomes more robust. Additionally, the resulting maps are more accurate, since measurements corrupted by people walking by are filtered out. Compared to alternative techniques such as [41] our approach uses a tracking technique and therefore is able to predict the positions of the person's even in situations in which the corresponding features are temporarily missing. To utilize the results of the people tracker during mapping, we need to describe how we perform the range scan registration and how we filter beams reflected by persons during the map generation process.

During scan alignment we need to know the probability $P(\text{hit}_{x,y} \mid \mathbf{X}^t)$ that a beam ending at position $\langle x, y \rangle$ is reflected by a person. In our current implementation, we consider the individual persons independently:

$$P(\text{hit}_{x,y} \mid \mathbf{X}^t) = 1 - \prod_{i=1}^{K} \left(1 - P(\text{hit}_{x,y} \mid \mathbf{x}_i^t)\right). \tag{10}$$

In this equation $P(\text{hit}_{x,y} \mid \mathbf{x}_i^t)$ is the likelihood that a beam ending at position $\langle x, y \rangle$ is reflected by person i, given the state $\mathbf{x}_i^t$ of that person. To compute this quantity, we construct a two-dimensional normalized histogram over a 15x15 cm discretization of the environment by counting how many samples representing the belief about $\mathbf{x}_i^t$ fall into each bin.

Next we need to specify how to determine the likelihood of a beam given the information obtained from the people tracking system. Suppose x_b and y_b are the coordinates of the cell in which the beam b ends. Accordingly, we can compute the probability $P(\text{hit}_b \mid \mathbf{x}_i^t)$ that a beam b is reflected by a person as

$$P(\text{hit}_b \mid \mathbf{X}^t) \quad = \quad P(\text{hit}_{x_b,y_b} \mid \mathbf{X}^t). \tag{11}$$

Finally, it remains to describe how we incorporate the quantity $h_b = P(\text{hit}_b \mid \mathbf{X}^t)$ into the scan alignment process. If we consider all beams as independent, the likelihood $p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$ of the most recent measurement given all previous scans is obtained as:

$$p(s_t \mid l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) \quad = \quad \prod_{b \in s_t} p(b \mid \hat{m}(\hat{l}^{t-1}, s^{t-1})))^{(1-h_b)}. \tag{12}$$

Thus, during the scan alignment we weigh each beam b according to the probability $1 - P(\text{hit}_b \mid \mathbf{X}^t)$. Note that this is a general form of a situation in which it is exactly known whether or not

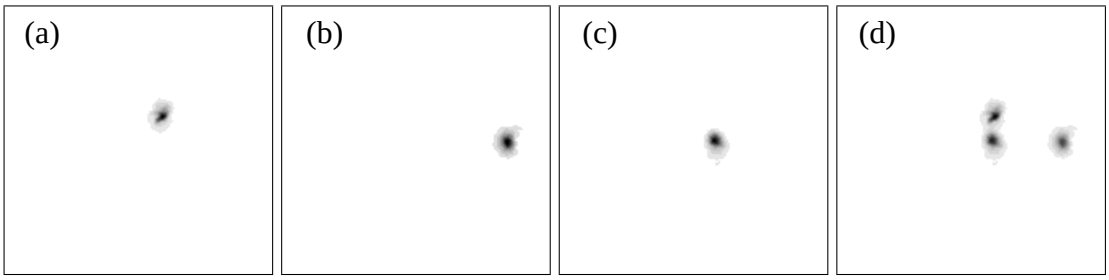

Figure 8. Example situation in which the quantity $P(\text{hit}_{x,y} \mid \mathbf{X}^t)$ (image d) is computed by combining the histograms for three individual trackers (image a-c).

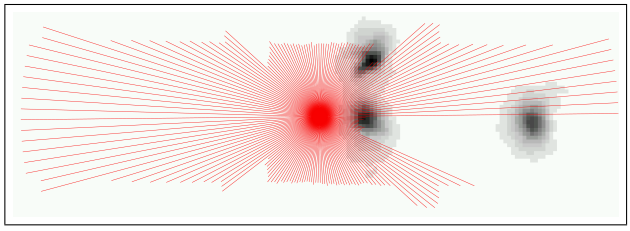

Figure 9. The weight of a laser beam is computed according to the value $P(\text{hit}_{x,y} \mid \mathbf{X}^t)$ of the cell in which it ends.

b is reflected by a person. If b is known to be reflected by a person, h_b equals 1 such that b does not change the likelihood of the scan.

Figures 8 and 9 show a typical application example. Here the robot is tracking three different persons. The histograms corresponding to the three sample sets that represent the belief about the person are depicted in pictures (a)-(c) (the darker the more likely it is that the corresponding area is covered by a person). The resulting belief according to Equation (10) is depicted in Figure 8(d). Finally, Figure 9 shows the current scan of the robot overlayed to the histogram depicted in Figure 8(d). According to Equation 12, the beams ending in a dark area have a lower weight during the scan alignment.

The second task is to filter out beams reflected by persons to avoid spurious objects in the resulting maps. In our current system we compute a bounding box for each sample set $\mathbf{S}_i^t$ and integrate only those beams whose endpoint does not lie in any of the bounding boxes. To cope with the possible time delay of the trackers, we also ignore corresponding beams of several

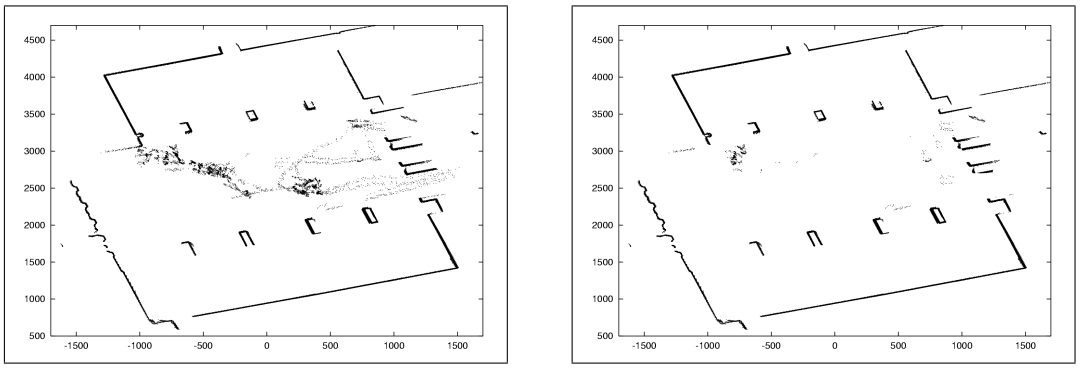

Figure 10. Maps of the Byzantine and Christian Museum in Athens created without (left) and with (right) people filtering.

previous and next scans before and after the person was detected. Note that one can be rather conservative during the map generation process, because the robot generally scans every part of the environment quite often. However, during scan alignment, a too conservative strategy may result in too few remaining beams which leads to reduced accuracy of the estimated positions.

Figure 10 shows maps of the Byzantine and Christian Museum in Athens that were recorded with and without incorporating the results of the people-tracker into the mapping process. Both maps actually were generated using the same data set. While the robot was gathering the data, up to 20 people were moving in this environment. The left image shows the endpoints of the laser-range data after localization. Clearly, the presence of moving persons resulted in many obstacles that do not correspond to permanent features of the environment, with an obvious effect to navigation tasks. The right image of Figure 10 shows the map resulting from our approach. As can be seen in this figure, our robot is able to eliminate almost all measurements attributed to moving people so that the resulting map provides a better representation of the true state of the static environment.

6 Interfaces

Robots in museums and exhibitions should be able to interact with on-site visitors in a natural way and to allow distant visitors to feel like being present in the site. Thus, the employment of intuitive human-robot interfaces is of paramount importance to the acceptance and the success

of the overall system. The interfaces should be tailored to the type of user; clearly there are similarities as well as important differences between distant and on-site users.

6.1 Web Interface

The developed web-interface has been designed to provide enhanced functionality and ease of use. Compared to interfaces of previous systems such as Xavier, Rhino and Minerva [34, 8, 33], it allows personalized control of the robot(s) with a number of desirable features. Instead of image streams that are updated via server-push or client-pull technology, it uses a commercial live streaming video and broadcast software [42] that provides continuous video transmissions to transfer images recorded with the robot's cameras to the remote user. Additionally, web-users have a more flexible control over the robot. They can control the robot exclusively for a fixed amount of time which generally is set to 10 minutes per user. Whenever a user has control over the robot, he/she can direct it to arbitrary points in the exhibition. The user can select from a list of predefined guided tours or direct the robot to visit particular exhibits or locations in the exhibition. At each point in time, the user can request a high-resolution image grabbed with the cameras maximal resolution. Furthermore, the interface allows the control of the pan-tilt unit of the robot. Thus, the user can look at any user-defined direction. Finally, the user may request the robot to move around an exhibit in order to view it from several directions.

The control page of the interface is depicted in Figure 11. The left side contains the predefined tours offered to the user as well as the list of exhibits that are known to the robot. The center shows the live-stream as well as a Java applet animating the robot in a 2D floor-plan. This map can also be used to directly move the robot to an exhibit or to an arbitrary location in the exhibition. Between the map and the live-stream, the interface includes control buttons as well as a message window displaying system status messages. The right part of the interface shows multi-media information about the exhibit including links to relevant background information.

6.1.1 Enhanced Visualizations

Once instructed by a Web user, the robot fulfills its task completely autonomously. Since the system also operates during opening hours, the robot has to react to the visitors in the museum.

Figure 11. Web interface of the TOURBOT system for exclusive control over the robot.

This makes it impossible to predict the robot's course of action beforehand. Therefore, it is highly important, to visualize the environment of the robot and the moving people therein, so that the web user gets a better understanding of what is going on in the museum and why the robot is carrying out the current actions.

A typical way of providing information to the users is video streams, recorded with static or robot-mounted cameras. This, however, has the disadvantage of limited perspectives and high bandwidth requirements. For these reasons, we developed a control interface, which provides the user with a virtual reality visualization of the environment including the robot and the people in its vicinity. Based on the state information received from the robot and our tracking algorithm, our control interface continuously updates the visualization. Depending on the level of detail of the virtual reality models used, the Internet user can obtain visualizations, whose quality is comparable to video streams. For example, Figure 12 shows two sequences of visualizations provided during the installation of the system in the Deutsches Museum Bonn in November 2001 along with images recorded with a video camera and with the robot's on-board camera.

Within the graphics visualization, people are shown as avatars. As can be seen, the visual-

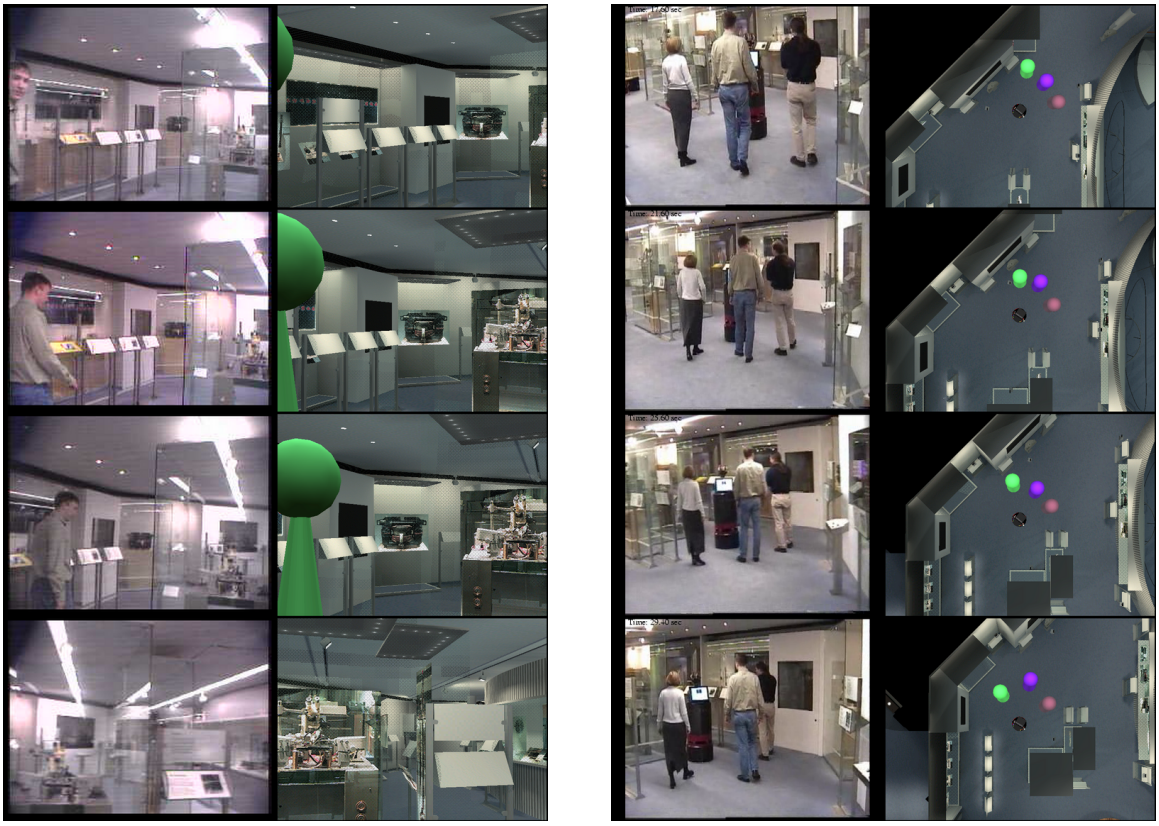

Figure 12. The enhanced 3D visualization allows arbitrary view-points. The left sequence shows the real and the virtual view through the robot's cameras. The right images show the robot guiding three people through the museum and a bird's eye view of the scene.

Figure 13. Person interacting with Albert during a Hannover trade fair demonstration.

ization is almost photo-realistic and the animated avatars capture the behavior of the people in the scene quite well.

Compared to the transmission of video streams, the graphics-based visualization highly reduces the bandwidth requirements of the control interface. TOURBOT's standard web interface used a single video stream to transmit images of 240 by 180 pixels in size with a frame rate of about 5 Hz. This still required a bandwidth of about 40kBit/s. Compared to that, the graphics-based visualization only needs about 1kBit/s to achieve the same frame rate, if we assume that 7 people are constantly present in the robot's vicinity. It has the additional advantage, that the bandwidth requirement is independent of the image size. The graphics-based solution, therefore, allows for more detailed visualizations. Beyond the bandwidth savings, the graphics-based visualization offers an increased flexibility to the Internet user. Virtual cameras can be placed anywhere and the viewpoints can even be changed at run-time, as illustrated in the right image sequence of Figure 12. Our current prototype implements these ideas. It uses Open Inventor models of the robot and of the environment for the 3D rendering. On start-up, the control interface connects to the robot via TCP/IP and after downloading the model, the visualization component receives state information from the robot and starts rendering the scene accordingly.

6.2 On-board Interface

Besides the web interface that is used by remote users, the robots have several means for communicating and interacting with the on-site visitors. A reduced version of the web interface is also displayed in a touch screen appropriately mounted at the rear side of the robot. Through this touch screen, the on site visitors may instruct the robot to guide them to specific exhibits. One of the main differences between the web and the on-board interface is that video streams and enhanced visualizations are not provided, since they are not actually required by the on-site visitors. Instead, the on-board interface makes extensive use of synthetic speech. To enhance the communication with users in the museum, the robots are equipped with a speaker-independent speech interface. We employ a commercially available speech system [28] that detects simple phrases. The input of the user is processed and the parsed phrase is used to generate corresponding actions. To improve the recognition rate, the software allows the definition of contexts, i.e., sets of phrases that are relevant in certain situations. Depending on user input or depending on the task that is currently carried out, the system can dynamically switch between the different contexts. The current system includes 20 different phrases, that can be used to request information about the robot, the exhibition site, or even the time and the weather. In several installations in populated environments we figured out that the overall recognition rate is approximately 90%. Figure 13 shows a scene in which a person interacts with the robot Albert during the Hannover trade fair in 2001. Here the person asked several questions about the robot and requested information about the time (*who are you?*, *where are you from?*, *what are you doing here?*). Depending on the input of the user the robot can dynamically generate speech output. The text to be spoken is converted into audio files that are played back.

Another very important aspect of the on-board interface is the capability of the robots to alter the facial expressions of their mechanical heads based on their internal status. Currently, there are three different facial expressions of the robot, namely “happy”, “neutral” and “angry”. These facial expressions are implemented by modifying the shape of the eyebrows and the mouth. Combined with a variety of voice messages, the robot uses these expressions to inform the on-site visitors regarding its internal status. For example, if the path of the robot is not obstructed by the on-site visitors, the robot appears happy. In the opposite case, the robot’s “mood” changes progressively in time. Moreover, the head of the robot is controlled so that it

looks towards the direction of intended motion. This way, on-site visitors adapt their motion so as not to obstruct its path.

7 System Installation and Demonstration

In the framework of the TOURBOT project a number of demonstration trials were undertaken in the premises of the participating museums. More specifically, the TOURBOT system has first been developed and fully tested in the laboratory environment. Following that, and in order to acquire performance data from actual museum visitors, the system has been installed and demonstrated in the three museums that participated in the project consortium. These demonstrations were combined with relevant events in order to publicize and disseminate the results of the project to professionals and the broader public. Factual information of these events is as follows:

- Foundation of the Hellenic World, Athens, Greece, May 28–June 2, 2001. Exhibition: "Crossia, Chitones, Doulamades, Velades - 4000 Years of Hellenic Costume." The exhibition area comprised 2000 square meters. During the trial the robot operated approximately 60 hours covering a distance of 14 kilometers. More than 1200 web users observed the exhibition through TOURBOT. A typical situation, in which the robot Lefkos guides visitors through the museum is shown in Figure 14(a).

- Deutsches Museum Bonn, Bonn, Germany, November 6–11, 2001 (see Figure 14(b)). Exhibition: "Part of the permanent exhibition, highlighting scientific achievements that were awarded the Nobel Prize." The exhibition area in which the robot moved comprised about 200 square meters. The system operated about 60 hours, covering a distance of 10 km. Approximately 1900 web visitors had a look around the museum via the robot.

- Byzantine and Christian Museum, Athens, Greece, December 3–7, 2001 (see Figure 14(c)). Exhibition: "Byzantium through the eyes of a robot." The exhibition area comprised about 330 square meters. During the trial the robot operated 40 hours, covering a distance of 5.3 kilometers. The number of web users was small in this trial, since during the first day of the trial at the Byzantine and Christian Museum a large number of (on-site) visitors were coming to the exhibition. This forced the TOURBOT team to the

Figure 14. (a) Robot Lefkos operating in the exhibition of the Foundation of the Helenic World. (b) Robot Rhino operating in the Deutsches Museum Bonn. (c) Robot Lefkos operating in the Byzantine and Christian Museum. (d) Robot Albert interacting with a person at the Heinz Nixdorf MuseumsForum. This picture is curtesy of Jan Braun, Heinz Nixdorf MuseumsForum.

decision to devote significantly more time of the system to on-site visitors as opposed to web visitors.

Additionally, TOURBOT was installed and operated for a longer period of time (Oct. 2001–Feb. 2002) at the Heinz Nixdorf MuseumsForum (HNF) in Paderborn, Germany (see Figure 14(d)). This was in the framework of the special exhibition “Computer.Gehirn” (Computer.Brain) with a focus on the comparison of the capabilities of computers/robots and human beings. Also in June 2002, TOURBOT was introduced for one week in the Museum of Natural History of the University of Crete, Heraklion, Greece.

7.1 Installation Time

The large number of test installations of the TOURBOT system required sophisticated tools for the setup of the overall system. The most crucial part of the overall procedure is the generation of the navigation map. However, based on the techniques described above, the overall mapping process could in all cases be accomplished within several hours. To avoid that the robot leaves its desired operational space or collides with obstacles that cannot be sensed, we manually create a second map, depicting such obstacles. This map is then fed to the collision avoidance module [6], thus preventing the robot from moving into the corresponding areas.

A further time consuming process is the generation of the multimedia-content that is presented to the user for each exhibit. The TOURBOT system includes a generic Multimedia database including HTML-pages, images, audio, and video sequences. Material in the database can be changed and/or edited using available software tools. Furthermore, the robot is equipped with a task specification that defines where the designated exhibits are and which content has to be presented.

Most of the multimedia information pertinent to the exhibits can be obtained directly from the exhibition sites, since pictures, text and other relevant material are often already contained in existing Web presentations.

The whole setup can therefore be accomplished in less than two days. This is an enormous speed-up compared to previous tour-guide systems. Figure 15 shows the time required to install the Rhino and Minerva systems [6, 38] in comparison to that of the TOURBOT system. As can

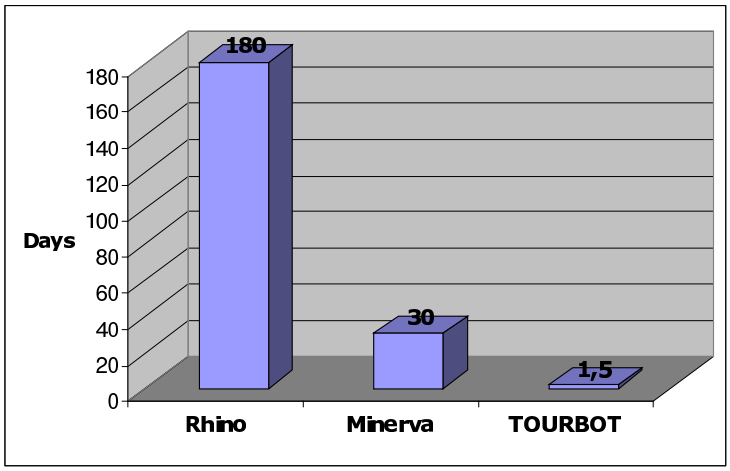

Figure 15. Time required to install the different tour-guide systems Rhino, Minerva, and TOURBOT.

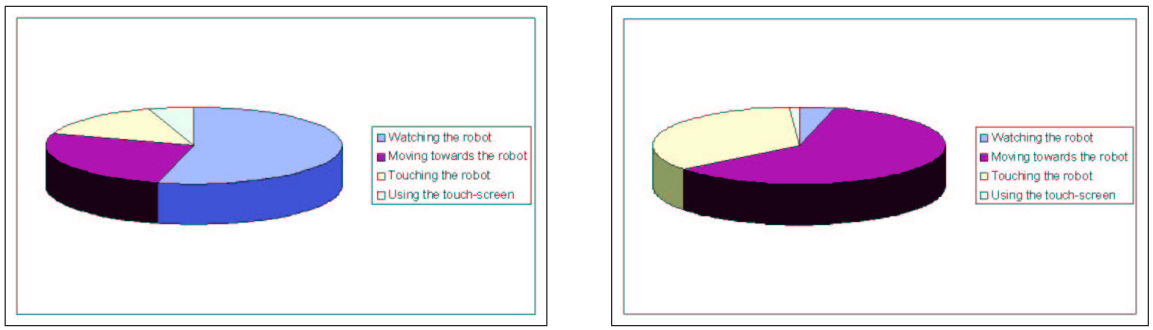

Figure 16. Reactions of Visitors to the Robotic Tour-guide. (a) Adult visitors, (b) Children.

be seen, the TOURBOT system requires significantly less time than Rhino and Minerva. Our experience with tour-guide robots in exhibition sites suggests that 3D models of exhibitions' premises are generally not available. The automatic generation of such models with the mobile robot itself is a subject of ongoing research [20].

7.2 Reactions of the Public

An interesting study was carried out in the context of the above trials regarding the visitor-robot interaction patterns. The observation of the interaction between visitors and the robot in the first two trails (at the Foundation of the Hellenic World, Athens, Greece, and the Deutsches

Museum Bonn, Bonn, Germany) led us to consider carrying out a more detailed study on the human-robot-interaction. This study was planned and carried out during the third trial in the Byzantine and Christian Museum, Athens, Greece. Based on the observations during the previous trials, the human-robot-interaction was qualitatively classified into four behaviors; the results are summarized in Figure 16. Considering these results one may come to some preliminary conclusions regarding the human-robot-interaction:

- Most visitors turn/move to the robot when they see it, be it immediately or after watching it for a while. This indicates that the robotic platform(s) attract the interest and are appealing to the visitors.
- The smaller counts that were observed in the fourth behavior (using the touch-screen) are justified in many ways. A major reason for that is that only one person has the possibility to use the touch-screen at any point in time.
- Comparing adults and children it can be stated that children are uninhibited and natural in their behavior. Significantly more counts relate to the behavior “Moving towards the robot” than to “Watching it”. On the contrary, adults tend to exhibit a more “reserved” behavior towards the robot.
- The performance in the correct use of the robot is better in adults than in children.

8 Conclusions

In this paper we presented a number of techniques that are needed for realizing web-operated mobile robots. These techniques include advanced map building capabilities, a method for obstacle avoidance that is based on a combination of range and visual information and a people tracking method. This last method has also been used to provide enhanced visualizations over the Internet. In addition to video streams our systems provides high-resolution virtual reality visualizations that also include the people in the vicinity of the robot. This increases the flexibility of the interface and simultaneously allows a user to understand the navigation actions of the robot.

The techniques described in this paper have been successfully deployed within the EU-funded projects TOURBOT and WebFAIR which aim at the development of interactive tour-

guide robots, able to serve web- as well as on-site visitors. Technical developments in the framework of these projects have resulted in robust and reliable systems that have been demonstrated and validated in real-world conditions. Equally important, the system set-up time has been drastically reduced, facilitating its porting in new environments.

Our current research extends the navigation capabilities of the robotic systems by addressing obstacle avoidance in the cases of objects that are not visible by the laser scanner [3], 3D mapping [20], mapping in dynamic environments [21], predictive navigation [13], and multi-robot coordination [14]. Moreover, in the context of the above projects additional issues are addressed. They consider (a) how to adapt this technology in order to fit the long-term operational needs of an exhibition site, (b) how to evaluate the robotic system in terms of its impact to the main function and objectives of the exhibition site (financial impact, accessibility, marketing and promotion, impact on visitor demographic, etc.), and (c) how to evaluate the content and educational added value to museum and exhibition visitors, and generate a feedback to the technology developers in order to improve in the future the robotic avatars and adapt further to the needs of the users.

9. Acknowledgments

This work has partly been supported by the by the IST Programme of Commission of the European Communities under contract numbers IST-1999-12643 and IST-2000-29456. The authors furthermore would like to thank the members of the IST-project TOURBOT for helpful comments and fruitful discussions.

References

[1] K.O. Arras and S.J Vestli, *Hybrid, high-precision localisation for the mail distributing mobile robot system MOPS*, Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1998.

[2] H. Asoh, S. Hayamizu, I. Hara, Y. Motomura, S. Akaho, and T. Matsui, *Socially embedded learning of office-conversant robot jijo-2*, Proceedings of IJCAI-97, IJCAI, Inc., 1997.

[3] H. Baltzakis, A. Argyros, and P. Trahanias, *Fusion of range and visual data for the extraction of scene structure information,* Intl. Conf. on Pattern Recognition, (ICPR), 2002.

[4] H. Baltzakis and P. Trahanias, *Hybrid mobile robot localization using switching state-space models,* Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2002.

[5] H. Baltzakis and P. Trahanias, *An iterative approach for building feature maps in cyclic environments,* Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[6] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, *Experiences with an interactive museum tour-guide robot,* Artificial Intelligence **114** (1999), no. 1-2.

[7] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun, *Sonar-based mapping with mobile robots using EM,* Proc. of the International Conference on Machine Learning (ICML), 1999.

[8] W. Burgard and D. Schulz, *Robust visualization for web-based control of mobile robots,* Robots on the Web: Physical Interaction through the Internet (K. Goldberg and R. Siegwart, eds.), MIT-Press, 2001.

[9] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós, *The SPmap: A probabilistic framework for simultaneous localization and map building,* IEEE Transactions on Robotics and Automation **15** (1999), no. 5, 948–953.

[10] I.J. Cox, *A review of statistical data association techniques for motion correspondence,* International Journal of Computer Vision **10** (1993), no. 1, 53–66.

[11] A. Dempster, N. Laird, and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm,* J. R. Statist. Soc. B **39** (1977), 185–197.

[12] H. Endres, W. Feiten, and G. Lawitzky, *Field test of a navigation system: Autonomous cleaning in supermarkets,* Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1998.

[13] A. Foka and P. Trahanias, *Predictive autonomous robot navigation*, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[14] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, *A probabilistic approach to collaborative multi-robot localization*, Autonomous Robots **8(3)** (2000).

[15] U. Frese and G. Hirzinger, *Simultaneous localization and mapping - a discussion*, IJCAI01 Workshop Reasoning with Uncertainty in Robotics (Seatle, Washington, USA), August 2001.

[16] K. Goldberg, S. Gentner, C. Sutter, J. Wiegley, and B. Farzin, *The mercury project: A feasibility study for online robots*, Beyond Webcams: An Introduction to Online Robots (K. Goldberg and R. Siegwart, eds.), MIT Press, 2002.

[17] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger, *The telegarden*, Proc. of ACM SIGGRAPH, 1995.

[18] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, *A novel approach to nonlinear/non-Gaussian Bayesian state estimation*, IEE Proceedings F **140** (1993), no. 2, 107–113.

[19] J.-S. Gutmann and K. Konolige, *Incremental mapping of large cyclic environments*, Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA), 1999.

[20] D. Hähnel, W. Burgard, and S. Thrun, *Learning compact 3d models of indoor and outdoor environments with a mobile robot*, Fourth European workshop on advanced mobile robots (EUROBOT'01), 2001.

[21] D. Hähnel, D. Schulz, and W. Burgard, *Map building with mobile robots in populated environments*, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2002.

[22] H. Hirukawa, I. Hara, and T. Hori, *Online robots*, Beyond Webcams: An Introduction to Online Robots (K. Goldberg and R. Siegwart, eds.), MIT Press, 2002.

[23] S. King and C. Weiman, *Helpmate autonomous mobile robot navigation system*, Proc. of the SPIE Conference on Mobile Robots, 1990, pp. 190–198.

[24] F. Lu and E. Milios, *Globally consistent range scan alignment for environment mapping*, Autonomous Robots **4** (1997), 333–349.

[25] T.S. Michael and T. Quint, *Sphere of influence graphs in general metric spaces*, Mathematical and Computer Modelling **29** (1994), 45–53.

[26] Hans P. Moravec and A.E. Elfes, *High resolution maps from wide angle sonar*, Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1985, pp. 116–121.

[27] I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, *An affective mobile educator with a full-time job*, Artificial Intelligence **114** (1999), no. 1-2.

[28] http://www.novotech-gmbh.de/.

[29] M.K. Pitt and N. Shephard, *Filtering via simulation: auxiliary particle filters*, Journal of the American Statistical Association **94** (1999), no. 446.

[30] H. Rauch, F. Tung, and C. Striebel, *Maximum likelihood estimates of linear dynamic systems*, American Institute of Aeronautics and Astronautics Journal **3** (1965), no. 8, 1445–1450.

[31] D. Rodriguez-Losada, F. Matia, R. Galan, and A. Jimenez, *Blacky, an interactive mobile robot at a trade fair*, Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2002.

[32] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, *Tracking multiple moving objects with a mobile robot*, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[33] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers, *Web interfaces for mobile robots in public places*, IEEE-Magazine on Robotics and Automation (2000).

[34] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan, *A layered architecture for office delivery robots*, Proc. of the First International Conference on Autonomous Agents (Agents), 1997.

[35] R. Smith, M. Self, and P. Cheeseman, *Estimating uncertain spatial relationships in robotics.*, Autonomous Robot Vehicles. (I. J. Cox and G. T. Wilfong, eds.), Springer-Verlag, 1990, pp. 167–193.

[36] K. Taylor and J. Trevelyan, *A telerobot on the World Wide Web*, Proceedings of the 1995 National Conference of the Australian Robot Association, 1995.

[37] S. Thrun, *A probabilistic online mapping algorithm for teams of mobile robots*, Journal of Robotics Research **20** (2001), no. 5, 335–363.

[38] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, *Probabilistic algorithms and the interactive museum tour-guide robot Minerva*, Journal of Robotics Research **19** (2000), no. 11.

[39] S. Thrun, W. Burgard, and D. Fox, *A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping*, Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2000.

[40] http://www.ics.forth.gr/tourbot/.

[41] C.-C. Wang and C. Thorpe, *Simultaneous localization and mapping with detection and tracking of moving objects*, Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 2002.

[42] http://www.webcam32.com/.

[43] http://www.ics.forth.gr/webfair/.