MINERVA: A Second-Generation Museum Tour-Guide Robot

Sebastian Thrun[1], Maren Bennewitz[2], Wolfram Burgard[2], Armin B. Cremers[2], Frank Dellaert[1], Dieter Fox[1]
Dirk Hähnel[2], Charles Rosenberg[1], Nicholas Roy[1], Jamieson Schulte[1], Dirk Schulz[2]

[1]School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213

[2]Computer Science Department III
University of Bonn
Römerstr. 164
53117 Bonn, Germany

Abstract

This paper describes an interactive tour-guide robot, which was successfully exhibited in a Smithsonian museum. During its two weeks of operation, the robot interacted with more than 50,000 people, traversing more than 44km. Our approach specifically addresses issues such as safe navigation in unmodified and dynamic environments, and short-term human-robot interaction.

1 Introduction

This article describes Minerva, a mobile robot designed to educate and entertain people in public places. The robot's purpose is to guide people through a museum, explaining what they see along the way. The robot was recently installed successfully in the Smithsonian's National Museum of American History, in an exhibition hosted by its Lemelson Center for Invention and Innovation. During a two-week installation period in the summer of 1998, the robot successfully educated (and entertained) tens of thousands of people.

Minerva is controlled by a fairly generic software approach for robot navigation and human robot interaction, which addresses the following problems:

- **Navigation in dynamic environments.** Public places are often packed with people. People behave not necessarily cooperatively (many try to "break" the system). Our approach provides means for safe and effective navigation through crowds.
- **Navigation in unmodified environments.** No modification of the environment is necessary for the robot's operation. Instead the our software enables robots to adapt to their environments.
- **Short-term human-robot interaction.** Our software is specially designed to interact with people—or crowds of people—who have not been exposed to robots before. To appeal to people's intuition, the robot's interface uses patterns of interactions similar to those found when people interact with each other.
- **Virtual tele-presence.** A Web interface enables people around the world to monitor the robot, control its movement, and watch images recorded by the robot's cameras.

This article describes the software architecture, reports results obtained in the museum, and compares our approach to Rhino, the world's first museum tour-guide robot, developed by the same team and deployed in mid-1997 [4].

2 General Software Architecture

Minerva's software architecture consists of approximately 20 distributed modules which communicate asynchronously using TCX [7]. These modules can broadly be classified into four groups: the hardware interface modules, the navigation modules, the interaction modules, and the Web interface. At the lowest level, various modules interface directly to the robot's sensors and effectors (lasers, sonars, cameras, motors, pan/tilt unit, face, speech unit, touch-sensitive display, Internet server, etc.). On top of that, various navigation modules performed functions like mapping, localization, collision avoidance, path planning, and global mission planning. The interaction modules determine the "emotional state" of the robot, control its head direction, and determine what it says (speech, sounds). Finally, the Web interface consists of modules concerned with displaying information such as images and the robot's position on the Web, and with receiving Web user commands.

Among the various design principles involved in the software design, three stick out as the most important ones:

1. **Probabilistic reasoning.** Minerva uses probabilistic methods to model and reason about its environment. For example, instead of determining the exact pose of the robot (the term *pose* refers to the robot's x-y location along with its heading direction), Minerva maintains a

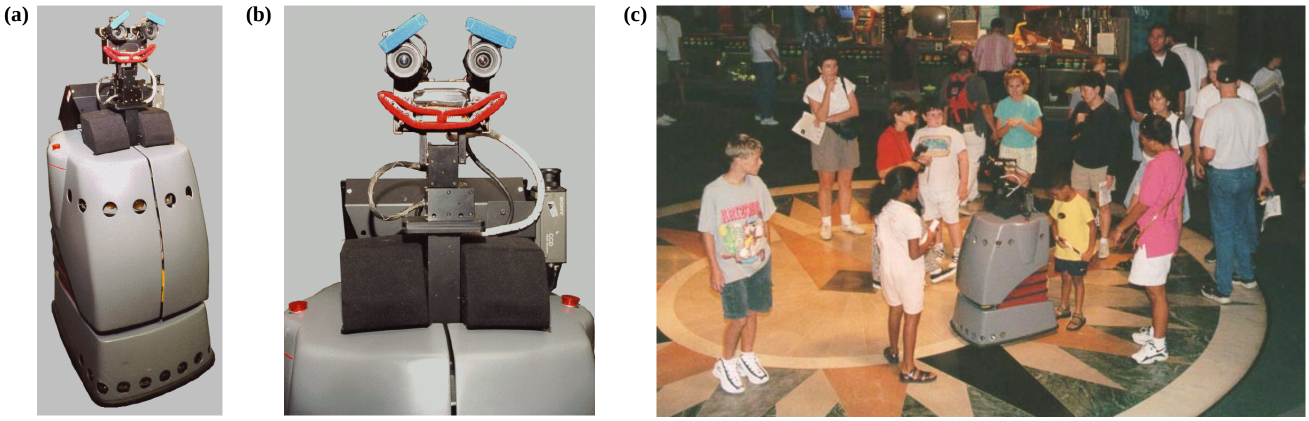

Figure 1: (a) Minerva. (b) Minerva's motorized face. (c) Minerva gives a tour in the Smithsonian's National Museum of American History.

probability distributions over poses, denoted $Pr(x, y, \theta)$. Probabilistic representations are essential for the robustness of the approach.

2. **Learning.** Learning plays a key role in Minerva. For example, the robot learns maps of its environment, sensor models, and it also learns models of travel times that affect the on-line composition of tours.
3. **Any-time algorithms.** Minerva's computationally demanding modules (such as the localization and the path planner) are implemented as *any-time* algorithms. Such algorithms return an answer at any time, whenever needed; however, the answer might not be optimal. As a result, people never have to wait for a computation to terminate.

The remainder of this paper describes Minerva's major software modules.

3 Mapping

Minerva uses two types of maps to orient itself: Occupancy maps [22, 6, 25], as shown in Figure 2, and texture maps of the museum's ceiling, as shown in Figure 3. Both maps are learned from sensor data. Sensor data—laser scans, camera images and odometry readings—are collected when manually joy-sticking the robot through its environment. The problem of building maps is usually referred to as "concurrent mapping and localization" [3, 17], as errors in odometry have to be corrected when building a map.

3.1 Occupancy Map

The occupancy map is learned first. To accommodate errors in odometry, which can easily grow as large as tens of meters, the approach described in [26] is employed. This approach phrases the concurrent mapping and localization problem as a *maximum likelihood estimation problem*, in which one seeks to determine the the most likely map given the data.

$$Pr(m|d) \tag{1}$$

Here m denotes the map, and d the data. The likelihood $Pr(m|d)$ takes into account the consistency of the odometry (small odometry errors are more likely than large ones), and it also considers the perceptual consistency (inconsistencies in perception are penalized).[1] As shown in [26], the likelihood function can be re-expressed as

$$\begin{aligned} Pr(m|d) \quad = \quad & \alpha \int \cdots \int \prod_{t=0}^{T} Pr(s^{(t)}|m, \xi^{(t)}) \qquad (2) \\ & \prod_{t=0}^{T-1} Pr(\xi^{(t+1)}|u^{(t)}, \xi^{(t)}) \; d\xi^{(0)}, \ldots, d\xi^{(T)}. \end{aligned}$$

where $\xi^{(t)}$ denotes the robot's pose at time t, $s^{(t)}$ denotes an observation (laser, camera) and $u^{(t)}$ an odometry reading. This expression can be maximized efficiently using the EM algorithm, as described in [26]; as a side effect, the estimation routine "guesses" the errors in the robot's odometry, and produces the most likely path that is probabilistically consistent with the map. The resulting map, for the museum, is shown in Figure 2. This map is approximately 67 by 53 meter in size. See [26] for more details.

3.2 Texture Maps of the Ceiling

In our previous work, we exclusively relied on occupancy maps for navigation. The sheer size and density of people

[1] For completeness, we note that Minerva used a simplified version of the method described in [26], where the "backwards phase" was omitted and maximum likelihood estimates of the robot's path were used for mapping instead of entire distribution.

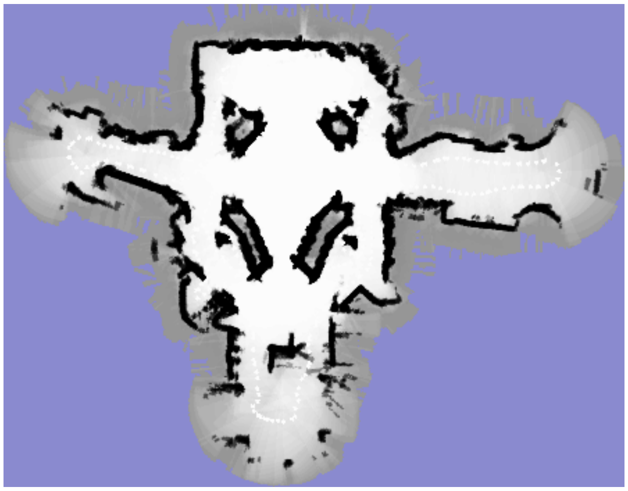

Figure 2: Occupancy map of the center portion of the Smithsonian museum.

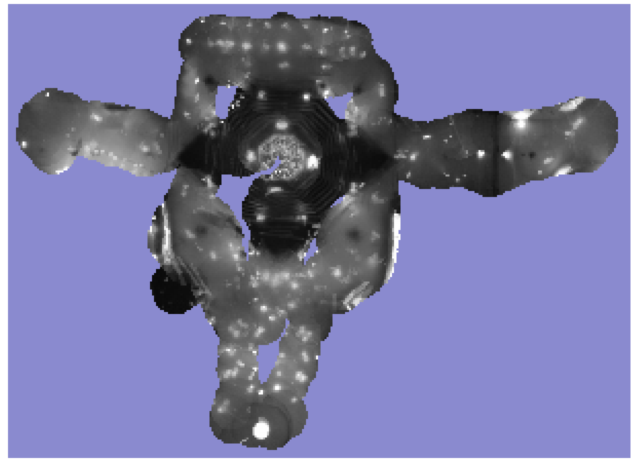

Figure 3: Mosaic of the museum's ceiling. The various bright spots correspond to lights. The center portion of the ceiling contains an opening—the lights there are approximately 15 meters higher.

in the present museum, however, forced us to augment our approach using a camera pointed at the ceiling. The ceiling map is a large-scale mosaic of a ceiling's texture. Such ceiling mosaics are more difficult to generate than occupancy maps. This is because the height of the ceiling is unknown, which makes it difficult to translate coordinates in the image plane into real-world coordinates.

A typical ceiling mosaic is shown in Figure 3. Our approach uses the (previously learned) occupancy map to adjust errors in the odometry. While those poses are not (yet) accurate to the precision that can be attained using the high-resolution vision sensors, they eliminate the difficult to solve global alignment problem. The likelihood $Pr(m|d)$ of the ceiling map is then maximized by searching in the space the following parameters: the pose ξ at which each image was taken, the height of ceiling segments, and two additional parameters per image specifying variations in lighting conditions. Our approach employs the well-known Levenberg-Marquardt algorithm for optimization. As a result, the images are brought into local alignment, the ceiling height is estimated, and a global mosaic is constructed. Figure 3 shows the ceiling mosaic used in the museum. A typical run for an environment of its size involves optimizing over about 3000 unknowns, which requires approximately 30 minutes of processing time on a state-of-the-art computer.

4 Localization

In everyday operation, Minerva continuously tracks its position using its maps. Position estimates are necessary for the robot to know where to move when navigating to a specific exhibit, and to ensure the robot does not accidentally leave its operational area. A key complication arises from the fact that during everyday operation, people frequently obstruct the robot's sensors. This applies to both sensor systems used for localization: The laser scanners, which are blocked by people's legs, and the ceiling cameras, which people often block intentionally, to confuse the robot.

Minerva employs a modified version of Markov localization [5, 15, 24]. Markov localization maintains a probability distribution over all possible poses, denoted $Pr(\xi)$. It uses Bayes rule to incorporate sensor readings

$$Pr(\xi^{(t)}|s^{(t)}) \quad = \quad \frac{Pr(s^{(t)}|\xi^{(t)})\ Pr(\xi^{(t)})}{Pr(s^{(t)})}, \qquad (3)$$

and it uses convolution to incorporate robot motion

$$Pr(\xi^{(t+1)}) \;=\; \int Pr(\xi^{(t+1)}|u^{(t)}\xi^{(t)})\ Pr(\xi^{(t)})\ d\xi^{(t)} \qquad (4)$$

As shown in [5], our implementation computes both terms in an any-time fashion. Figure 4 illustrates how Minerva localizes itself from scratch (global localization). Initially, the robot does now know its pose; thus, $Pr(\xi^{(0)})$ is distributed *uniformly*. After incorporating one sensor reading (laser and camera) according to the update rule (3), $Pr(\xi^{(1)})$ is distributed as shown in Figure 4a. While this distribution is multi-modal, high likelihood is already placed near the correct pose. After moving forward, which is modeled using (4), and after incorporating another sensor reading, the final distribution $Pr(\xi^{(2)})$ is centered around the correct pose, as shown in Figure 4b.

Unfortunately, Markov localization assumes the environment is *static*. Populated environments, such as the museum, are highly dynamic. The key idea to accommodate dynamics is to *filter* the sensor readings. More specifically,

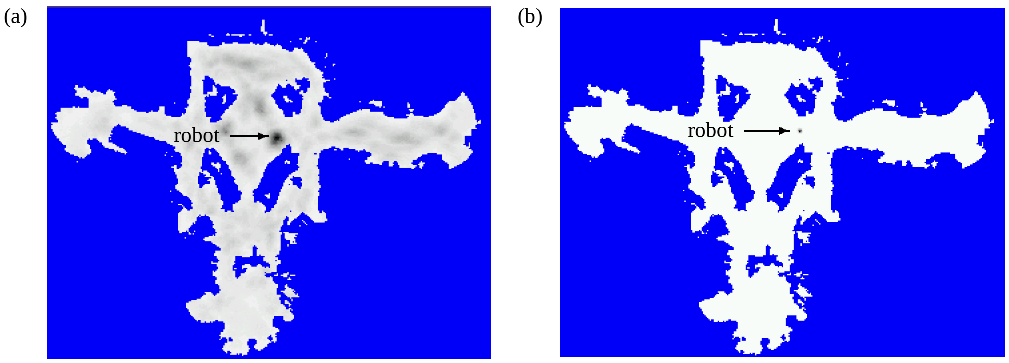

Figure 4: Global localization: (a) Pose probability $Pr(\xi)$ distribution after integrating a first laser scan (projected into 2D). The darker a pose, the more likely it is. (b) $Pr(\xi)$ after integrating a second sensor scan. Now the robot knows its pose with high certainty/accuracy.

the robot sorts sensor readings into two buckets, one which contains readings that are assumed to be corrupted by people, and one which are assumed to correspond to static obstacles in the map (called: authentic). Figure 5 shows an example of a laser scan. Both figures are generated from a single laser scan; Authentic readings are shown in Figure 5a, and corrupted ones in Figure 5b. This example is prototypical and demonstrates the high accuracy in the identification of corrupted readings.

The filter is described in detail in [11] (called there *novelty filter*). In essence, the robot computes the *expected* sensor reading

$$E_{P(o|\xi,m)}[o] \quad := \quad = \int dist(\xi)\ Pr(\xi)\ xi \qquad (5)$$

where $dist(\xi)$ denotes the exact measurement one would expect when the robot's pose is ξ, which is computed using ray-tracing. $Pr(\xi)$ is the robot's current belief. Proximity measurements are assumed to be corrupted if they are *shorter* than the expected value $E_{P(o|\xi,m)}[o]$; otherwise, they are labeled authentic. This filter, which is analyzed in detail in [11], proved extremely efficient in sorting out "corrupted" sensor readings. It is essential for localization in dynamic environment.

5 Collision Avoidance

Minerva's collision avoidance module controls the momentary motion direction and velocity of the robot so as to avoid collisions with obstacles—people and exhibits. Most mobile robot collision avoidance methods consider only the kinematics of a robot; they do not take dynamics into account. This is legitimate at speeds where robots can stop almost instantaneously. At velocities of up to 163 cm/sec, which was Minerva's maximum speed when under exclusive Web control, inertia and torque limits impose constraints on robot motion which may not be neglected. Hence, it is essential to consider the robot's dynamics.

Minerva's collision avoidance method, called μDWA is described in depth in [9, 10]. Here we will only sketch it. In essence, the input to μDWA are raw proximity sensor readings along with a desired target location, based on which μDWA sets the robot's velocity (translation and rotation). It does this by obeying a collection of constraints, which come in two flavors: hard and soft. Hard constraints establish the basic safety of the robot (e.g., the robot must always be able to come to a full stop before impact) and they also express dynamic constrains (e.g., torque limits). Soft constraints are used to trade off the robot's desire to move towards the goal location, and its desire to move away from obstacles into open space. In combination, these constraints ensure safe and smooth local navigation.

The μDWA algorithm was originally designed for circular robots with synchro-drive. Minerva, however, possesses a non-holonomic differential drive, and the basic shape resembles that of a rectangle. Collision avoidance with rectangular robots is generally regarded more difficult. However, μDWA could easily be extended to robots of this shape by adapting the basic geometric model. The approach was able to safely steer the robot at speeds of 1.6m/sec, which is twice as high as that of any robot previously used in our research. This suggests that the μDWA approach applies to a much broader class of robots than previously reported.

6 Path Planning

The path planner computes paths from one exhibit to another. The problem of path planning for mobile robots has been solved using a variety of different methods [19]. Previous path planners, however, do not take into account the danger of getting lost; instead, they minimize path length. In wide, open environments, the choice of the path influences the robot's ability to track its position. In particular, locations like the wide open center portion of the museum lack the necessary reference points to maintain accurate localization. To minimize the chances of getting lost, it is therefore important to take uncertainty into account when planning paths.

Our idea is simple but effective: In analogy to ships,

	average	min	max
static	398 ± 204 sec	121 sec	925 sec
with learning	384 ± 38 sec	321 sec	462 sec

Table 1: This table summarizes the time spent on individual tours. In the first row, tours were pre-composed by static sequences of exhibits; in the second row, tours were composed on-the-fly, based on a learned model of travel time, successfully reducing the variance by a factor of 5.

(a) "authentic" readings **(b)** "corrupted" readings

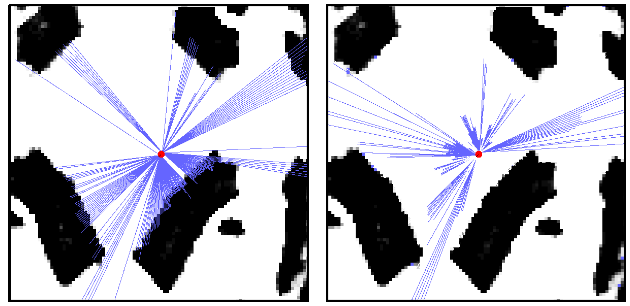

Figure 5: Filtering out corrupted sensor data in dynamic environments: (a) Authentic readings only; (b) readings corrupted by people. Only authentic readings are used for localization. Corrupted readings are employed to find people, e.g., when determining the robot's mood.

which typically stay close to coasts to avoid getting lost (unless they are equipped with a global positioning system), Minerva's path planner is called a *coastal planner*. In essence, paths are generated according to a mixture of two criteria: path length and *information content*. The latter measure, information content, reflects the amount of information a robot is expected to receive at different locations in the environment. A typical map of information content is shown in Figure 6a. Here the grey scale indicates information content: the darker a location, the less informative it is. This figure illustrates that the information content is smallest in the center area of the museum.

Formally, information content is defined as the *expected reduction in entropy upon sensing*, i.e.,

$$H[Pr(\xi)] - \int Pr(\xi)\; E[s|\xi]\; H[Pr(\xi'|s)]\; d\xi. \qquad (6)$$

Here $E[s|\xi]$ denotes the expected sensor reading at pose ξ. When constructing the map shown in Figure 6a, this expression is computed off-line for every location, making the assumption that the robot knows its position within a small, Gaussian-distributed uncertainty margin. Our approach also exploits the fact that the robot's sensors cover a $360°$ field of view, which allows us to ignore the orientation θ when considering information content. When computing (6), the presence of people is taken into account by modeling noise in sensing (assuming 500 randomly positioned people in the museum).

As described above, paths are generated by simultaneously minimizing path length and maximizing information content, using dynamic programming [14]. A typical result is shown in Figure 6b. Here the path (in white) avoids the center region of the museum, even though the shortest path would lead straight through this area. Instead, it generates a path that makes the robot stay close to obstacles, where chances of getting lost are much smaller. In comparative tests, we found that this planner improved the certainty in localization by a factor of three, when compared to the shortest-path approach.

7 High-Level Control

It was generally desirable for tours to last approximately six minutes—which was determined to be the duration the average visitor would like to follow the robot. Unfortunately, in practice the rate of progress depends crucially on the number and the behavior of the surrounding people. Thus, the duration of tours can vary widely if the the exhibits are preselected. To meet the target duration as closely as possible, tours are composed dynamically, based on the crowdedness in the museum.

To address this problem, Minerva uses a flexible high-level interface, capable of composing tours on-the-fly. This interface learns the time required for moving between pairs of exhibits, based on data recorded in the past (using maximum likelihood estimators). After an exhibit is explained, the interface chooses the next exhibit based on the remaining time. If the remaining time is below a threshold, the tour is terminated and Minerva instead returns to the center portion of the museum. Otherwise, it selects exhibits whose sequence best fit the desired time constraint. The learning algorithm (maximum likelihood estimator) and the decision algorithm were implemented in RPL, a language for reactive planning [20], which is interfaced to the lower-level control structures using HLI, a component of GOLEX [12].

Table 1 illustrates the effect of dynamic tour decomposition on the duration of tours. Minerva's environment contained 23 designated exhibits, and there were 77 sensible pairwise combinations between them (certain combinations were invalid since they did not fit together thematicly). In the first days of the exhibition, all tours were static. The first row in Table 1 illustrates that the timing of those tours varies significantly (by an average of 204 seconds). The average travel time was estimated using 1,016 examples, collected during the first days of the project. The second row in Table 1 shows the results when tours were composed dynamically. Here the variance of the duration of a tour is only 38 seconds.

The high-level interface also made the robot return to its charger periodically, so that we could hot-swap its batteries.

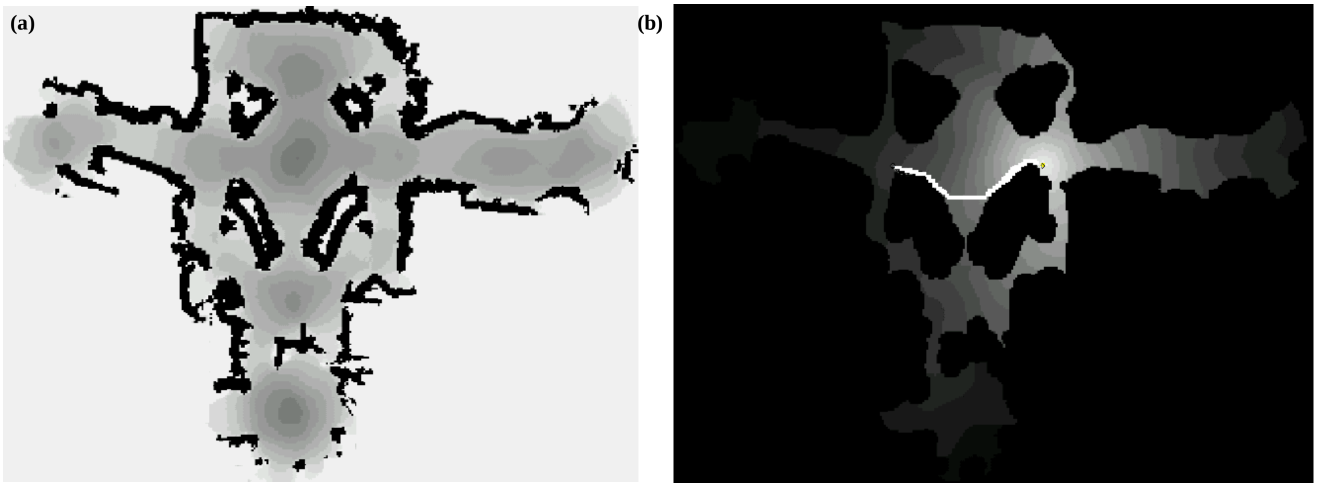

Figure 6: Coastal navigation: The entropy map, shown in (a), characterizes the information loss across different locations in the unoccupied space. The darker an area, the less informative it is. The information loss is largest in the center of the museum, far away from any obstacle. (b) Path generated by the planner, taking both information loss and distance into account. Here Minerva avoids the center area of the museum.

8 Spontaneous Short-Term Interaction

Interaction with people was Minerva's primary purpose. The type of interaction was characterized by several factors:

- Visitors of the museum typically had no prior exposure to robotics technology, and many of them did not intend to interact with the robot when visiting the museum. Visitors could not be "instructed" beforehand as to how to operate the robot.
- The robot often had to interact with crowds of people, not just with single individuals.
- Most people spent less then 15 minutes (even though some spend hours, or even days).

This type of interaction is characteristic for robots that operate in public places (such as information kiosks, receptionists). It differs significantly from the majority of interactive modes studies in the field (e.g., [1, 16, 18]), which typically assumes long-term interaction with people one-on-one. The robot's interaction was driven by two goals: to attract people so that they would join a tour, and to clear its path so that it could make maximum progress when giving tours.

When giving tours, Minerva uses its face, its head direction, and its voice to maximize its progress. A stochastic finite state automaton (with 4 states) is employed to model simple "emotional states" (moods), which allowed the robot to communicate its intent to visitors in a social context with which they were already familiar [8, 23]. Moods ranged from happy to angry, depending on the persistence of the people who blocked its path. When happy, Minerva smiled and politely asked for people to step out of the way; when angry, it's face frowned and her voice sounded angry. Most museum visitors had no difficulty understanding the robot's intention and "emotional state."

To attract people, Minerva used a memory-based reinforcement learning approach [2, 21]. Reinforcement was received in proportion of the proximity of people; coming too close, however, led to a penalty (violating Minerva's space). Minerva's behavior was conditioned on the current density of people. Possible actions included different strategies for head motion (e.g., looking at nearest person), different facial expressions (e.g., happy, sad, angry), and different speech acts (e.g., "Come over," "do you like robots?"). Learning occurred during one-minute-long "mingling phases" that took place between tours.

We found that acts best associated with a "positive" attitude attracted the most people. For example, when grouping speech acts and facial expressions into two categories, friendly and unfriendly, we found that the former type of interaction performed significantly better than the first (with 95% confidence). However, these results have to be interpreted with care, as people's response was highly stochastic and the amount of data we were able to collect during the exhibition is insufficient to yield statistical significance in most cases.

9 Web Interface

One of the goals of the project was to enable remote to establish a "virtual tele-presence" in the museum, using the Internet. Therefore, Minerva was connected to the Web[2], where Web users all over the world controlled Minerva and could "look through its eyes."

During opening hours, Minerva was controlled predominately by the visitors of the museum, which could select

[2] See `http://www.cs.cmu.edu/~minerva`.

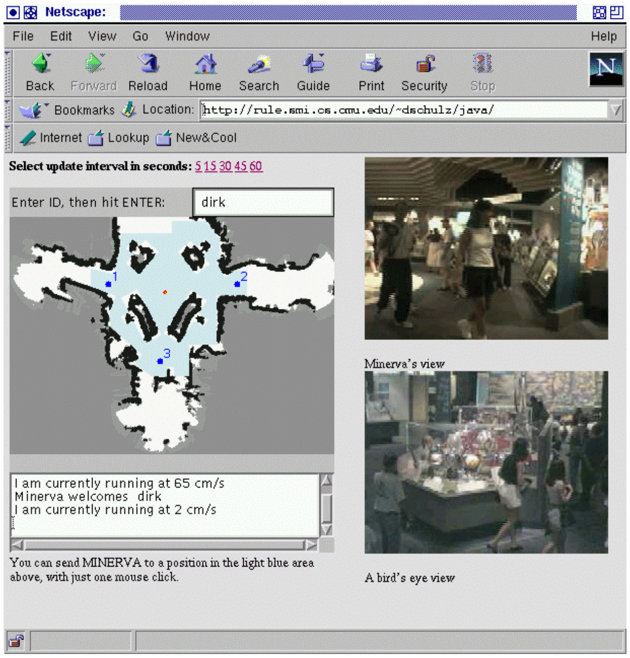

Figure 7: Web control interface. Users can log in on the left side of the window, and specify target locations by clicking in the map. The map shows current robot position, pending target locations, and a dialogue box displays the current speed of the robot. On the right. users can watch images recorded using the robot's camera (top image) and by a stationary camera mounted on a pan/tilt unit (bottom image).

tours using a touch-sensitive screen mounted at Minerva's back. Every third tour, however, was selected by Web users, using a voting scheme: Votes for individual tours were counted, and the most popular tour was chosen. At all times, Web users could watch camera images recorded by Minerva and an off-board, stationary camera (mounted on a pan/tilt unit and equipped with a zoom), and they could also see the robot's location displayed in the map. To facilitate updating the position of Minerva several times a second, Web users downloaded a robot simulator written in Java, and used TCP communication and server-push technology to communicate the position of the robot in approximately real time.

During several special scheduled Internet events, all of which took place when the museum was closed to visitors, Web users were given exclusive control of the robot. Using the interface shown in Figure 7, they could schedule target points, which the robot approached in the order received. The number of pending target points was limited to five. All rows in Table 2 marked "Web only" correspond to times where Web users assumed exclusive control over the robot. In one case, Minerva moved at an average velocity of 73.8 cm/sec. Its maximum velocity was 163 cm/sec, which was attained frequently.

date	uptime	travel time	distance	avg. speed	tours	exhibits	mode
Aug 24	7:16:08	2:34:36	2,881.13m	31.3cm/sec	52	174	
Aug 25	7:41:52	2:17:05	2,725.90m	33.1cm/sec	55	169	
Aug 26	6:57:35	2:39:24	2,642.23m	27.6cm/sec	28	102	
Aug 27	5:40:58	1:33:00	1,147.12m	31.7cm/sec	53	203	
	1:56:21	0:50:55	1,755.98m	57.5cm/sec	28	104	Web only
Aug 28	6:48:59	2:08:14	2,416.15m	31.4cm/sec	54	192	
Aug 29	5:40:23	1:50:22	2,436.92m	36.7cm/sec	59	219	
Aug 30	6:42:36	2:17:58	3,305.44m	39.9cm/sec	66	231	
Aug 31	7:25:57	2:09:02	3,372.91m	43.6cm/sec	77	258	
Sept 1	7:11:54	2:22:40	3,707.19m	43.3cm/sec	61	230	
Sept 2	4:28:07	1:27:33	1,954.19m	37.2cm/sec	37	137	
Sept 3	9:56:53	3:25:08	5,332.76m	43.3cm/sec	54	203	
Sept 4	1:13:15	0:52:34	2,143.86m	68.0cm/sec		103	Web only
	6:49:35	2:04:49	2,611.71m	34.9cm/sec	48	168	
	2:17:04	1:17:00	3,411.41m	73.8cm/sec		175	Web only
Sept 5	6:15:46	1:42:34	2,173.90m	35.3cm/sec	49	156	
total	94:23:20	31:32:54	44,018.8m	38.8cm/sec	620	2,668	

Table 2: Summary statistics of Minerva's operation. The rows labeled "Web only" indicate times when the museum was closed to the public, and Minerva was under exclusive Web control. At all other times, Web users and museum visitors alternated the control of the robot. Minerva's top speed was 163 cm/sec.

10 Comparison with Rhino

Minerva is the second generation of museum tour-guide robots. The first museum tour-guide robot was called Rhino, and we installed it in mid-1997 in the Deutsches Museum Bonn (Germany) [4]. Rhino's success motivated the creation of another tour-guide robot, called Chips (or Sage), which was recently developed by a different team of researchers [23]. The latter robot uses optical markers to facilitate navigation.

Navigation in the Smithonian museum posed completely new challenges that were not present in the Deutsches Museum Bonn. Minerva's environment was an order of magnitude larger, with a particular challenge arising from the large open area in the center portion of the museum. Minerva also had to cope with an order of magnitude more people ($>$50,000) than Rhino ($\sim$2,000).

To accommodate these difficulties, Minerva's navigation system was more sophisticated. In particular, Rhino did not use camera images for localization, and its motion planner did not consider information gain when planning paths. In addition, Rhino was supplied with a manually derived map; it lacked the ability to learn maps from scratch. We believe that these extensions were essential for Minerva's success.

A key difference between both robots relates to their interactive capabilities. As mentioned above, Rhino's interaction was more rudimentary. It lacked a face, did not exhibit "emotional states," and it did not actively attract or engage people. As a result, Minerva was much more effective in attracting people and making progress. When compared to the Rhino project, we consistently observed that people cleared the robot's path much faster. We found that both robots maintained about the same average speed (Minerva: 38.8 cm/sec, Rhino: 33.8 cm/sec), despite the fact that Minerva's environment was an order of magnitude more crowded. These numbers illustrate the effectiveness of Minverva's interactive approach to making progress.

In comparison with Rhino, people also appeared more satisfied and amused. When asked what level of animal (from a list of five options) Minerva's intelligence was most comparable to, we received the following answers: human: 36.9%; monkey: 25.4%; dog: 29.5%; fish: 5.7%; amoeba:2.5%. 27.0% of all people (predominately kids of 10 years of age or less) believed Minerva was "alive," whereas 69.8% thought it was not (3.2% were undecided). A total of 63 people were asked (36 male, 27 female).

Minerva also possesses an improved Web interface, which enabled Web users to specify arbitrary target locations instead of choosing locations from a small pool of pre-specified locations. Rhino's Web interface prescribes a small set of 13 possible target locations, which corresponded to designated target exhibits. When under exclusive Web control, Minerva was twice as fast as Rhino (see Table 2). In everyday operation, the maximum speed of both robots was limited to walking speed (70 cm/sec).

11 Conclusion

This article described the software architecture of a mobile tour-guide robot, which was successfully exhibited for a limited time period at the Smithsonian's National Museum of American History. Our approach contains a collection of new ideas, addressing challenges arising from the size and dynamics of the environment, and from the need to interact with crowds of people. The empirical results of the exhibition indicate a high level of robustness and effectiveness. Future research issues include the integration of speech recognition, to further develop the robot's interactive capabilities.

Acknowledgments

We are deeply indebted to the Lemelson Center of the National Museum of American History, for their superb support of this project. We also thank Anne Watzman for managing our relation to the media so well, and Greg Armstrong for his excellent hardware support. Special thanks also to Real World Interface for excellent hardware support, without which this project would not have been possible.

This research is sponsored in part by DARPA via AFMSC (contract number F04701-97-C-0022), TACOM (contract number DAAE07-98-C-L032), and Rome Labs (contract number F30602-98-2-0137). Additional financial support was received from Daimler Benz Research and Andy Rubin, all of which is gratefully acknowledged.

References

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

[2] C. A. Atkeson. Using locally weighted regression for robot learning. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 958–962, Sacramento, CA, April 1991.

[3] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[4] W. Burgard, A.B., Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[5] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Menlo Park, August 1996. AAAI, AAAI Press/MIT Press.

[6] A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

[7] C. Fedor. TCX. An interprocess communication system for building robotic architectures. programmer's guide to version 10.xx. Carnegie Mellon University, Pittsburgh, PA 15213, December 1993.

[8] C. Breazeal (Ferrell). A motivational system for regulating human-robot interaction. In *Proceedings of AAAI'98*, pages 54–61, Madison, WI, 1998.

[9] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation*, 4(1), 1997.

[10] D. Fox, W. Burgard, and S. Thrun. A hybrid collision avoidance method for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

[11] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.

[12] D. Haehnel, W. Burgard, and G. Lakemeyer. GOLEX: Bridging the gap between logic (GOLOG) and a real robot. In *Proceedings of the 22st German Conference on Artificial Intelligence (KI 98)*, Bremen, Germany, 1998.

[13] I. Horswill. Specialization of perceptual processes. Technical Report AI TR-1511, MIT, AI Lab, Cambridge, MA, September 1994.

[14] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press and Wiley, 1960.

[15] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[16] R.E. Kahn, M.J. Swain, P.N. Prokopowicz, and R.J. Firby. Gesture recognition using the perseus architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 734–741, San Francisco, CA, 1996.

[17] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, Cambridge, MA, 1998. MIT Press.

[18] D. Kortenkamp, E. Huber, and P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of AAAI-96*, pages 915–921. AAAI Press/The MIT Press, 1996.

[19] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[20] Drew McDermott. The RPL manual. Can be obtained from `http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/mcdermott.html`, 1993.

[21] A. W. Moore and C. G. Atkeson. Memory-based function approximators for learning control. MIT AI-Lab Memo, July 1992.

[22] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.

[23] I.R. Nourbakhsh. The failures of a self-reliant tour robot with no planner. Can be obtained at `http://www.cs.cmu.edu/~illah/SAGE/index.html`, 1998.

[24] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI-95*, pages 1080–1087, Montreal, Canada, August 1995. IJCAI, Inc.

[25] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[26] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. also appeared in Autonomous Robots 5, 253–271.