MINERVA: A Tour-Guide Robot That Learns

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
Pittsburgh, PA 15213

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

Abstract. This paper describes an interactive tour-guide robot which was successfully exhibited in a Smithsonian museum. Minerva employed a collection of learning techniques, some of which were necessary to cope with the challenges arising from its extremely large and crowded environment, whereas others were used to aid the robot's interactive capabilities. During two weeks of highly successful operation, the robot interacted with thousands of people, traversing more than 44km at speeds of up to 163 cm/sec in the un-modified museum.

1 Introduction

This paper presents Minerva, the latest in a series of mobile tour-guide robots. The idea of a tour-guide robot goes back to Horswill [8], who implemented a vision-based tour-guide robot in the AI lab of MIT. More recently, Burgard and colleagues developed a more sophisticated tour-guide robot called *Rhino*, which that successfully installed in the Deutsches Museum Bonn [1]. Inspired by this, Norbakhsh and colleagues developed a similar robot called *Chips* for the Carnegie Museum of Natural History [14].

Building on these successes, the Minerva project pursued two primary goals:

1. **Scaling up:** Minerva's environment was an order of magnitude larger and more crowded than previous museum environments. The robot's top speed of 163 cm/sec was more than double than that of previous robots. An additional scaling goal was a reduction in lead-time for installing such robots (from several months to only three weeks). As in the Rhino project, the environment was not modified in any way to facilitate the robot's operation.
2. **Improved user interaction.** To aid the robot's interactive capabilities, Minerva was designed to exhibit a life-like character, strongly relying on common tokens in inter-human interaction. For example, Minerva possessed a moving head with motorized face and a voice. The robot's behavior was a function of its "emotional state," which adjusted in accordance to people's behavior. These new interactive means were essential for the robot's ability to attract, guide, and educate people—something which was recognized as a clear deficiency of previous tour-guide robots.

Fig. 1. Panoramic view of the Material World Exhibition, Minerva's major operation area which is located in the entrance area of the Smithsonian's National Museum of American History (NMAH).

These goals mandated the pervasive use of *learning*, at several levels of the software architecture. Minerva

- employs statistical techniques to acquire geometric maps of the environment's floor plan and its ceiling (other such robots did not learn their map), thereby facilitating the installation process,
- uses the ML estimator to acquire high-level models of time-to-travel, enabling the robot to follow much tighter schedules when giving tours, and
- employs reinforcement learning to acquire skills for attracting people, thereby increasing its overall effectiveness.

In addition, Minerva continuously estimates other important quantities, such as its own location, the location of people, and its battery charge.

During a two-week exhibition that took place in August/September 1998 in one of the world's largest museums, the Smithsonian's National Museum of American History (NMAH), Minerva successfully gave tours to tens of thousands of visitors. The "Material World" exhibition (see Figure 1), where most exhibits were located, was a huge and usually crowded area, which almost entirely lacked features necessary for localization (even the ceiling was uninformative, as the center area had a huge whole with a continuously moving pendulum inside). During the exhibition, more than 100,000 people visited the NMAH, and thousands of people interacted with the robot. The robot successfully traversed 44km, at an average speed of 38.8 cm/sec and a maximum speed of 163 cm/sec. The top speed, however was only attained after closing hours. During public hours, we kept the robot at walking speed, to avoid frightening people.

This article describes major components of Minerva's software. Since we adopted much of the software approach by Burgard and colleagues [1], the focus is here on research that goes beyond this work. In particular, we describe in detail Minerva's learning components, and only highlight other, previously published components .

2 Learning Maps

Previous tour-guide robots were unable to learn a map; instead they relied on humans to provide the necessary information. To facilitate the rapid installation in novel environments, Minerva learns maps of its environments. In particular,

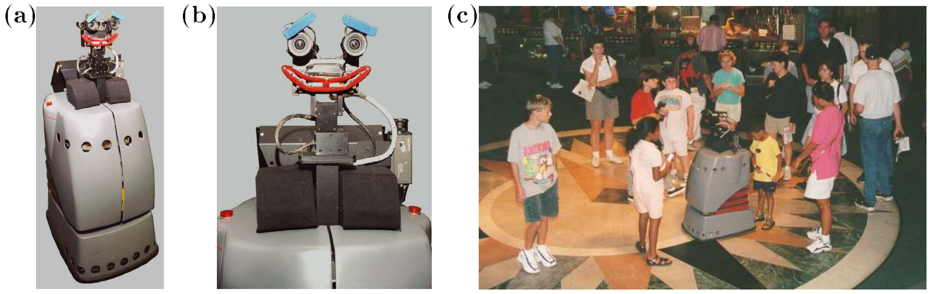

Fig. 2. (a) Minerva. (b) Minerva's motorized face (in happy mode). (c) Minerva gives a tour in the Smithsonian's National Museum of American History.

Minerva learns two types of maps, *Occupancy maps* (see also [3, 17]), and *ceiling texture maps*. The use of a dual map is unique; however, it is necessary in large, open and densely crowded environments such as the museum, where a single sensor modality is often insufficient to track the robot's position reliably.

2.1 Statistical Mapping with EM

Both mapping algorithms are variants of a single, overarching approach originally presented in [18]. The general problem is typically referred to as *concurrent mapping and localization*, which indicates its chicken-and-egg nature: Building maps when the positions at which sensor readings were taken are *known* is relatively simple, and so is determining the robot's positions when the map is *known*; however, if neither the map nor the robot's positions are known, the problem is hard.

The key idea is to treat 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 (odometry and range data/camera images). 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). As shown in [18], the likelihood function can be re-expressed as

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

where $x^{(t)}$ denotes the robot's pose at time t, $s^{(t)}$ denotes an observation (laser, camera) and $u^{(t)}$ an odometry reading.

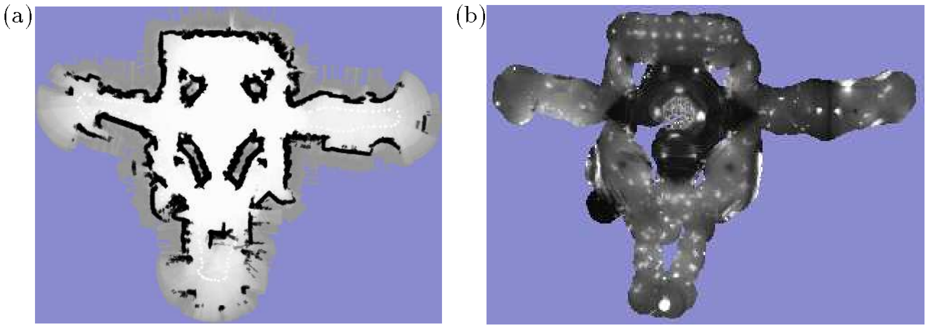

Fig. 3. (a) Occupancy map and (b) ceiling mosaic of oddly-shaped ceiling (center area has a hole and a moving pendulum).

Finding the *global* maximum of $Pr(m|d)$ is computationally infeasible. However, local maxima can be found efficiently using the EM algorithm, a well-known statistical approach to likelihood maximization. This approach interleaves phases of localization (in the E-step) with mapping phases (the M-step). More specifically, in the E-step our approach estimates the various past robot locations denoted x

$$Pr(x|m,d) \tag{3}$$

assuming knowledge of the map m (initially, there is no map and x is exclusively estimated from odometry). The M-step computes the most likely map based on the previously computed x:

$$\underset{m}{\text{argmax}}\ Pr(m|x,d) \tag{4}$$

As argued in [18], iteration of both steps leads to a local maximum in likelihood space. In practice, this approach has been shown to generate maps of unprecedented complexity and size. However, the approach in [18] required that a person put tape on the floor and pushed a button every time the robot traversed a piece of tape—something that we want to avoid since it is an obstacle in rapid, robust installation.

2.2 Occupancy Maps

Our approach deviates from [18] in two aspects: First, we omit the "backward phase" in the E-step (localization), and second, we replace the density $Pr(x|m,d)$ by its maximum likelihood estimate $\text{argmax}_x\ Pr(x|m,d)$. We found that both simplifications have only a marginal effect on the result; however, they greatly reduce the computational and memory requirements. This, in turn, makes it possible to use *raw laser data* for localization and mapping, instead of the "button pushes" that were required in the original approach to reduce the computation.

Simplified speaking, the resulting mapping algorithm performs Markov localization [2, 10, 15] (E-step), interleaved with occupancy grid mapping [3, 17](M-step). Both are easily done in real-time. In practice, we found that ≤ 5 iterations are sufficient, so that the resulting mapping algorithm is fast. A map of the museum is shown in Figure 3a. This map, which was used during the entire exhibition, is approximately 67 by 53 meter in size.

2.3 Texture Maps of the Ceiling

The sheer size, openness, and density of people in the present museum made it necessary to learn a map of the museum's ceiling, using a (mono) B/W camera pointed up. The ceiling map is a large-scale *mosaic* of a ceiling's texture (c.f., Figure 3b). Such ceiling mosaics are difficult to generate, since the *height* of the ceiling is unknown, which makes it difficult to map the image plane to world coordinates.

Just like the occupancy grid algorithm, our approach omits the backwards phase. In addition, we make the restrictive assumption that all location distributions are normal distributed (Kalman filters), which makes the computation extremely fast. Unfortunately, this assumption requires that all uncertainty in the robot's pose is *unimodal*—an assumption that is typically only true when the position error is small at all times (c.f., [12]).

The plain EM approach, as described above, does *not* produce small error. However, a modified version does, for environments of the size of the museum and for sensors as rich in information as cameras. The idea is to interleaves the E-step and the M-step *for each sensor item*, which is much finer grained a level than the approach described above. Whenever a sensor item is processed (new and past data alike), it is first localized and then the map is modified accordingly. A mosaic of the museum's ceiling is shown in Figure 3b. One can clearly see the ceiling lights and other structures of the museum's ceiling (whose height varied drastically).

3 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. Here we adopt Markov localization, as previously described in [2, 10, 15]. Markov localization is a special case of the E-step above.

Figure 4 illustrates how Minerva localizes itself from scratch (global localization). Initially, the robot does now know its pose; thus, $Pr(x|m, d)$ is distributed *uniformly.* After incorporating one sensor reading (laser and camera), $Pr(x|m, d)$ 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 and subsequently incorporating another sensor reading, the final distribution $Pr(x|d, m)$ is centered around the correct pose, as shown in Figure 4b. We also employed

(a) (b)

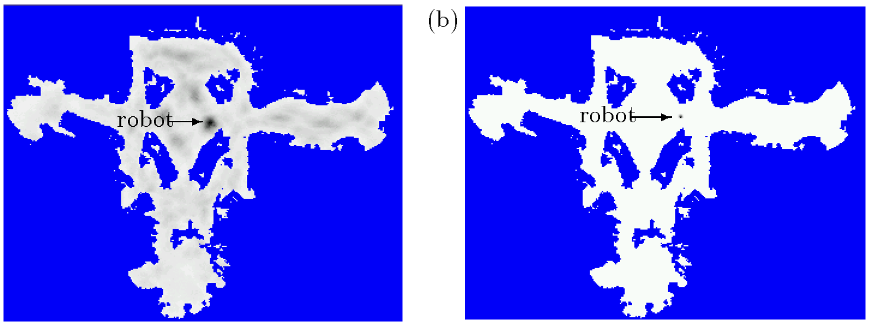

Fig. 4. Global localization: (a) Pose probability $Pr(x)$ distribution after integrating a first laser scan (projected into 2D). The darker a pose, the more likely it is. (b) $Pr(x)$ after integrating a second sensor scan. Now the robot knows its pose with high certainty/accuracy.

a filtering technique described in depth in [7], to accommodate the crowds that frequently blocked the robot's sensors (and thereby violated the Markov assumption that underlies Markov localization).

4 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. At velocities of up to 163 cm/sec, which was Minerva's maximum speed when under exclusive Web control, inertia and torque limits impose severe constraints on robot motion which may *not* be neglected. Hence, we adopted a collision avoidance method called μDWA, originally developed by Fox and colleagues [5, 6]. This approach considers torque limits in collision avoidance, thereby providing safety even at high speeds.

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 1.63 cm/sec, which is more than twice as high as that of any autonomous robot previously used in similar applications. This suggests that the μDWA approach applies to a much broader class of robots than previously reported.

5 "Coastal" Navigation

Minerva's *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

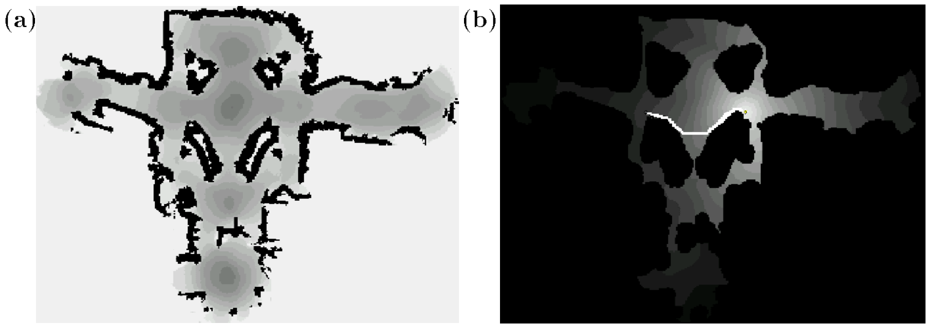

Fig. 5. Coastal navigation: The entropy map, shown in (a), characterizes the information loss across different locations in the unoccupied space. The darker an area, the less informative it is. (b) Path generated by the planner, taking both information loss and distance into account. Minerva avoids the center area of the museum.

methods [11]. Most mobile robot path planners, however, do not take into account the danger of getting lost; instead, they minimize path length (see [?] for an exception). In wide, open environments, the choice of the path influences the robot's ability to track its position. 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, 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 5a. 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.,

$$H[Pr(x)] \; - \int Pr(x) \; E[s|x] \; H[Pr(x'|s)] \; dx. \tag{5}$$

Here $E[s|x]$ denotes the expected sensor reading at pose x. When constructing the map shown in Figure 5a, 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 (5), the presence of people is taken into account by modeling noise in sensing (assuming 500 randomly positioned people in the museum).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		26	68	14	28															
2					23	38	13													
3				81	66	51		66						60						
4																76	22			
5											62								49	
6		41					44													
7			44	1	55			42						51						
8									44	63										
9																				
10																				
11				34								16	69							
12				61	53	69		72	32				87	55						
13												28								
14	33		39																	
15									60											
16																		46		68
17								59						13		57				
18					46	42		31	36					31						12
19				1		25		58			69		12							
20			57	62														37		
21				55	24	20		15						74						
22		208		66	46	38		38	23					56	39					
23				113	76	59		24	46					59						

Table 1. Time (in sec) it takes to move from one exhibit to another, estimated from 1,016 examples collected in the museum. These times, plus the (known) time used for explaining an exhibit, form the basis for the decision-theoretic planner.

As described above, paths are generated by simultaneously minimizing path length and maximizing information content, using dynamic programming [9]. A typical result is shown in Figure 5b. 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.

6 Learning to Compose Tours

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 pre-selected. 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 control module, capable of composing tours on-the-fly. This module *learns* the time required for moving between pairs of exhibits, based on data recorded in the past (using the maximum likelihood estimator). After an exhibit is explained, the interface chooses the next exhibit based on the remaining time. If the remaining time

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

Table 2. 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.

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 [13].

Table 2 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 2 illustrates that the timing of those tours varies significantly (by an average of 204 seconds). The average travel time, shown in Table 1, was estimated using 1,016 examples, collected during the first days of the project. The second row in Table 2 shows the results when tours were composed dynamically. Here the variance of the duration of a tour is only 38 seconds. Minerva's high-level interface also made the robot return to its charger periodically, so that we could hot-swap its batteries.

7 Spontaneous Short-Term Interaction

Interaction with people was Minerva's primary purpose—it is therefor surprising that previous tour-guide robots' interactive capabilities were rather poor. The type of interaction was *spontaneous* and *short-term*: Visitors of the museum typically had no prior exposure to robotics technology, and they 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 studied in the field, which typically assumes long-term interaction with people one-on-one.

To maximize Minerva's effectiveness, we decided to give Minerva "human-like" features, such as a motorized face, a neck, and an extremely simple finite state machine emulating "emotions," and use reinforcement learning to shape her interactive skills.

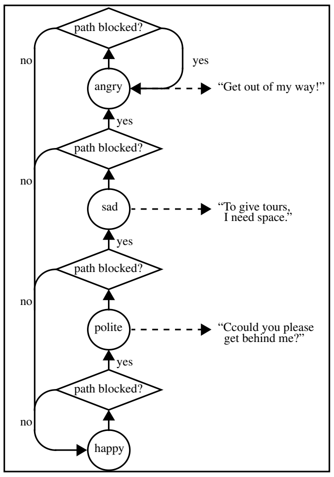

Fig. 6. The finite state automaton that governs Minerva's "emotional" states.

7.1 Emotional States

When giving tours, Minerva used its face, its head direction, and its voice to communicate with people, so as to maximize its progress. A stochastic finite state machine shown in Figure 6 is employed to model simple "emotional states" (moods), which allowed the robot to communicate its intent to visitors in a social context familiar to people from human-human interaction [4]. 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, its face frowned and her voice sounded angry. Most museum visitors had no difficulty understanding the robot's intention and "emotional state." In fact, the ability to exhibit such extremely simplified "emotions" proved one of the most appreciated aspect of the entire project.

The effect of this approach is best evaluated by comparing it with Rhino [1], which uses a similar approach for navigation but mostly lacks these interactive capabilities. We consistently observed that people cleared the robot's path much faster than reported by the Rhino team. 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 shed some light on the effectiveness of Minerva's interactive approach. People clearly "understood' the robot's intentions, and usually cooperated when they observed the robot's "mood" changed.

7.2 Learning to Attract People

How can a robot attract attention? Since there is no obvious answer, we applied an on-line learning algorithm. More specifically, Minerva used a memory-based

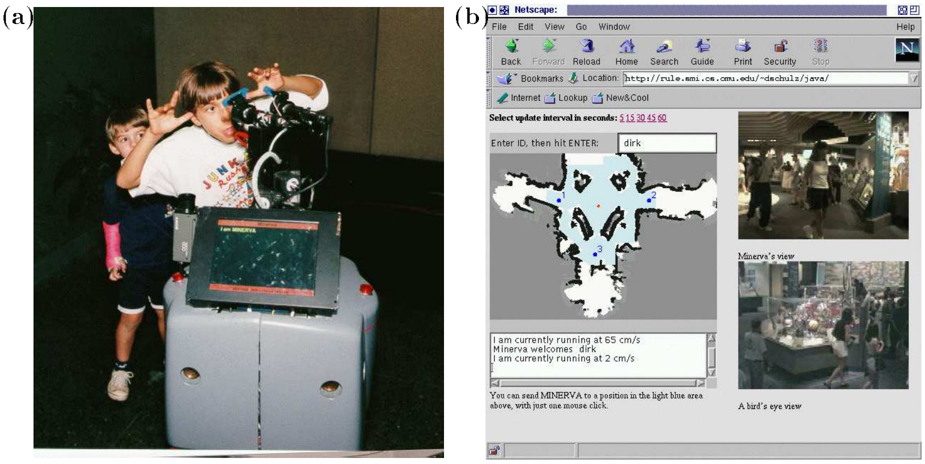

Fig. 7. (a) User interaction; (b) The Web interface for monitoring and control.

reinforcement learning approach [16] (*no* delayed award). 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. During learning, the robot chose with high probability the best known action (so that it attracted as many people as possible); however, with small probability the robot chose a random action, to explore new ways of interaction.

During the two weeks, Minerva performed 201 attraction interaction experiments, each of which lasted approximately 1 minute. Over time, Minerva developed a "positive" attitude (saying friendly things, looking at people, smiling). As shown in Figure 8, 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, 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; hence, we are unable to comment on the effectiveness of individual actions.

8 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

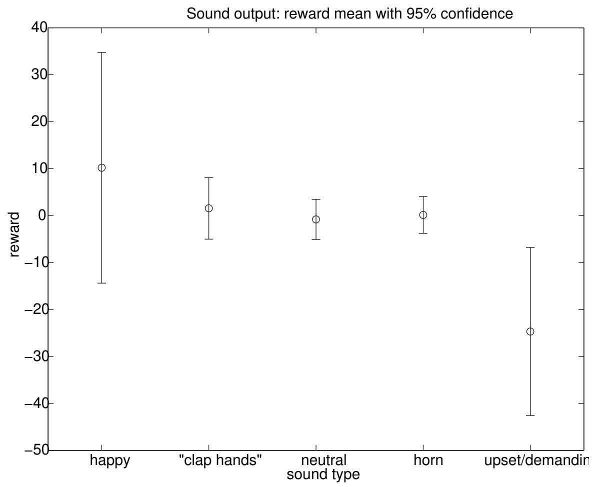

Fig. 8. Statistics of people's response to different styles of interaction (from friendly on the left to upset/demanding on the right). The data were input to a reinforcement learning algorithm, which learned on-line optimal interaction patterns.

National Museum of American History. During more than 94 hours of operation (31.5 hours of motion), Minerva gave 620 tours and visited 2,668 exhibits. The robot interacted with thousands of people, and traversed more than 44km. Its average speed was 38.8 cm/sec, and its maximum speed was 163 cm/sec. The map learning techniques enabled us to develop the robot in 3 weeks (from the arrival of the base platform to the opening of the exhibition). A much improved Web interface (Figure 7b) gave people direct control of the robot when the museum was closed to the public.

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. Special emphasis has been placed on learning, to accommodate the challenges that arose in this unprecedently large, open and crowded environment. In particular, Minerva differed from previous tour-guide robots in its ability to learn maps (of the floor-plan and the ceiling), models of typical navigation times for scheduling tours, and patterns of interaction when attracting people. In addition, Minerva differed from previous tour-guide robots in that she exhibited a "personality," adopting various cues for interaction that people are already familiar with. 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 making this project possible and their superb support. We also thank Anne Watzman for managing our relation to the media, and

Greg Armstrong for his excellent hardware support. Special thanks also to IS Robotics and its Real World Interface Division 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. W. Burgard, A.B., Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. AAAI-98.
2. W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. AAAI-98.
3. A. Elfes. *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation.* PhD thesis, CMU, 1989.
4. C. Breazeal (Ferrell). A motivational system for regulating human-robot interaction. AAAI-98.
5. D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation*, 4(1), 1997.
6. D. Fox, W. Burgard, and S. Thrun. A hybrid collision avoidance method for mobile robots. ICRA-98.
7. D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position estimation for mobile robots in dynamic environments. AAAI-98.
8. I. Horswill. Specialization of perceptual processes. TR 1511, MIT, AI Lab.
9. R. A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, 1960.
10. L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. IROS-96.
11. J.-C. Latombe. *Robot Motion Planning.* Kluwer, 1991.
12. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4, 1997.
13. D. McDermott. The RPL manual, 1993. Can be obtained from `http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/mcdermott.html`
14. 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.
15. R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. IJCAI-95.
16. R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.
17. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), 1998.
18. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31, 1998.