Figure 1: Nursebots Flo (left) and Pearl (center and right) interacting with elderly people during one of our field trips.

this, our software possesses a collection of probabilistic modules concerned with people sensing, interaction, and control. In particular, Pearl uses efficient particle filter techniques to detect and track people. A POMDP algorithm performs high-level control, arbitrating information gathering and performance-related actions. And finally, safety considerations are incorporated even into simple perceptual modules through a risk-sensitive robot localization algorithm. In systematic experiments, we found the combination of techniques to be highly effective in dealing with the elderly test subjects.

Hardware, Software, And Environment

Figure 1 shows images of the robots Flo (first prototype, now retired) and Pearl (the present robot). Both robots possess differential drive systems. They are equipped with two on-board Pentium PCs, wireless Ethernet, SICK laser range finders, sonar sensors, microphones for speech recognition, speakers for speech synthesis, touch-sensitive graphical displays, actuated head units, and stereo camera systems. Pearl differs from its predecessor Flo in many respects, including its visual appearance, two sturdy handle-bars added to provide support for elderly people, a more compact design that allows for cargo space and a removable tray, doubled battery capacity, a second laser range finder, and a significantly more sophisticated head unit. Many of those changes were the result of feedback from nurses and medical experts following deployment of the first robot, Flo. Pearl was largely designed and built by the Standard Robot Company in Pittsburgh, PA.

On the software side, both robots feature off-the-shelf autonomous mobile robot navigation system [5, 24], speech recognition software [20], speech synthesis software [3], fast image capture and compression software for online video streaming, face detection tracking software [21], and various new software modules described in this paper. A final software component is a prototype of a flexible reminder system using advanced planning and scheduling techniques [18].

The robot's environment is a retirement resort located in Oakmont, PA. Like most retirement homes in the nation, this facility suffers from immense staffing shortages. All experiments so far primarily involved people with relatively mild cognitive, perceptual, or physical inabilities, though in need of professional assistance. In addition, groups of elderly in similar conditions were brought into research laboratories for testing interaction patterns.

Navigating with People

Pearl's navigation system builds on the one described in [5, 24]. In this section, we describe three major new modules, all concerned with people interaction and control. These modules overcome an important deficiency of the work described by [5, 24], which had a rudimentary ability to interact with people.

Locating People

The problem of locating people is the problem of determining their x-y-location relative to the robot. Previous approaches to people tracking in robotics were feature-based: they analyze sensor measurements (images, range scans) for the presence of features [13, 22] as the basis of tracking. In our case, the diversity of the environment mandated a different approach. Pearl detects people using map differencing: the robot learns a map, and people are detected by significant deviations from the map. Figure 3a shows an example map acquired using preexisting software [24].

Mathematically, the problem of people tracking is a combined posterior estimation problem and model selection problem. Let N be the number of people near the robot. The posterior over the people's positions is given by

$$p(y_{1,t}, \ldots, y_{N,t} | z^t, u^t, m) \tag{1}$$

where $y_{n,t}$ with $1 \leq n \leq N$ is the location of a person at time t, z^t the sequence of all sensor measurements, u^t the sequence of all robot controls, and m is the environment map. However, to use map differencing, the robot has to know its own location. The location and total number of nearby people detected by the robot is clearly dependent on the robot's estimate of its own location and heading direction. Hence, Pearl estimates a posterior of the type:

$$p(y_{1,t}, \ldots, y_{N,t}, x^t | z^t, u^t, m) \tag{2}$$

where x^t denotes the sequence of robot poses (the path) up to time t. If N was known, estimating this posterior would be a high-dimensional estimation problem, with complexity cubic in N for Kalman filters [2], or exponential in N with particle filters [9]. Neither of these approaches is, thus, applicable: Kalman filters cannot globally localize the robot, and particle filters would be computationally prohibitive.

Luckily, under mild conditions (discussed below) the posterior (2) can be factored into $N + 1$ conditionally independent estimates:

$$p(x^t | z^t, u^t, m) \prod_n p(y_{n,t} | z^t, u^t, m) \tag{3}$$

This factorization opens the door for a particle filter that scales linearly in N. Our approach is similar (but not identical) to the Rao-Blackwellized particle filter described in [10]. First, the robot path x^t is estimated using a particle filter, as in the Monte Carlo localization (MCL) algorithm [7] for mobile robot localization. However, each particle in this filter is associated with a set of N particle filters, each representing one of the people position estimates $p(y_{n,t} | z^t, u^t, m)$. These *conditional* particle filters represent people position estimates *conditioned* on robot path estimates—hence capturing the inherent dependence of people and robot location estimates. The data association between measurements and people is done using maximum likelihood, as in [2]. Under the (false) assumption that this maximum likelihood estimator is always correct, our approach can be shown to converge to the correct posterior, and it does so with update time linear in N. In practice, we found that the data association is correct in the vast majority of situations. The nested particle filter formulation

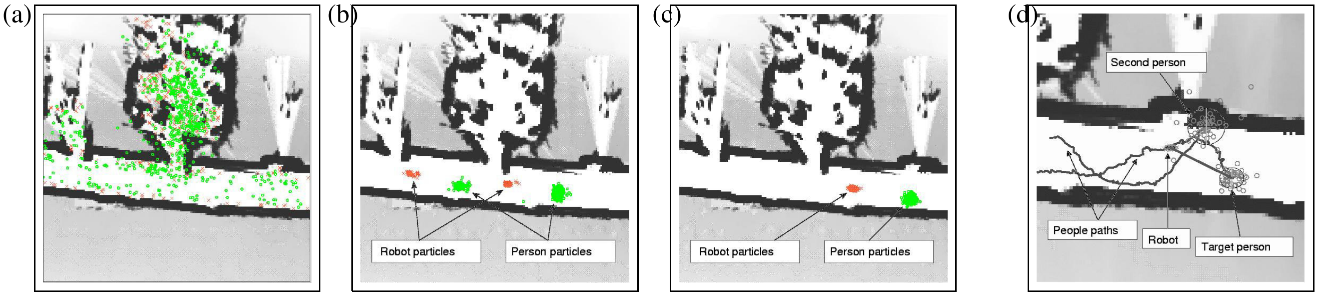

Figure 2: (a)-(d) Evolution of the conditional particle filter from global uncertainty to successful localization and tracking. (d) The tracker continues to track a person even as that person is occluded repeatedly by a second individual.

has a secondary advantage that the number of people N can be made dependent on individual robot path particles. Our approach for estimating N uses the classical AIC criterion for model selection, with a prior that imposes a complexity penalty exponential in N.

Figure 2 shows results of the filter in action. In Figure 2a, the robot is globally uncertain, and the number and location of the corresponding people estimates varies drastically. As the robot reduces its uncertainty, the number of modes in the robot pose posterior quickly becomes finite, and each such mode has a distinct set of people estimates, as shown in Figure 2b. Finally, as the robot is localized, so is the person (Figure 2c). Figure 2d illustrates the robustness of the filter to interfering people. Here another person steps between the robot and its target subject. The filter obtains its robustness to occlusion from a carefully crafted probabilistic model of people's motion $p(y_{n,t+1}|y_{n,t})$. This enables the conditional particle filters to maintain tight estimates while the occlusion takes place, as shown in Figure 2d. In a systematic analysis involving 31 tracking instances with up to five people at a time, the error in determining the number of people was 9.6%. The error in the robot position was 2.5 ± 5.7 cm, and the people position error was as low as 1.5 ± 4.2 cm, when compared to measurements obtained with a carefully calibrated static sensor with ± 1 cm error.

When guiding people, the estimate of the person that is being guided is used to determine the velocity of the robot, so that the robot maintains roughly a constant distance to the person. In our experiments in the target facility, we found the adaptive velocity control to be absolutely essential for the robot's ability to cope with the huge range of walking paces found in the elderly population. Initial experiments with fixed velocity led almost always to frustration on the people's side, in that the robot was either too slow or too fast.

Safer Navigation

When navigating in the presence of elderly people, the risks of harming them through unintended physical contact is enormous. As noted in [5], the robot's sensors are inadequate to detect people reliably. In particular, the laser range system measures obstacles 18 cm above ground, but is unable to detect any obstacles below or above this level. In the assisted living facilities, we found that people are easy to detect when standing or walking, but hard when on chairs (e.g., they might be stretching their legs). Thus, the risk of accidentally hitting a person's foot due to poor localization is particularly high in densely populated regions such as the dining areas.

Following an idea in [5], we restricted the robot's operation area to avoid densely populated regions, using a manually augmented map of the environment (black lines in Figure 3a – the white space corresponds to unrestricted free space). To stay within its operating area, the robot needs accurate localization, especially at the boundaries of this area. While our approach yields sufficiently accurate results on average, it is important to realize that probabilistic techniques never provide hard guarantees that the robot obeys a safety constraint. To address this concern, we augmented the robot localization particle filter by a sampling strategy that is sensitive to the increased risk in the dining areas (see also [19, 25]). By generating samples in high-risk regions, we minimize the likelihood of being mislocalized in such regions, or worse, the likelihood of entering prohibited regions undetected. Conventional particle filters generate samples in proportion to the posterior likelihood $p(x^t|z^t, u^t, m)$. Our new particle filter generates robot pose samples in proportion to

$$l(x_t)\ p(x^t|z^t, u^t, m) \prod_n p(y_{n,t}|z^t, u^t, m) \qquad (4)$$

where l is a *risk* function that specifies how desirable it is to sample robot pose x_t. The risk function is calculated by considering an immediate cost function $c(x, u)$, which assigns costs to actions a and robot states x (in our case: high costs

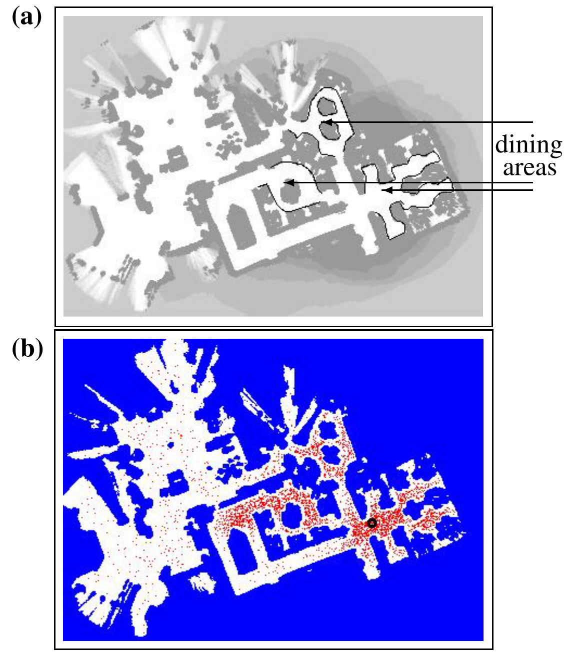

Figure 3: (a) Map of the dining area in the facility, with dining areas marked by arrows. (b) Samples at the beginning of global localization, weighted expected cumulative risk function.

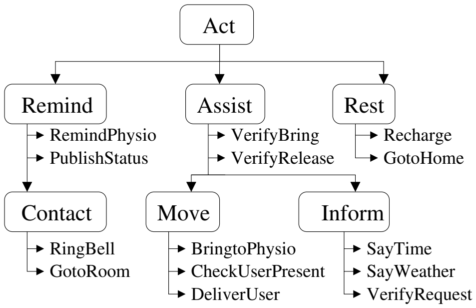

Figure 4: Dialog Problem Action Hierarchy

Observation	True State	Action	Reward
pearl hello	request_begun	say_hello	100
pearl what is like	start_meds	**ask_repeat**	-100
pearl what time is it for will the	want_time	say_time	100
pearl was on abc	want_tv	ask_which_station	-1
pearl was on abc	want_abc	say_abc	100
pearl what is on nbc	want_nbc	**confirm_channel_nbc**	-1
pearl yes	want_nbc	say_nbc	100
pearl go to the that pretty good what	send_robot	ask_robot_where	-1
pearl that that hello be	send_robot_bedroom	**confirm_robot_place**	-1
pearl the bedroom any i	send_robot_bedroom	go_to_bedroom	100
pearl go it eight a hello	send_robot	ask_robot_where	-1
pearl the kitchen hello	send_robot_kitchen	go_to_kitchen	100

Table 1: An example dialog with an elderly person. Actions in bold font are clarification actions, generated by the POMDP because of high uncertainty in the speech signal.

for violating an area constraints, low costs elsewhere). To analyze the effect of poor localization on this cost function, our approach utilizes an augmented model that incorporates the localizer itself as a state variable. In particular, the state consists of the robot pose x_t, and the state of the localizer, b_t. The latter is defined as accurate ($b_t = 1$) or inaccurate ($b_t = 0$). The state transition function is composed of the conventional robot motion model $p(x_t|u_{t-1}, x_{t-1})$, and a simplistic model that assumes with probability α, that the tracker remains in the same state (good or bad). Put mathematically:

$$p(x_t, b_t|u_{t-1}, x_{t-1}, b_{t-1}) = \\ p(x_t|u_{t-1}, x_{t-1}) \cdot \left[\alpha I_{b_t=b_{t-1}} + (1-\alpha) I_{b_t \neq b_{t-1}}\right] \quad (5)$$

Our approach calculates an MDP-style value function, $V(x, b)$, under the assumption that good tracking assumes good control whereas poor tracking implies random control. This is achieved by the following value iteration approach:

$$V(x,b) \longleftarrow \\ \begin{cases} \min_u c(x,u) + \gamma \sum_{x'b'} p(x', b'|x, b, u) V(x', b') & \text{if } b = 1 \text{ (good localization)} \\ \sum_u c(x,u) + \gamma \sum_{x'b'} p(x', b'|x, b, u) V(x', b') & \text{if } b = 0 \text{ (poor localization)} \end{cases} \quad (6)$$

where γ is the discount factor. This gives a well-defined MDP that can be solved via value iteration. The risk function is them simply the difference between good and bad tracking: $l(x) = V(x, 1) - V(x, 0)$. When applied to the Nursebot navigation problem, this approach leads to a localization algorithm that preferentially generates samples in the vicinity of the dining areas. A sample set representing a uniform uncertainty is shown in Figure 3b—notice the increased sample density near the dining area. Extensive tests involving real-world data collected during robot operation show not only that the robot was well-localized in high-risk regions, but that our approach also reduced costs after (artificially induced) catastrophic localization failure by 40.1%, when compared to the plain particle filter localization algorithm.

High Level Robot Control and Dialog Management

The most central new module in Pearl's software is a probabilistic algorithm for high-level control and dialog management. High-level robot control has been a popular topic in AI, and decades of research has led to a reputable collection of architectures (e.g., [1, 4, 12]). However, existing architectures rarely take uncertainty into account during planning.

Pearl's high-level control architecture is a hierarchical variant of a partially observable Markov decision process (POMDP) [14]. POMDPs are techniques for calculating optimal control actions under uncertainty. The control decision is based on the full probability distribution generated by the state estimator, such as in Equation (2). In Pearl's case, this distribution includes a multitude of multi-valued probabilistic state and goal variables:

- robot location (discrete approximation)
- person's location (discrete approximation)
- person's status (as inferred from speech recognizer)
- motion goal (where to move)
- reminder goal (what to inform the user of)
- user initiated goal (e.g., an information request)

Overall, there are 288 plausible states. The input to the POMDP is a factored probability distribution over these states, with uncertainty arising predominantly from the localization modules and the speech recognition system. We conjecture that the consideration of uncertainty is important in this domain, as the costs of mistaking a reply can be large.

Unfortunately, POMDPs of the size encountered here are an order of magnitude larger than today's best exact POMDP algorithms can tackle [14]. However, Pearl's POMDP is a highly structured POMDP, where certain actions are only applicable in certain situations. To exploit this structure, we developed a *hierarchical* version of POMDPs, which breaks down the decision making problem into a collection of smaller problems that can be solved more efficiently. Our approach is similar to the MAX-Q decomposition for MDPs [8], but defined over POMDPs (where states are unobserved).

The basic idea of the hierarchical POMDP is to partition the action space—not the state space, since the state is not fully observable—into smaller chunks. For Pearl's guidance task the action hierarchy is shown in Figure 4, where *abstract actions* (shown in circles) are introduced to subsume logical subgroups of lower-level actions. This action hierarchy induces a decomposition of the control problem, where at each node all lower-level actions, if any, are considered in the context of a local sub-controller. At the lowest level, the control problem is a regular POMDP, with a reduced action space. At higher levels, the control problem is also a POMDP, yet involves a mixture of physical and abstract actions (where abstract actions correspond to lower level POMDPs.)

Let $\bar{u}$ be such an abstract action, and $\pi_{\bar{u}}$ the control policy associated with the respective POMDP. The "abstract" POMDP is then parameterized (in terms of states x, observations z) by assuming that whenever $\bar{u}$ is chosen, Pearl uses

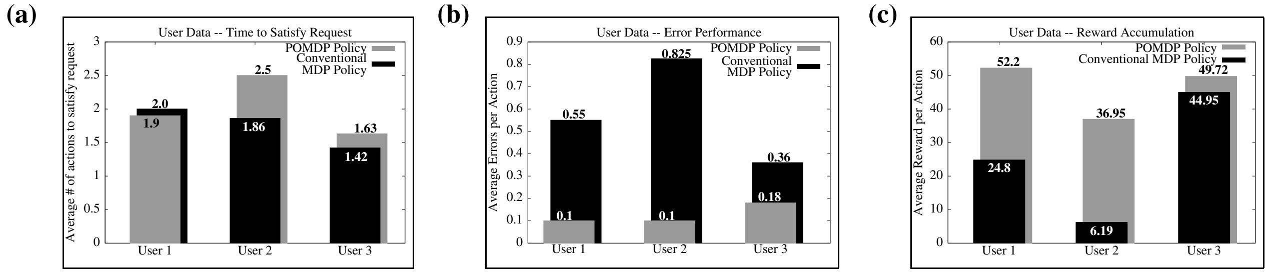

Figure 5: Empirical comparison between POMDPs (with uncertainty, shown in gray) and MDPs (no uncertainty, shown in black) for high-level robot control, evaluated on data collected in the assisted living facility. Shown are the average time to task completion (a), the average number of errors (b), and the average user-assigned (*not* model assigned) reward (c), for the MDP and POMDP. The data is shown for three users, with good, average and poor speech recognition.

lower-level control policy $\pi_{\bar{u}}$:

$$\begin{aligned} p(x'|x,\bar{u}) &= p(x'|x,\pi_{\bar{u}}(x)) \\ p(z|x,\bar{u}) &= p(z|x,\pi_{\bar{u}}(x)) \\ R(x,\bar{u}) &= R(x,\pi_{\bar{u}}(x)) \end{aligned} \tag{7}$$

Here R denotes the reward function. It is important to notice that such a decomposition may only be valid if reward is received at the leaf nodes of the hierarchy, and is especially appropriate when the optimal control transgresses down along a single path in the hierarchy to receive its reward. This is approximately the case in the Pearl domain, where reward is received upon successfully delivering a person, or successfully gathering information through communication.

Using the hierarchical POMDP, the high-level decision making problem in Pearl is tractable, and a near-optimal control policy can be computed off-line. Thus, during execution time the controller simply monitors the state (calculates the posterior) and looks up the appropriate control. Table 1 shows an example dialog between the robot and a test subject. Because of the uncertainty management in POMDPs, the robot chooses to ask a clarification question at three occasions. The number of such questions depends on the clarity of a person's speech, as detected by the Sphinx speech recognition system.

An important question in our research concerns the importance of handling uncertainty in high-level control. To investigate this, we ran a series of comparative experiments, all involving real data collected in our lab. In one series of experiments, we investigated the importance of considering the uncertainty arising from the speech interface. In particular, we compared Pearl's performance to a system that ignores that uncertainty, but is otherwise identical. The resulting approach is an MDP, similar to the one described in [23]. Figure 5 shows results for three different performance measures, and three different users (in decreasing order of speech recognition performance). For poor speakers, the MDP requires less time to "satisfy" a request due to the lack of clarification questions (Figure 5a). However, its error rate is much higher (Figure 5b), which negatively affects the overall reward received by the robot (Figure 5c). These results clearly demonstrate the importance of considering uncertainty at the highest robot control level, specifically with poor speech recognition.

In a second series of experiments, we investigated the importance of uncertainty management in the context of highly imbalanced costs and rewards. In Pearl's case, such costs are indeed highly imbalanced: asking a clarification question is much cheaper than accidentally delivering a person to a wrong location, or guiding a person who does not want to be walked. In this experiment we compared performance using

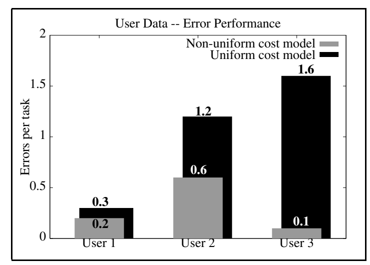

Figure 6: Empirical comparison between uniform and non-uniform cost models. Results are an average over 10 tasks. Depicted are 3 example users, with varying levels of speech recognition accuracy. Users 2 & 3 had the lowest recognition accuracy, and consequently more errors when using the uniform cost model.

two POMDP models which differed only in their cost models. One model assumed uniform costs for all actions, whereas the second model assumed a more discriminative cost model in which the cost of verbal questions was lower than the cost of performing the wrong motion actions. A POMDP policy was learned for each of these models, and then tested experimentally in our laboratory. The results presented in figure 6 show that the non-uniform model makes more judicious use of confirmation actions, thus leading to a significantly lower error rate, especially for users with low recognition accuracy.

Results

We tested the robot in five separate experiments, each lasting one full day. The first three days focused on open-ended interactions with a large number of elderly users, during which the robot interacted verbally and spatially with elderly people with the specific task of delivered sweets. This allowed us to gauge people's initial reactions to the robot.

Following this, we performed two days of formal experiments during which the robot autonomously led 12 full guidances, involving 6 different elderly people. Figure 7 shows an example guidance experiment, involving an elderly person who uses a walking aid. The sequence of images illustrates the major stages of a successful delivery: from contacting the person, explaining to her the reason for the visit, walking her through the facility, and providing information after the successful delivery—in this case on the weather.

In all guidance experiments, the task was performed to completion. Post-experimental debriefings illustrated a uniform high level of excitement on the side of the elderly. Overall, only a few problems were detected during the operation. None of the test subjects showed difficulties understanding the major functions of the robot. They all were able to operate the robot after less than five minutes of introduction. However, initial flaws with a poorly adjusted speech recognition

(a) Pearl approaching elderly (b) Reminding of appointment (c) Guidance through corridor (d) Entering physiotherapy dept. (e) Asking for weather forecast (f) Pearl leaves

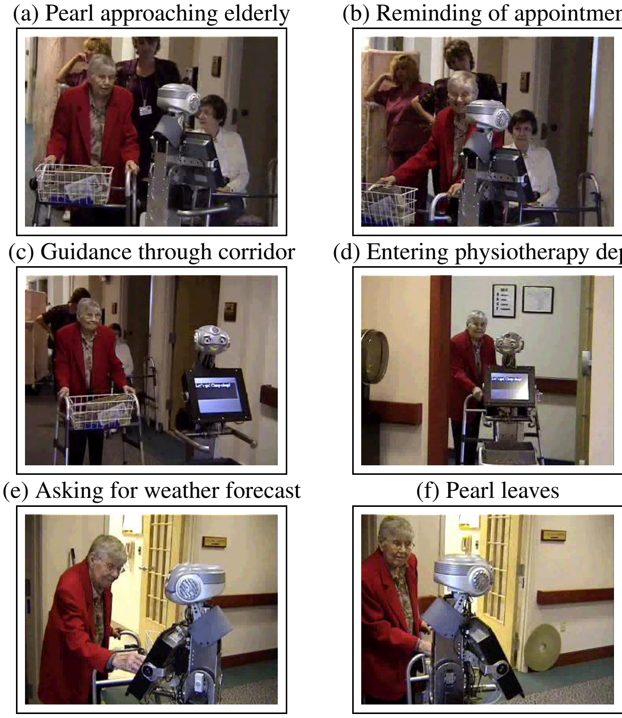

Figure 7: Example of a successful guidance experiment. Pearl picks up the patient outside her room, reminds her of a physiotherapy appointment, walks the person to the department, and responds to a request of the weather report. In this interaction, the interaction took place through speech and the touch-sensitive display.

system led to occasional confusion, which was fixed during the course of this project. An additional problem arose from the robot's initial inability to adapt its velocity to people's walking pace, which was found to be crucial for the robot's effectiveness.

Discussion

This paper described a mobile robotic assistant for nurses and elderly in assisted living facilities. Building on a robot navigation system described in [5, 24], new software modules specifically aimed at interaction with elderly people were developed. The system has been tested successfully in experiments in an assisted living facility. Our experiments were successful in two main dimensions. First, they demonstrated the robustness of the various probabilistic techniques in a challenging real-world task. Second, they provided some evidence towards the feasibility of using autonomous mobile robots as assistants to nurses and institutionalized elderly. One of the key lessons learned while developing this robot is that the elderly population requires techniques that can cope with their degradation (e.g., speaking abilities) and also pays special attention to safety issues. We view the area of assistive technology as a prime source for great AI problems in the future.

Possibly the most significant contribution of this research to AI is the fact that the robot's high-level control system is entirely realized by a *partially observable Markov decision process* (POMDP) [14]. This demonstrates that POMDPs have matured to a level that makes them applicable to real-world robot control tasks. Furthermore, our experimental results suggest that uncertainty matters in high-level decision making. These findings challenge a long term view in mainstream AI that uncertainty is irrelevant, or at best can be handled uniformly at the higher levels of robot control[6, 17]. We conjecture instead that when robots interact with people, uncertainty is pervasive and has to be considered at all levels of decision making, not solely in low-level perceptual routines.

References

[1] R. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1998.

[3] A.W. Black, P. Taylor, and R. Caley. *The Festival Speech Synthesis System*. University of Edinburgh, 1999.

[4] R.A. Brooks. A robust layered control system for a mobile robot. TR AI memo 864, MIT, 1985.

[5] 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*

[6] G. De Giacomo, editor. Notes *AAAI Fall Symposium on Cognitive Robotics*, 1998.

[7] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. *ICRA-99*

[8] T. Dietterich. The MAXQ method for hierarchical reinforcement learning. *ICML-98*.

[9] A. Doucet, N. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer, 2001.

[10] A Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic bayesian networks. *UAI-2000*.

[11] G. Engelberger. Services. In *Handbook of Industrial Robotics*, John Wiley and Sons, 1999.

[12] E. Gat. Esl: A language for supporting robust plan execution in embedded autonomous agents. Noted *AAAI Fall Symposium on Plan Execution*, 1996.

[13] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1), 1999.

[14] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101, 1998.

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

[16] G. Lacey and K.M. Dawson-Howe. The application of robotics to a mobility aid for the elderly blind. *Robotics and Autonomous Systems*, 23, 1998.

[17] G. Lakemeyer, editor. Notes *Second International Workshop on Cognitive Robotics*, Berlin, 2000

[18] C.E. McCarthy, and M. Pollack. A Plan-Based Personalized Cognitive Orthotic. *AIPS-2002*.

[19] P. Poupart, L.E. Ortiz, and C. Boutilier. Value-directed sampling methods for monitoring POMDPs. *UAI-2001*.

[20] M. Ravishankar. Efficient algorithms for speech recognition, 1996. Internal Report.

[21] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998.

[22] D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particles filters and statistical data association. *ICRA-2001*.

[23] S. Singh, M. Kearns, D. Litman, and M. Walker. Reinforcement learning for spoken dialogue systems. *NIPS-2000*.

[24] 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. *International Journal of Robotics Research*, 19(11), 2000.

[25] S. Thrun, Langford. J., and V. Verma. Risk sensitive particle filters. *NIPS-2002*.

[26] US Department of Health and Human Services. Health, United states, 1999. Health and aging chartbook, 1999.