Submitted to IJCAI-97

Discovering Landmarks for Mobile Robot Navigation

Sebastian Thrun
Department of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213-3891

Abstract

Positioning addresses the problem of determining the position of a mobile robot from sensor data. This paper presents a method with which a mobile robot can automatically discover a set of landmarks used as reference for positioning, and also train artificial neural networks to recognize them. The method is applicable to a large variety of sensors and environments. Extensive experiments with a mobile robot equipped with sonar sensors and a camera, illustrate that landmarks selected by the robot are superior to landmarks carefully selected by human experts. Thus, the new method has two advantages over previous ones: It improves the ability of a robot to position itself, and it eliminates the burden of having to select appropriate landmarks by hand.

1 Introduction

To operate successfully, mobile robots must know where they are. *Mobile robot positioning*, that is the process of determining and tracking the position (location) of mobile robots relative to their environments, has received considerable attention over the past few years. Accurate positioning is a key prerequisite for successful navigation in large-scale environments, particularly when global models are used (such as maps, drawings, topological descriptions, CAD models, *cf.* various chapters in [9]).

A recent book on mobile robot positioning [2] demonstrates the importance of the problem and provides a unique description of the state-of-the-art. *Landmark-based* positioning comprises the most popular family of approaches to date [2]. *Landmarks* for the purpose of this paper will be defined as stationary objects, places, or families thereof whose locations are usually known. Many of the early landmark-based approaches reviewed in [2] require artificial landmarks such as bar-code reflectors [4], reflecting tape, ultrasonic beacons, or visual patterns that are easy to recognize, such as black rectangles with white dots [1]. Some of the more recent approaches use more natural landmarks for positioning, which do not require modifications of the environment. For example, the Helpmate robot, which is commercially deployed at over 50 hospitals worldwide, uses ceiling lights to position itself [7]. The approaches in [10; 11] recognize certain gateways, doors, and/or other vertical objects. Map matching approaches often use places as landmarks which are recognized and compared with each other [13; 17; 18]. This list highlights only a few important representatives from a large body of literature on mobile robot positioning.

To the best of our knowledge, none of the existing approaches enables a robot to select the landmarks by itself. It is common practice that a human expert selects the landmarks used for positioning, and provides computer routines for their recognition. As a result, these approaches are tight to a particular type sensor and a particular type environment. For example, if ceiling lights are used as primary landmark–an approach which has become popular in recent years—, positioning will fail if the environment does not contain ceiling lights, or the robot does not a possess a sensor capable of detecting such lights. In addition, it is often unclear how good a set of landmarks really is. The perceptual apparatus of humans differs substantially from that of robots, and landmarks which work well for humans are often not appropriate for robots. All these arguments illustrate the need for methods which enable robots to choose landmarks by themselves.

This paper presents a method that enables a robot to learn its landmarks by itself, and to learn routines for their recognition. In order to do so, the robot has to be provided with a set of sensor snapshots labeled with the position at which they were taken. These data is used (1) to establish correspondence between the sensors of the robot and the world coordinates, and (2) to train artificial neural networks that recognize landmarks. Learning is driven by a simple principle: The minimization of the positioning error.

A large experimental study compares the performance of landmarks selected by human experts to automatically selected landmarks, using our new approach. The results of this study illustrate the two main advantages of the new approach:

1. The burden of having to select landmarks by hand and having to code routines for their recognition is eliminated.
2. Landmarks discovered by our approach consistently outperform our current best hand-selected landmarks (doors and ceiling lights).

Our new method also enables a robot to customize its positioning algorithm to a particular environment and to its sensors, making it more widely applicable than methods that rely on static, built-in landmarks.

2 Approach

This section presents our learning algorithm for the automatic selection of landmarks. Due to brevity, the algorithm is only sketched here; see [16] for the complete description and mathematical derivation.

The input to our algorithm is a set of sensor snapshots along with the position at which they were taken. These data is used to establish the correspondence between world coordinates and sensor values and to select appropriate landmarks. In our experiments, these data is collected by repeatedly starting the robot at a well-defined position, and using its internal dead-reckoning to determine the position of subsequent snapshots. After learning, our approach enables a robot to estimate is position from sensor data.

2.1 Markov Positioning

We will first outline the Markov positioning algorithm upon which our approach is based. Markov positioning has recently been employed successfully in various state-of-the-art mobile robot systems (*e.g.*, [3; 6; 8; 10; 12; 14; 15]). A key feature of Markov positioning is its ability to handle uncertainty and noise in an elegant way. It is also capable of representing ambiguities and degree-of-belief.

Figure 1 illustrates the basic method. The key idea of Markov positioning is to compute a probability density over all positions, denoted by $Bel(\xi)$. $Bel(\xi)$ expresses the robot's subjective belief (uncertainty) that its current position is ξ, where ξ denotes an arbitrary position. Initially, $Bel(\xi)$, reflects the initial uncertainty: If the robot knows its initial position and the goal of positioning is to compensate slippage and drift, $Bel(\xi)$ will be initialized with a point-centered distribution that has a peak at the correct position. Conversely, if the robot has no initial knowledge about its position, $Bel(\xi)$ is initialized with a uniform distribution. The latter situation is depicted in Figure 1a.

The belief $Bel(\xi)$ is updated whenever the robot senses or moves:

1. **Sensing:** The robot's sensors provide information concerning the presence or absence of landmarks. Suppose the robot can recognize n different landmarks. Let $s \in \{0,1\}^n$ denote the *landmark configuration vector* for all n landmarks, *i.e.*, the i-th component of s is 1 if and only if the i-th landmark has been observed in the most recent sensor reading. Then, s is used to refine the position density in the following way:

$$Bel(\xi) \quad \longleftarrow \quad \frac{P(s|\xi)\ Bel(\xi)}{P(s)} \tag{1}$$

$$\text{where } \ P(s) \quad = \quad \int P(s|\bar{\xi})\ Bel(\bar{\xi})\ d\bar{\xi}$$

$P(s)$ is a normalizer which ensures that $Bel(\xi)$ sums up to 1. Here $Bel(\xi)$ on the right-hand side of (1) denotes the belief prior to sensing, and $P(s|\xi)$ denotes the probability of observing the landmark configuration s at position ξ. Update equation (1) is illustrated by the transition of Figure 1a to Figure 1b. Assuming that the robot just observed the landmark "door," it multiples its prior belief by the probability of observing a door. Since the environment contains three doors, the resulting belief has three peaks at the three door positions.

The probability $P(s|\xi)$ is a *map of the environment*, since it specifies the likelihood of observing each individual landmark at each position ξ. For the example in Figure 1, $P(s|\xi)$ specifies the chances of observing doors at the various positions ξ. In general, $P(s|\xi)$ can be modeled by parametric densities such as Gaussians (Kalman filters) or mixtures thereof [15], or by splines or piecewise constant functions. In our implementation, $P(s|\xi)$ is estimated from data. $P(s|\xi)$ is represented by a piecewise constant function, a representation previously employed successfully in [8].

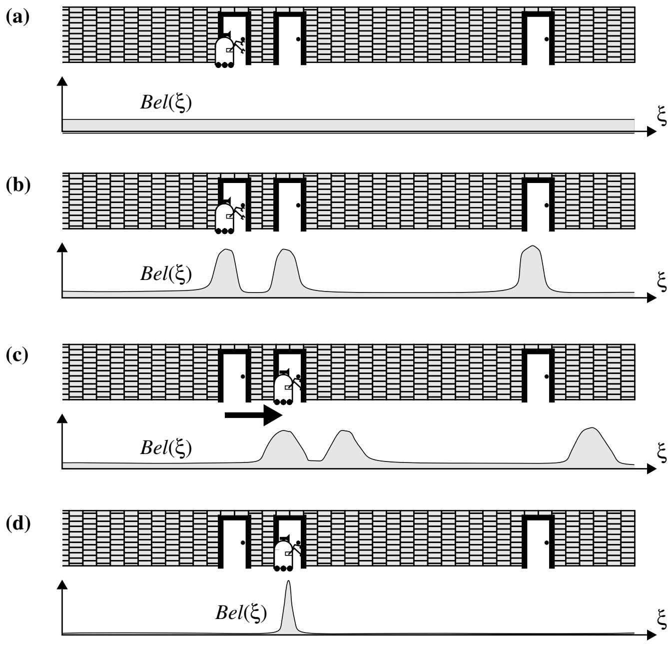

Figure 1: Markov positioning, see text.

2. **Motion:** Whenever the robot executes a motion command, denoted by u, it updates its internal belief according the following rule:

$$Bel(\xi) \quad \longleftarrow \quad \int P(\xi|u,\xi')\ Bel(\xi')\ d\xi' \tag{2}$$

Here $Bel(\xi')$ on the right-hand side of (2) is the robot's belief prior to executing u. The conditional probability $P(\xi|u,\xi')$ denotes the probability that the motion command u, if executed at ξ', makes the robot move to ξ. For example, if u is the command for moving 1 meter forward, $P(\xi|u,\xi')$ will be a bell-shaped density centered around $\xi = \xi' + 1$m. An approximate version of $P(\xi|u,\xi')$ can easily be derived from the robot's kinematics.

Figure 1c illustrates the effect of robot motion on the internal belief $Bel(\xi)$. Here the robot moves forward. As a result, the density is shifted towards the right, and (since motion introduces uncertainty) is flattened out. Figure 1d illustrates the effect of a second sensor query. Here it is assumed that the robot detects a second door next to it, and applies Equation (1) to update $Bel(\xi')$. The resulting density contains a single peak at the correct position, which can be interpreted that the robot now knows where it is.

2.2 The Average Positioning Error

The key to automatically selecting landmarks is to *minimize the positioning error after taking a sensor snapshot*. To measure this error, let us examine the update Equation (1), which specifies the way the robot updates its internal probability in response to a sensor reading. Equation (1) transforms the belief $Bel(\xi)$ prior to sensing to a new, refined belief, which is usually more accurate. The accuracy of this new (posterior) belief depends on the landmarks employed by the robot.

This argument will now be formalized. Let ξ^* denote the *true* position of the robot, and let $e(\xi^*, \xi)$ denote an error function for measuring the error between the true position ξ^* and an arbitrary other position ξ (e might be the Kullback-Leibler divergence or a metric distance). The *average (Bayesian) positioning error at* ξ^*, denoted by $E(\xi^*)$, is obtained by integrating the error e over all belief position ξ, weighted by the corresponding likelihood:

$$E_{\text{pri}}(\xi^*) \quad := \quad \int e(\xi^*, \xi) \; Bel(\xi) \; d\xi \qquad (3)$$

If this error is computed right before taking a sensor snapshot, it is called the *prior error at* ξ^* with respect to the next sensor reading; hence the notation $E_{\text{pri}}(\xi^*)$.

We will now derive an expression that describes the error after taking a sensor snapshot. By definition, the robot will sense the landmark configuration s with probability $P(s|\xi^*)$ when being at ξ^*. In response, it will update its belief according to Equation (1). The *posterior error at* ξ^*, which is the error the robot is expected to make at ξ^* after sensing, is obtained by applying the update rule (1) to the error (3)

$$E_{\text{post}}(\xi^*) \;=\; \sum_s \int e(\xi^*, \xi) \frac{P(s|\xi) \; Bel(\xi)}{P(s)} P(s|\xi^*) \; d\xi \qquad (4)$$

E_{post} is averaged over all possible landmark vectors s, weighted by their likelihood $P(s|\xi^*)$. The normalizer $P(s)$ is computed just like in Equation (1).

Thus far, the posterior error E_{post} corresponds to a single position ξ^* only. By averaging over all possible positions ξ^*, weighted by their likelihood of occurrence $P(\xi^*)$, we obtain the *average posterior error*:

$$E_{\text{post}} \quad := \quad \int E_{\text{post}}(\xi^*) \; P(\xi^*) \; d\xi^* \qquad (5)$$

In the absence of any better knowledge, we will assume that $P(\xi^*)$ is *uniformly distributed*, *i.e.*, each robot position is equally likely.

Notice that the posterior error E_{post} depends on (1) the map $P(s|\xi)$, and (2) the prior belief (uncertainty) $Bel(\xi)$. The map $P(s|\xi)$ depends on the landmarks used by the robot. Thus, different landmarks can yield different average posterior error E_{post}, an observation that is crucial for the selection of landmarks.

The posterior error E_{post} enables a robot to *compare* landmarks with one another. Put differently, by comparing the error E_{post} for different landmarks, the robot can decide which one yields more accurate positioning. This itself is a key result of our research, since it enables one to compare different landmark-based positioning routines to each other, and to decide which one is best.

2.3 Neural Network Training

The objective of landmark learning is to minimize E_{post}. To establish broad enough a search space, landmarks are recognized by artificial neural networks. These networks—there are n of them for the n components of s—map sensor scans (*e.g.*, camera images, sonar scans) to n-dimensional vectors $s \in [0,1]^n$. Consequently, every regularity of the environment which can be recognized by n neural networks can potentially be a landmark. Landmarks, thus, need not to correspond to individual objects in the environment; in fact, they might correspond to groups of objects or places with similar features (such as doors), even to features which would not necessarily be used as landmark by humans (such as a reflection on a shiny wall). Obviously, some features will be better than others; Minimizing E_{post} will be the driving force in the selection of landmarks.

The robot selects landmarks indirectly, by training neural networks which recognize them. One way to minimize E_{post} would be to repeatedly guess weights of the neural networks at random, then use the data to construct $P(s|\xi)$, compute E_{post}, and then pick those weights that minimize E_{post}. However, random sampling is of course hopelessly inefficient. Our learning approach utilizes the fact that E_{post} is differentiable in the weights of the neural networks (see [16] for an exact derivation of the derivatives of E_{post}). To train the networks, each weight w_{ij} is adjusted in proportion to the negative gradient $-\nabla_{w_{ij}} E_{\text{post}}$ (just like weights and biases are adjusted in Back-Propagation): Repeated application of this gradient descent update rule leads to the nearest local minimum in weight space; thus, the approach is capable of identifying the locally best landmark detectors.

Notice the difference of our training regime to conventional supervised learning (Back-Propagation): Our approach does not assume the existence of target values for the outputs of the networks. Instead, it assumes there is a differentiable error function E_{post}, which is to be minimized. The output units of the individual networks bear some resemblance to hidden units in a conventional Back-propagation network, whose response characteristic is not pre-determined but develops during training.

As noted in the previous section, the results of training depend on the prior uncertainty $Bel(\xi)$. Different uncertainties $Bel(\xi)$ lead to different landmarks. For example, whether or not a robot is underneath a ceiling light is considerably uninformative if the robot is globally uncertain in its position. The same information can be most informative in situations where the uncertainty is small. In general, every time the belief $Bel(\xi)$ is updated (through (1) or (2)), the robot could train its networks to find those landmarks that are optimal for the current belief $Bel(\xi)$. In practice, we found it to be sufficient to train a collection of six prototypical uncertainties (see below), and to use those in everyday operation. Training the landmark detectors off-line is necessary, since our current training routines are too slow for real-time learning.

3 Experimental Results

3.1 Three Setups

The central question driving our experiment was: Will landmarks selected automatically by the robot outperform landmarks carefully selected by a human expert? In our experimental study we were therefore particularly interested in comparing our method to other state-of-the-art positioning algorithms.

Figure 2: The RWI B21 robot used in our research.

Figure 2 depicts the robot used in our experiments. A total of three experimental comparisons was conducted, using three different experimental setups (and three different datasets). We based our first two comparisons on a mobile robot control system built by a different research team in our department. This team is primarily interested in reliable long-term mobile robot operation; for which reason it has operated a mobile robot almost on a daily basis over the last two years [8; 14]. Accurate positioning is essential for reliable point-to-point navigation. Their positioning algorithm is a (restricted) version of Markov positioning, using *doors* as landmarks. They chose doors because in our building doors are by far the most regular and easily distinguishable feature. In experimental setup #1, the camera of our robot was directed towards the outside wall of the building. Here the number of doors is considerably large. In setup #2, the camera was directed towards the interior of the building. Here doors are about twice as rare and 60% wider. In both cases, the environment was dynamic and did not obey the Markov assumption: People frequently blocked the robot's sensors, doors were sometimes open, sometimes closed, and since our corridor possesses two large windows the illumination depended on the time of day at which the data was recorded. Just like the other research team, we trained a neural network (with supervised learning) to recognize doors. After training, the network recognized doors with 97.7% accuracy, measured on an independent test set.

In the third comparison (setup #3), we used ceiling lights as landmark. Here the camera was pointed upward. Since ceiling lights are considerably easy to recognize, stationary, and barely blocked by obstacles, they have become extraordinary popular for mobile robot positioning. As in the other two comparisons, we trained a single network to recognize such lights.

We collected a total of 9,815 sensor snapshots, which were divided about equally between the three different datasets (3,232+3,110+3,473). In each experiment, approximately half the data was used for training, and the other half for testing. To collect the data efficiently, we repeatedly had our robot navigate in a 89 meter long segment of our building, using its built-in obstacle avoidance and wandering routines [5]. In each "run," the robot was started at the same initial position and sent down the same corridor in the same direction. Instead of measuring the exact position of the robot by hand (which is practically infeasible for the large number of positions used here), we used the robot's odometry to derive the position label. While odometry is certainly inaccurate in the long run, we found it to be sufficiently accurate in the short run. After some cleanup using the position tracking method described in [17], the position data was accurate within $\pm.5$m. Color camera images were pre-processed using various low-level feature extraction techniques. The final dimension of all sensor values (vision and sonar) was 164; thus, each neural network had 164 input units. We employed layered networks with six hidden units and a single output unit. In our initial experiments we found out that neither the image encoding nor the number of hidden units had a significant influence on the results.

The reader should notice that our results are limited in that in all experiments the robot moved along the same corridor in the same direction. However, we have no reason to believe that the result for the opposite direction or for different corridors would differ substantially. A second, more important restriction resulted from the fact that we used a *one-dimensional* variable to model position. In general, the position of a mobile robot comprises three dimensions: x, y, and θ. Representing ξ by a one-dimensional variable greatly reduced the number of examples required to estimate the density $P(s|\xi)$. This simplification is somewhat justified by the fact that our corridors are narrow, and our obstacle avoidance routines attempts to keep the robot always on a one-dimensional manifold in the center of the corridor.

Different landmarks cannot be compared by just a single experiment. This is because as motivated above, the utility of a landmark greatly depends on the prior uncertainty $Bel(\xi)$. Thus, to conduct as insightful a comparison as possible, we generally evaluated the utility of the different landmarks under several different uncertainties $Bel(\xi)$, all of which were uniformly distributed and centered around the true (unknown) position. In particular, these uniform distributions had the following widths: $[-1\text{m}, 1\text{m}]$, $[-2\text{m}, 2\text{m}]$, $[-5\text{m}, 5\text{m}]$, $[-10\text{m}, 10\text{m}]$, $[-50\text{m}, 50\text{m}]$, and $[-89\text{m}, 89\text{m}]$ (*i.e.*, global uncertainty).

3.2 Results For a Single Network

The central result of our study is: The landmarks selected by the robot uniformly outperform the ones chosen carefully by a human expert. This result was confirmed in every single comparison!

In an initial series of experiments, only a single network was used, *i.e.*, $n = 1$. Figures 3-5 survey the main results obtained in the three different experimental setups (camera pointed to the outside wall, camera pointed to the inside wall, camera pointed towards the ceiling). The vertical axes measure the *error reduction*: If, for example, the prior error is four times as large as the posterior error after taking a sensor snapshot, the corresponding error reduction is 75%. The advantage of the error reduction (over the absolute error) is that all results are in the same scale, which facilitates their comparison. The vertical axes in Figures 3-5 show results obtained for different prior uncertainties $Bel(\xi)$. In all diagrams, the dashed lines depict the error reduction obtained with the human-selected landmarks (door and ceiling lights), whereas the solid lines show the error reduction for robot-selected landmarks. 95% confidence bars are also plotted.

As can be seen from these figures, landmarks selected by the robot are uniformly superior to landmarks selected by hand. For example, the dashed line in Figure 3 indicates that doors

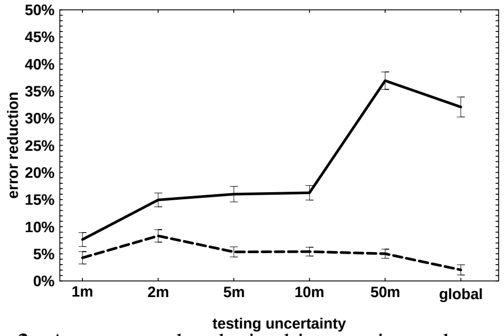

Figure 3: Average results obtained in experimental setup #1. The dashed (solid) line indicates the error reduction of manually (automatically) selected landmarks, for different prior uncertainties.

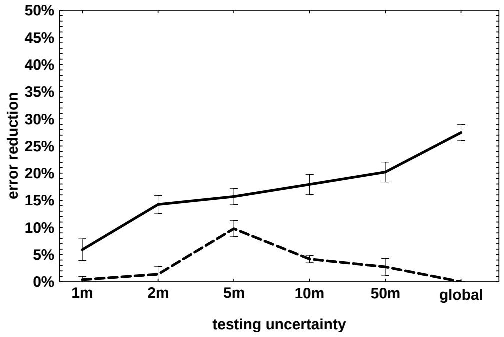

Figure 4: Results obtained for experimental setup #2.

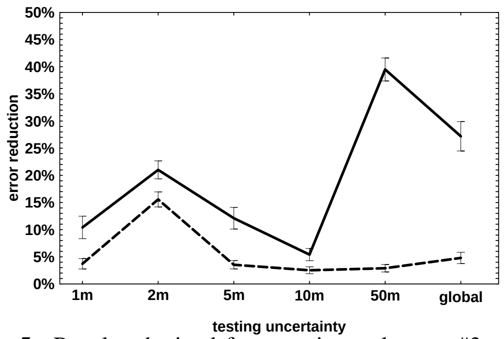

Figure 5: Results obtained for experimental setup #3. The poor performance for ±10m is currently being analyzed in our lab.

are best when the uncertainty of the robot is ±2m. Here, a single sensor reading reduces the error on average by 8.31%. Our new approach finds a network (and hence a set of landmarks) which reduces the error by 14.9% for ±2m uncertainty, under otherwise equal conditions, which is 1.8 times as good as the 8.31% reported above. The advantage of the robot-selected landmarks becomes even larger for more global uncertainties. For example, if the robot's uncertainty is ±50m, the new approach reduces the error by 36.9%, which is approximately 7.4 times as good the average reduction when doors are used as landmarks (5.02%).

The same result can be found in Figures 4 and 5. Here, too, the robot-selected landmarks are uniformly superior to the hand-selected ones. All differences in Figures 3-5 are statistically significant at the 95% level.

experimental setup	#1	#2	#3
hand-selected	8.3±1.1%	1.4±1.5%	15.6±1.4%
$n = 1$	14.9±1.3%	14.2±1.6%	21.0±1.7%
$n = 2$	33.5±1.9%	36.7±1.9%	34.5±2.0%
$n = 3$	39.7±2.0%	36.3±2.3%	41.5±1.8%
$n = 4$	41.7±2.5%	36.7±1.9%	42.9±1.6%

Table 1: Comparison of hand-selected and robot-selected landmarks (±2m uncertainty). n specifies the dimension of the landmark vector s (*i.e.*, the number of networks), and the numbers in the table give the average error reduction ± their 95% confidence interval. The results in the first two rows (labeled "hand-selected" and "$n = 1$") can also be found in Figures 3-5 at ±2m.

3.3 Results For Multiple Networks

The performance difference between robot-selected and hand-selected landmarks is even larger if multiple networks are trained simultaneously, *i.e.*, if $n>1$. Multiple networks tend to specialize to different landmarks. This is a side effect of minimizing E_{post}: If all networks recognized the same set of landmarks, their output would be redundant and the result would be the same as if $n = 1$. If their output differs, however, the networks will extract *more* information from the sensors as more landmarks can be distinguished, which will usually lead to improved results.

We conducted a series of experiments in the three experimental domains for ±2m uncertainty. Table 1 summarizes the average error reduction (and 95% confidence interval) for the three different experimental setups. The two primary results are: (1) The error reduction increases as the number of networks n increases. (2) For larger choices of n, the advantage of robot-selected landmarks over hand-selected landmarks is enormous. For example, hand-selected doors reduce the error by 8.3% or 1.4%, whereas the robot-selected landmarks reduce the error by as much as 41.7%, or 36.7%. There results truly surpassed all our expectations.

3.4 What Landmarks Does the Robot Discover?

Analyzing the trained neural networks led to various interesting findings. For example, in several experiments the robot recognized a certain spot in our corridor, whose physical appearance did appear to differ from other places. Closer investigation of this spot unveiled that due to an irregular pattern of the ceiling lights, it is slightly darker than the rest of the corridor. This illumination difference is barely visible for human eyes (since human eyes compensate for the total level of illumination). Our camera, however, is very sensitive to the total level of illumination, which explains why the robot selected this spot as a landmark.

When landmarks were sought for small uncertainties (such as ±1m or ±2m), the robot selected doors as its primary landmark, just like our hand-selected landmarks. However, the robot also used dark spots on the wall and adjacent corridors as landmarks, which accounts for the performance difference. When the networks were trained for larger uncertainties, the nature of the landmarks changed. For example, for global uncertainty in the experimental setup #1, the network became a wall color detector. A third of our corridor is orange, and two third are light brown. If the robot is globally ignorant about its position, wall color is vastly superior to doors, as illustrated by the enormous performance difference (32.1% for robot-selected landmarks instead of 2.04% for doors).

4 Conclusion

We have presented a new positioning method which enables a robot to select its landmarks by itself, and to train artificial neural network to recognize them. Equipped with a set of sensor snapshots labeled by their (approximate) position, our new method trains neural networks to minimize the expected positioning error after taking a sensor snapshot (posterior error). The resulting networks recognize landmarks which, according to the data, are most informative for estimating a robot's position. In a large experimental study, landmarks selected by our approach was empirically found to outperform landmarks carefully selected by human experts. Thus, our new approach has two advantages over previous ones: It improves the ability of a robot to localize itself, and it eliminates the burden of having to select appropriate landmarks by hand.

Notice that the principle of minimizing the posterior error can be used for a variety of related problems:

1. **Active perception.** As shown in [16], the expected posterior error E_{post} is a function of the pointing direction of the robot's sensors. For example, if the robot has control over the pointing direction of its camera, minimizing E_{post} provides a rational criterion for determining the optimal pointing direction. In [16], experimental results are described that demonstrate the appropriateness of this approach for active vision.
2. **Sensor selection.** Our framework is also capable of determining what sensors a mobile robot to equip with, and where to mount them. By comparing E_{post} for different type sensors and mounting positions, one can identify which ones best serve the purpose of positioning. This might be helpful for the designers of mobile robots.
3. **Active navigation.** Finally, minimizing E_{post} yields a rational criterion for determining the motion direction of a robot (*cf.* [16]). The expected a posteriori error clearly depends on the location where a measurement is taken; thus, if the task of the robot is to position itself as efficiently as possible, our framework can be used to determine a motion direction. A similar idea has recently been presented by Kaelbling and colleagues [6], who proposes to use the expected entropy of $Bel(\xi)$ to determine the motion direction of a robot.

A potential limitation of the current approach is its computational complexity. Training the networks consumes a considerable amount of time (between 30 minutes and 4 hours on a Pentium Pro). Once the networks are trained, estimating the map and updating the belief are extremely fast and can easily be done in real-time. Since the networks can be trained offline (meaning that a fixed set of landmarks is used in everyday operation), we do not view this as a major limitation.

A key advantage of our method is its ability to customize itself to different environments. For example, while ceiling lights might be appropriate landmarks in some environments, they are inappropriate in others. Any routine that relies on a static, built-in type of landmarks is bound to fail in environments which do not possess the respective landmarks. By providing a method that supports the automatic customization of a robot to its environment, we hope to have provided a new level of flexibility that facilitates the design of future service robots (*e.g.*, service robots operated in private homes, whose design varied greatly from home to home).

References

[1] J. Borenstein. *The Nursing Robot System*. PhD thesis, Technion, Haifa, Israel, 1987.

[2] J. Borenstein, B. Everett, , and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. Peters, 1996.

[3] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. *Proceedings of AAAI*, 1996.

[4] H.R. Everett, D.W. Gage, G.A. Gilbreth, R.T. Laird, and R.P. Smurlo. Real-world issues in warehouse navigation. *Proceedings of the SPIE*, 1994.

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

[6] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. *Proceedings of IROS*, 1996.

[7] S. King and C. Weiman. Helpmate autonomous mobile robot navigation system. In *Proceedings of SPIE*, 1990.

[8] S. Koenig and R. Simmons. Passive distance learning for robot navigation. In *Proceedings of ICML*, 1996.

[9] D. Kortenkamp, R.P. Bonassi, and R. Murphy, editors. *AI-based Mobile Robots: Case studies of successful robot systems*, MIT Press (to appear).

[10] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of AAAI*, 1994.

[11] B. Kuipers and Y.-T. Byun. A robust qualitative method for spatial learning in unknown environments. In *Proceeding of AAAI*, 1988.

[12] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

[13] B. Schiele and J. Crowley. A comparison of position estimation techniques using occupancy grids. In *Proceedings of ICRA*, 1994.

[14] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of IJCAI*, 1995.

[15] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehnicles*, Springer, 1990.

[16] S. Thrun. A Bayesian Approach to Landmark Discovery and Active Perception for Mobile Robot Navigation TR CMU-CS-96-122, 1996.

[17] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In [9].

[18] B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, (to appear).