Active Markov Localization for Mobile Robots

Dieter Fox, Wolfram Burgard

Dept. of Computer Science III, University of Bonn,
D-53117 Bonn, Germany [1]

Sebastian Thrun

Dept. of Computer Science, Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Localization is the problem of determining the position of a mobile robot from sensor data. Most existing localization approaches are passive, i.e., they do not exploit the opportunity to control the robot's effectors during localization. This paper proposes an *active* localization approach. The approach is based on Markov localization and provides rational criteria for (1) setting the robot's motion direction (exploration), and (2) determining the pointing direction of the sensors so as to most efficiently localize the robot. Furthermore, it is able to deal with noisy sensors and approximative world models. The appropriateness of our approach is demonstrated empirically using a mobile robot in a structured office environment.

Key words: Robot Position Estimation, Autonomous Service Robots

1 Introduction

To navigate reliably in indoor environments, a mobile robot must know where it is. Over the last few years, there has been a tremendous scientific interest in algorithms for estimating a robot's location from sensor data. A recent book on this issue [3] illustrates the importance of the localization problem and provides a unique description of the state-of-the-art.

The majority of existing approaches to localization are *passive*. Passive localization exclusively addresses the estimation of the location based on an incoming stream of sensor data. It rests on the assumption that neither robot motion, nor the pointing direction of the robot's sensors can be controlled. *Active localization* assumes that during localization, the localization routine has partial or full control over the robot, providing the opportunity to increase the efficiency and the robustness of localization. Key open issues in active localization are *"where to move"* and *"where to look"* so as to best localize the robot.

This paper demonstrates that active localization is a promising research direction for developing more efficient and more robust localization methods. In other sub-fields of artificial intelligence

[1] The work of these authors was partially supported by EC Contract No ERBFMRX-CT96-0049 under the TMR Programme.

(such as heuristic search and machine learning), the value of active control during learning and problem solving has long been recognized. It has been shown, both through theoretical analysis and practical experimentation, that the complexity of achieving a task can be greatly reduced by actively interacting with the environment. For example, choosing the right action during exploration can reduce exponential complexity to low-degree polynomial complexity, as for example shown in Koenig's and Thrun's work on exploration in heuristic search and learning control [14,21]. Similarly, active vision (see e.g., [1]) has also led to results superior to passive approaches to computer vision. In the context of mobile robot localization, actively controlling a robot is particularly beneficial when the environment possesses relatively few features that enable a robot to unambiguously determine its location. This is the case in many office environments. For example, corridors and offices often look alike for a mobile robot, hence random motion or perpetual wall following is often incapable for determining a robot's position, or very inefficient.

In this paper we demonstrate that actively controlling the robot's actuators can significantly improve the efficiency of localization. Our framework is based on *Markov localization*, a passive probabilistic approach to localization which was recently developed in different variants by [6,11,17,19]. At any point in time, Markov localization maintains a probability density (*belief*) over the entire configuration space of the robot; however, it does not provide an answer as to how to control the robot's actuators. The guiding principle of our approach is to control the actuators so as to minimize future expected uncertainty. Uncertainty is measured by the entropy of future belief distributions. By choosing actions to minimize the expected future uncertainty, the approach is capable of actively localizing the robot.

The approach is empirically validated in the context of two localization problems:

(1) **Active navigation,** which addresses the questions of where to move next, and
(2) **Active sensing,** which addresses the problem of what sensors to use and where to point them.

Our implementation assumes that initially, the robot is given a metric map of its environment, but it does not know where it is. Notice that this is a difficult localization problem; most existing approaches (see, e.g., [3]) concentrate on situations where the initial robot location is known and are not capable of localizing a robot from scratch. Our approach has been empirically tested using a mobile robot equipped with a circular array of 24 ultrasound sensors. The key experimental result is that the efficiency of localization is improved drastically by actively controlling the robot's motion direction and by actively controlling its sensors.

2 Related Work

While most research has concentrated on passive localization (see e.g., [3]), active localization has received considerably little attention in the mobile robotics community. This is primarily because the majority of literature concerned with robot control (e.g., the planning community) assumes that the position of the robot is known, whereas research on localization has mainly focused on the estimation problem itself. In recent years, navigation under uncertainty has been addressed by a few researchers [17,19], who developed the Markov navigation paradigm. However, both their approaches do not aim at actively localizing the robot. Localization occurs as a side effect when operating the robot under uncertainty. Moreover, as argued by Kaelbling [11], there exist conditions under which the approach reported in [19] can exhibit cyclic behavior due to uncertainty in localization.

On the forefront of localization driven navigation, [15] used a rehearsal procedure to check whether a location has been visited while learning a map. In [13] the problem of active localization is treated theoretically in finding "critical directions within the environment" under the assumption of perfect sensors.

In [12], acting in the environment is modeled as a partially observable Markov decision process (POMDP). This approach derives an *optimal* strategy for moving to a target location given that the position of the robot is not known perfectly. In [11] this method is extended by actions allowing the robot to improve its position estimation. This is done by minimizing the expected entropy after the immediate next robot control action. While this approach is computationally tractable, its greediness might prevent it from finding efficient solutions in realistic environments. For example, if disambiguating the robot's position requires the robot to move to a remote location, greedy single-step entropy minimization can fail to make the robot move there. In our own work [20], we have developed robot exploration techniques for efficiently mapping unknown environments. While such methods give better-than-random results when applied to localization, their primary goal is not to localize a robot, and there are situations in which they will fail to do so.

3 Markov Localization

3.1 General Equations

This section briefly outlines the basic Markov localization algorithm upon which our approach is based (see [18] for a detailed introduction). The key idea of Markov localization is to compute a probability distribution over all possible locations in the environment. $Bel(L_t = l)$ denotes the robot's belief of being at position l at time t. Here, l is a location in x-y-α space where x and y are Cartesian coordinates and α is the robot's orientation. Initially, $Bel(L_0)$ reflects the initial state of knowledge: if the robot knows its starting position, $Bel(L_0)$ is centered on the correct location; if the robot does not know its initial location, $Bel(L_0)$ is uniformly distributed to reflect the global uncertainty of the robot—the latter is the case in all our experiments.

The belief Bel is updated whenever …

… the robot moves. Robot motion is modeled by the conditional probability $p_a(l \mid l')$. $P_a(l \mid l')$ denotes the probability that motion action a, when executed at l', carries the robot to l. $p_a(l \mid l')$ is used to update the belief upon robot motion, where $\widehat{Bel}(L_t = l)$ denotes the resulting belief at time t:

$$\widehat{Bel}(L_t = l) \leftarrow \sum_{l'} p_a(l \mid l')\ Bel(L_{t-1} = l') \quad (1)$$

In our implementation, $p_a(l \mid l')$ is obtained from a model of the robot's kinematics.

… the robot senses. Let s denote a sensor reading, and $p(s \mid l)$ the likelihood of perceiving s at l. The probability $p(s \mid l)$ is usually referred to as *map of the environment*, since it specifies the probability of observations at the different locations in the environment. When sensing s, the belief is updated according to the following rule:

$$Bel(L_t = l) \leftarrow \frac{p(s \mid l)\ \widehat{Bel}(L_t = l)}{p(s)} \quad (2)$$

Here $p(s)$ is a normalizer that ensures that the belief Bel sums up to 1 over all l.

While our description of Markov navigation is brief, it is important that the reader grasps the essentials of the approach: The robot maintains a belief distribution $Bel(L)$ which is updated upon robot motion, and upon the arrival of sensor data. Such probabilistic representations are well-suited for mobile robot localization due to their ability to handle ambiguities and to represent degree-of-belief.

3.2 Position Probability Grids

While all implementations of Markov localization rely on the update cycle presented in the previous section, the existing implementations can be distinguished particularly by the discretization of the state space L and the world model they rely on. [17,19,11,10] use a *topological* representation of the belief state, where each possible location l corresponds to a node in a topological map of the environment. Due to the nature of this representation, the sensings s are abstract features extracted from proximity sensors (e.g. a percept s can be the detection of a T-junction between two hallways within an office building).

In contrast to these techniques, our implementation is based on a fine-grained, *geometric* variant of Markov localization, where the spatial resolution is usually between 10 and 15 cm and the angular resolution is usually 1 or 2 degrees. We obtain the likelihood $p(s \mid l)$ directly from a metric model of the environment and a model of proximity sensors. The advantage of this approach is that it can operate based on the raw data of proximity sensors and thus permits the exploitation of arbitrary geometric features of the environment such as the width of a corridor or the size of a cupboard. Additionally, it can easily be extended to incorporate the abstract features or landmarks used in [17,19,11]. The disadvantage of this grid-based method lies in the huge state space which has to be maintained. For a mid-size environment of size $30 \times 30\text{m}^2$ and an angular resolution of 2° the state space consists of $7,200,000$ states. To deal with such state spaces in real-time, we modified the basic approach in several ways In essence, our efficient implementation is based on the following two techniques, which reduce the complexity by several orders of magnitude:

(1) **Pre-computation.** The sensor model $p(s \mid l)$ for proximity sensors is pre-computed based on a map and stored in a large look-up table. More specifically, our approach pre-computes for each x-y location and each possible sensor angle α the distance o to the nearest obstacle in that direction, which is the expected measurement in a noise-free world. During localization, probabilities of the type $p(s \mid o)$ are computed by a mixture of a Gaussian-uniform and a geometric density function. An example of such a function given a specific distance o is depicted in Figure 1.

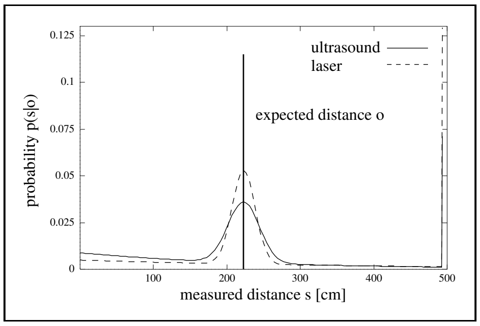

Fig. 1. Sensor model of ultrasound sensors and laser-range finders.

Please note that the higher accuracy of laser-range finders versus ultrasound sensors is represented by a smaller standard deviation of the Gaussian distribution. With this sensor model, computing $p(s \mid l)$ amounts to a fast series of two table look-ups (see [5] for more details).

(2) **Selective Computation.** Most of the time the probability mass is centered on a small number of location. With the exception of the initial global localization phase, the vast majority of probabilities $Bel(L = l)$ are usually close to 0 and can safely be ignored. This observation is the basis for a selective computation scheme, which enhances the computational speed of the algorithm. Our implementation only considers locations l for which $Bel(L = l)$ is above a certain threshold [2] (see [2] for more details).

[2] In our current implementation θ is set to 1% of the a priori position probability.

With these modifications, the time required for processing a sensor scan usually takes less than 0.2 seconds on an Intel Pentium-200 processor.

4 Active Markov Localization

Position probability grids have been shown to be able to robustly estimate the position of a robot in unstructured and populated environments [6,4,9,8]. However, Markov localization is passive since it provides no criteria on how to control the robot (e.g. in order to resolve ambiguities). In this section we will derive criteria on how to control the actuators of the robot so as to best localize a robot.

4.1 General Equations

To choose optimal actions we have to trade off the utility $U(a)$ and costs $C(a)$ of each individual action a.

4.1.1 Utility of Actions

In order to eliminate uncertainty in the position estimate $Bel(L)$, the robot must choose actions which help it distinguish different locations. The entropy of the belief, obtained by the following formula

$$H(L) = -\sum_{l} Bel(L = l) \, \log Bel(L = l), \qquad (3)$$

measures the uncertainty in the robot position: If $H(L) = 0$, $Bel(L)$ is centered on a single position, whereas the entropy is maximal, if the robot is completely uncertain and $Bel(L)$ is uniformly distributed.

Let $E_a[H(L_{t+1})]$ denote the expected entropy *after* having performed action a at time t and *after* having fired the sensors of the robot; then we can measure the utility $U_t(a)$ of performing an action a by the decrease in uncertainty:

$$U_t(a) = H(L_t) - E_a[H(L_{t+1})] \qquad (4)$$

By averaging over all possible sensings s, we obtain Eq. (5). Here $p(s \mid a)$ is the probability of perceiving sensing s after execution of action a.

$$\begin{aligned}
& E_a[H(L_{t+1})] \\
&= \sum_{s} H(L_{t+1} \mid s, a) p(s \mid a) && (5) \\
&= -\sum_{s,l} Bel(L_{t+1} = l \mid s, a) \cdot \\
&\qquad \log Bel(L_{t+1} = l \mid s, a) \; p(s \mid a) && (6) \\
&= -\sum_{s,l} p(s \mid l) \; Bel(L_{t+1} = l \mid a) \cdot \\
&\qquad \log \frac{p(s \mid l) Bel(L_{t+1} = l \mid a)}{p(s \mid a)} && (7)
\end{aligned}$$

Let $Bel(L_{t+1} = l \mid s, a)$ denote the belief of being at position l after having performed a and perceived s; then expression (6) is obtained from the definition of the entropy given in Eq. (3). By applying the Markov update equations (1) and (2) we finally get Eq. (7) for computing the expected entropy for action a. Here, $p(s \mid a)$ serves as a normalizer ensuring that $Bel(L_{t+1})$ sums up to one over all l.

4.1.2 Costs of Actions

To find the best action we have to trade off the utility of each action against the cost of executing the action. Costs $C_t(a)$ strongly depends on the particular actions and can range from the time needed to perform an action to the amount of energy used up by the action (see e.g. Section 4.2).

4.1.3 Action Selection

Given the utility and costs of actions, the robot chooses at any point t in time the action a^* that maximizes

$$a^* = \underset{a}{\operatorname{argmax}}(U_t(a) - \beta \cdot C_t(a)). \quad (8)$$

Here $\beta \geq 0$ determines the relative importance of certainty versus costs. The choice of β depends on the application. In our experiments, β was set to 1.

In the next two sections we will present applications of this general scheme to (1) actively navigating the robot and to (2) actively choosing the optimal sensing direction.

4.2 *Active Navigation*

Active navigation addresses the problem of determining where to move so as to best position the robot. At first glance, one might use simple motor control actions (such as "move 1 meter forward") as basic actions in active navigation. However, just looking at the immediate next motor command is often insufficient. For example, Figure 2 shows a typical situation occuring during global localization in our department. Here the robot was placed in the corridor and the figure gives the belief state $Bel(L)$ after some meters of random motion within this corridor (more likely positions are darker). The two local maxima stem from the symmetry of the corridor and in such a situation the robot has to move into one of the offices in order to uniquely determine its position.

We have chosen to consider arbitrary target points as atomic actions in active navigation. Target points are specified relative to the current robot location, not in absolute coordinates. For example, an action $a = move(-9\text{m}, -4\text{m})$ will make the robot move to a location 9 meters behind it and 4 meters to the left. Because actions are represented relative to the robot's position, the absolute position of such targets

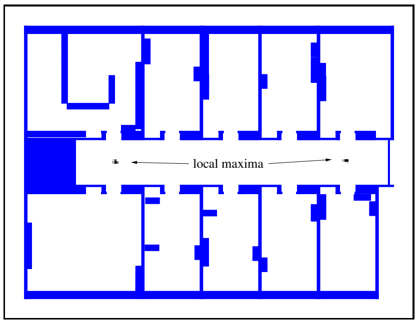

Fig. 2. Outline of the department along with position probabilities.

strongly depends on the actual belief of the position estimate. Figure 3 shows a situation where the belief is concentrated on the two positions marked by the big circles. The action $a = move(-9\text{m}, -4\text{m})$ might thus carry the robot to the location marked "1" or to the location marked "2".

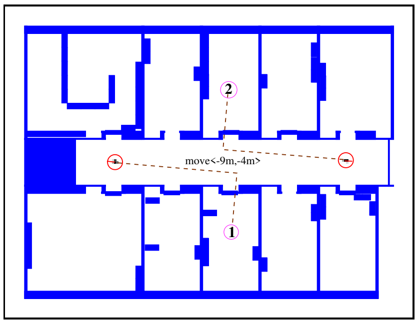

Fig. 3. Absolute positions of target points.

To see, let $f_a(l)$ be the coordinate transformation, which expresses the real-world coordinates of the target location of action a, assuming that the robot is at l. Further let a_f and a_s denote the forward and the sideward component of the movement action a, respectively. Then the three components of the target point of action a executed at location l are given by Eq. (9).

$$\begin{aligned} [f_a(l)]_x &= l_x + \cos l_\alpha a_f + \sin l_\alpha a_s \\ [f_a(l)]_y &= l_y + \sin l_\alpha a_f - \cos l_\alpha a_s \\ [f_a(l)]_\alpha &= l_\alpha + a_\alpha \end{aligned} \quad (9)$$

The remainder of this section specifies the computation of the utility and the costs of navigation actions.

4.2.1 *Utility of Navigation Actions*

In order to determine the utility of moving to a relative target location using Eq. (7) we have to specify the entropy expected upon executing a navigation action a. To compute $Bel(L_{t+1} \mid a)$ we apply the inverse of the coordinate transformation given in Eq. (9) and get the following equation (compare Eq. (7)):

$$E_a[H(L_{t+1})] = -\sum_{s,l} p(s \mid l)\; Bel(L_t = {f_a}^{-1}(l)) \cdot \log \frac{p(s \mid l) Bel(L_t = {f_a}^{-1}(l))}{p(s \mid a)} \quad (10)$$

4.2.2 *Costs of Navigation Actions*

To estimate $C(a)$ for an action a we estimate the expected costs on the cost-optimal path from the current location of the robot to the target location. To do so, our approach rests on the assumption that a map of the environment is available, which specifies which point l is occupied and which one is not. In our implementation the world model is given as an occupancy grid map.

Occupancy probabilities: Let $p_{occ}(l)$ denote the probability that location l is blocked by an obstacle. The robot has to compute the probability that a target point a is occupied. Recall that the robot does not know its exact location; thus, it must estimate the probability that a target point a is occupied. This probability will be denoted $p_{occ}(a)$. Geometric considerations permit the "translation" from $p_{occ}(l)$ (in real-world coordinates) to $p_{occ}(a)$ (in robot coordinates):

$$p_{occ}(a) = \sum_{l} Bel(L = l)\, p_{occ}(f_a(l)) \quad (11)$$

Here, $f_a(l)$ is the coordinate transformation introduced in Eq. (9). Thus, $p_{occ}(f_a(l))$ is the occupancy probability of the position $f_a(l)$ reached when executing action a at position l. The expected occupancy probability $p_{occ}(a)$ of the target point reached by action a is then obtained by averaging over all locations l, weighted by the robot's subjective belief $Bel(L = l)$ of being there. The result is the expected ocupancy of a point a relative to the robot.

Cost and cost-optimal paths: Based on $p_{occ}(a)$, the expected path length and the cost-optimal policy can be obtained through *value iteration*, a popular version of dynamic programming (see e.g., [16] for details). Value iteration assigns to each location a a *value* $v(a)$ that represents its distance to the robot. Initially, $v(a)$ is set to 0 for the location $a = (0, 0)$ (which is the robot's location), and ∞ for all other locations a. The value function $v(a)$ is then updated recursively according to the following rule:

$$v(a) \longleftarrow p_{occ}(a) + \min_b [v(b)] \quad (12)$$

Here $v(b)$ is minimized over all *neighbors* of a, i.e., all locations that can be reached from a with a single, atomic motor command. Eq. (12) assumes that the costs for traversing a point a is proportional to the probability that a is occupied ($p_{occ}(a)$). Iteratively applying this equation leads to the cost function $C(a)$ for reaching any point a relative to the robot, and hill climbing in v (starting at a) gives the cost-optimal path from the robot's current position to any location a.

This completes the description of active navigation with the purpose of localization. To summarize, actions represent arbitrary target points relative to the robot's current position. Actions are selected by maximizing a weighted sum of (1) expected decrease in uncertainty (entropy) and (2) costs of moving there. Costs are considered because they may vary drastically between different target points.

4.3 Active Sensing

By active sensing we attack the problem of where to point the robot's sensors so as to best localize the robot. Corresponding to active navigation this is realized by pointing the sensor into the direction which maximizes the expected utility.

4.3.1 Utility of Sensing Actions

To see, let $a_\alpha = point(\alpha)$ denote the action of pointing the sensor into the direction α relative to the robot's orientation. The expected entropy of such an action is given in Eq. (13) (compare Eq. (7)).

$$E_{a_\alpha}[H(L_{t+1})] = -\sum_{s_\alpha, l} p(s_\alpha \mid l)\ Bel(L_t = l) \cdot \log \frac{p(s_\alpha \mid l) Bel(L_t = l)}{p(s_\alpha \mid a)} \qquad (13)$$

Here s_α are only those sensings perceivable in the corresponding direction α. Please note that we replaced $Bel(L_{t+1} = l \mid a_\alpha)$ in Eq. (7) by $Bel(L_t = l)$ because pointing a sensor in a specific direction does not change the location of the robot.

In this work we assume that the costs for pointing a sensor do not depend on the pointing direction. Thus we can neglect costs during active sensing and select the pointing direction depending solely on the utility.

5 Experimental Results

In this section we test the influence of our active extension to Markov localization on the performance of the position estimation.

5.1 Efficient Implementation

The active navigation and sensing methods described here have been implemented and tested using *position probability grids* introduced in Section 3.2. In our implementation of active navigation the discretization of the possible actions is as small as the resolution of the applied position probability grid. The complexity of computing the utility of a single action is in $O(|L| \cdot |S|)$, where $|L|$ is the number of possible locations and $|S|$ is the number of possible sensings (compare Eq. (7)). This results in an overall complexity of $O(|L|^2 \cdot |S|)$ for computing the best action a according to Eq. (4). In order to make the computation of the utilities tractable we approximate L by a set L_m of m Gaussian densities with means $\mu_i \in L$. The equations presented in Section 4 are only applied to the means μ_i. This simplification is somewhat justified by the observation that in practice, $Bel(L)$ is usually quickly centered on a small number of hypotheses and approximately zero anywhere else [3]. The centers of the Gaussians μ_i are computed at runtime, by scanning the belief state for local maxima l whose probability $Bel(L = l)$ exceeds a certain threshold. While this modification often reduces the size of $|L|$ by more than four orders of magnitude we are convinced that the set of these local maxima represents the most important aspects of the belief state at a sufficient rich level. Together with the fast sensor model introduced in Section 3.2 action selection can be performed in reasonable time for active navigation and in real-time for active sensing (see next sections).

[3] Before this concentration of the belief state is established, the technique introduced here does not apply.

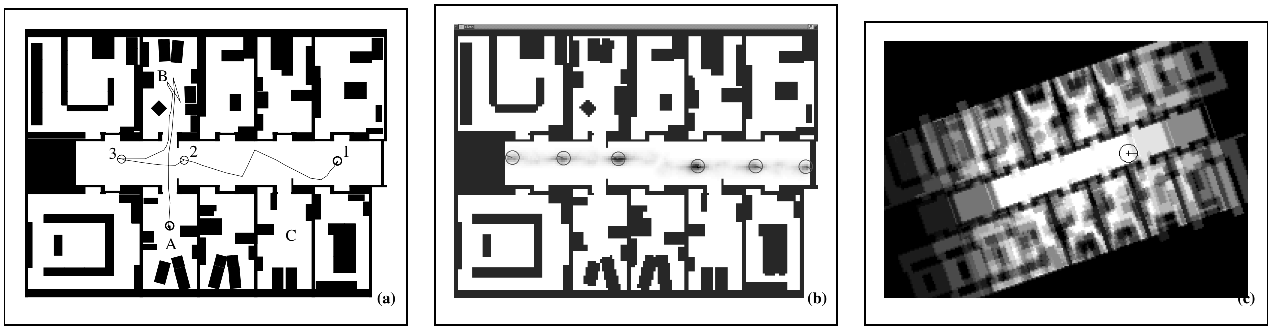

Fig. 4. (a) Environment and path of the robot. (b) Belief $Bel(L)$ and (c) occupancy probabilities $p_{occ}(a)$ at pos. 2.

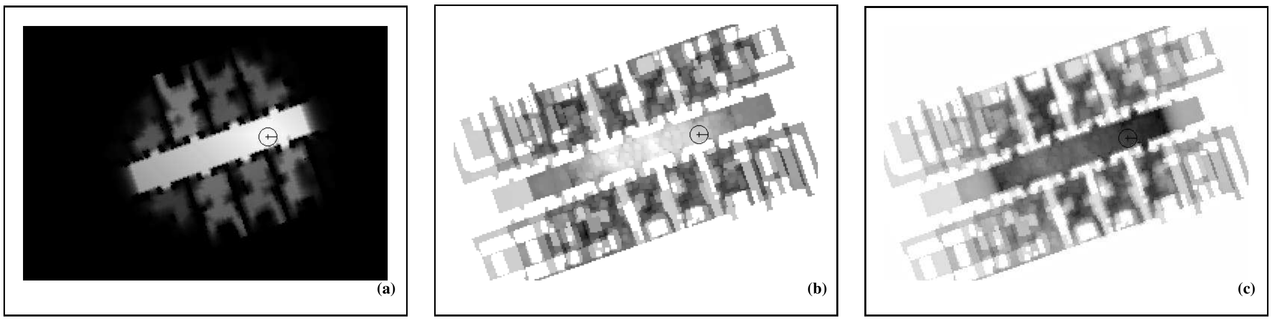

Fig. 5. (a) Expected costs $C(a)$, (b) utility $U(a)$, and (c) payoff $U(a) - \beta C(a)$ at pos. 2.

5.2 Active navigation

Active navigation was tested by placing the robot in the corridor of our department as plotted in Figure 4 (a). Notice that the corridor in this environment is basically symmetric and possesses various places that look alike, making it difficult for the robot to determine where it is. In this environment, the robot must move into one of the offices, since only here it finds distinguishing features due to the different furniture in the offices. The state of the doors has no influence on the position estimation itself and is only used for determining the expected costs of moving to the different target points. In this particular experiment, the robot is only able to uniquely localize itself by moving either into room B or C.

In a total of 10 additional experiments, random wandering consistently failed to localize the robot. Please note that wall following is less efficient than random wandering, since it prevents the robot from moving into rooms. A simple strategy might be to follow a wall until the next door is reached and to move into the corresponding room. Whereas this approach would lead to an efficient solution in this situation, it cannot be applied in arbitrary and possibly less structured environments. In more than 20 experiments using the active navigation approach presented here, the robot always managed to localize itself in a considerably short amount of time.

Figure 4 (a) also shows a representative example of the path taken during active exploration. In this particular run we started the robot at position 1 in the corridor facing south-west. The task of the robot was to determine its position within the environment and then to move into room A (so that we could see that localization was successful).

In order to deal with the computational complexity of active localization (see Section 5.1), the robot starts with random motion until the belief is concentrated on several local maxima. After about ten meters of such random motion, the robot reached position 2 (c.F. 4(a)). Figure 4 (b) depicts the belief $Bel(L)$ at this point in time (more likely positions are darker). The positions and orientations of the six local maxima considered for computing the utility and costs of actions are marked by the six circles.

The expected occupancy probabilities $p_{occ}(a)$, obtained by Eq. (11), are depicted in Figure 4 (c). High probabilities are shown in dark colors. Note that this figure roughly corresponds to a weighted overlay of the environmental map relative to the six local maxima, where the weights are given by the probabilities of these maxima. Figure 4 (c) also contains the origin of the corresponding coordinate system. This point represents the current position of the robot or the action $a = move(0\text{m}, 0\text{m})$, respectively (with robot facing right). Figure 5 (a) displays the expected costs for reaching the different target points. These costs have been computed using value iteration based on Eq. (12). Figure 5 (b) shows the utility of the target points, according to Eq. (4). As can be seen there, the expected decrease in uncertainty of locations in rooms is high, thus making them favorable for localization. The utility is also high, however, for the two ends of the corridor, since those can further reduce uncertainty. Based on the utility-cost trade-off depicted in Figure 5 (c), the robot now decides to first pick a target at the end of the corridor.

At this point it is important to notice that the exact trajectory from the current position to the target point cannot be computed off-line. This is due to unavoidable inaccuracies in the world model and to unforeseen obstacles in populated environments such as our office. These difficulties are increased if the position of the robot is not known, as is the case during localization. To overcome these problems the robot must be controlled by a reactive collision avoidance technique. In our implementation a global planning module uses dynamic programming as described in section 4.2 to generate a cost minimal path to the target location (see [20]). Intermediate target points on this path are presented to our reactive collision avoidance technique described in [7]. The collision avoidance then generates motion commands to safely guide the robot to these targets. An overview of the architecture of the navigation system is given in [22,4].

After having reached the end of the corridor (position 3 in Figure 4 (a)) the belief state contains only two local maxima (see Figure 6 (a)). The occupancy probabilities and the resulting costs of the different actions for this belief are depicted in Figure 6 (b) and (c), respectively. Please note that due to the state of the doors, the costs for reaching room B or C are remarkably lower than those for reaching the other rooms. The ambiguity in the belief displayed in Figure 6 (a) can no longer be resolved without leaving the corridor. Accordingly the utility shown in Figure 7 (a) is low for target points in the corridor compared to the utility of actions which guide the robot into the rooms. Because of the state of the doors and the resulting costs, the overall payoff as displayed in Figure 7 (b) is maximal for target points in rooms B and C.

As shown in Figure 4 (a) the robot decided to move into the room behind it on the right, which in this case turned out to be room B. Here the robot has been able to resolve the ambiguity between the rooms B and C based on the different furniture in the two rooms. After having uniquely determined its location the robot moved straight to the target location in room A. Figure 7 (c) shows the belief state at this point. Please note that only ultrasound sensors were used in these experiments.

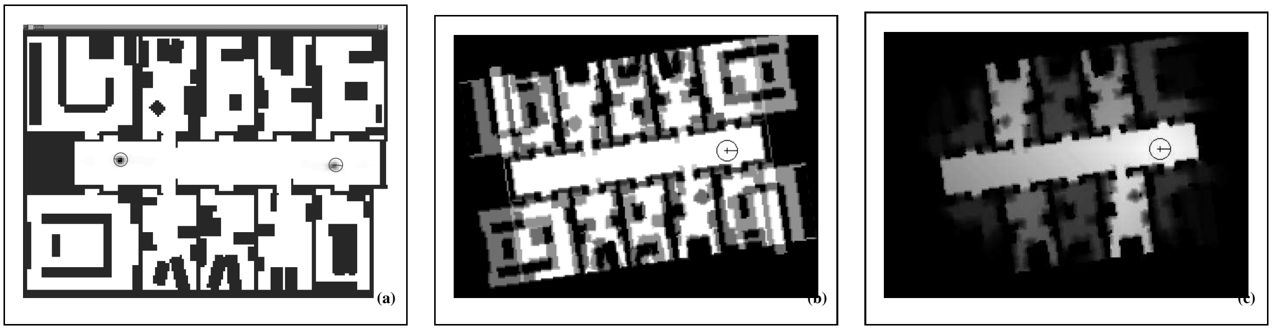

Fig. 6. (a) Belief $Bel(L)$ at pos. 3, (b) occupancy probabilities $p_{occ}(a)$, and (c) expected costs $C(a)$.

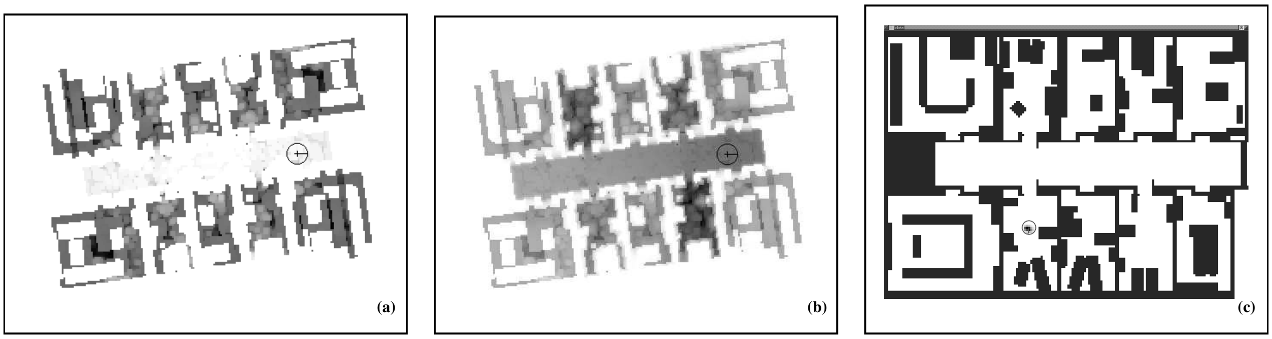

Fig. 7. (a) Utility $U(a)$, (b) payoff $U(a) - \beta C(a)$ at pos. 3, and (c) final belief.

5.3 Active Sensing

In the following experiments we demonstrate how the efficiency of localization can be improved by choosing the optimal pointing direction of the robot's sensors. Our experiments are conducted with two different sensors (ultrasound and laser-range finder) in order to demonstrate the ability of our technique to select the kind of sensor which is most appropriate in the given situation (e.g. a camera mounted on a pan head with two different zoom values). The difference between ultrasound sensors and laser-range finders lies in their accuracy. While ultrasound sensors have an angular resolution of $15°$, our laser-range finders have an angular resolution of $1°$. In addition to this, laser-range finders are able to measure obstacles more accurate than ultrasound sensors. Please note that in our approach these differences are represented solely in the model of the sensors, specifically in the probabilities $p(s \mid o)$ of measuring distance s if an obstacle is placed in distance o (c.F. 1).

Figure 8 shows the setup of the experiments: the robot was placed in the corridor of our department and moved up and down with a constant velocity of 30 cm/sec. Obviously, this corridor ($23 \times 4.5\, m^2$, all doors closed) is symmetric. In order to allow the robot to uniquely determine its location we installed a single box on one side of the corridor. This obstacle could only be detected by the robot's ultrasound sensors. Thus, to uniquely determine its location, the robot had to choose ultrasound sensors pointing towards this box.

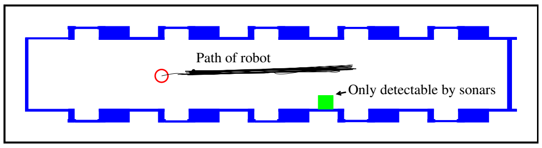

Fig. 8. Experimental setup.

To simulate active sensing, we allowed the robot to read only a single sensor at any point in time. As a passive method, we chose a sensor at ran-

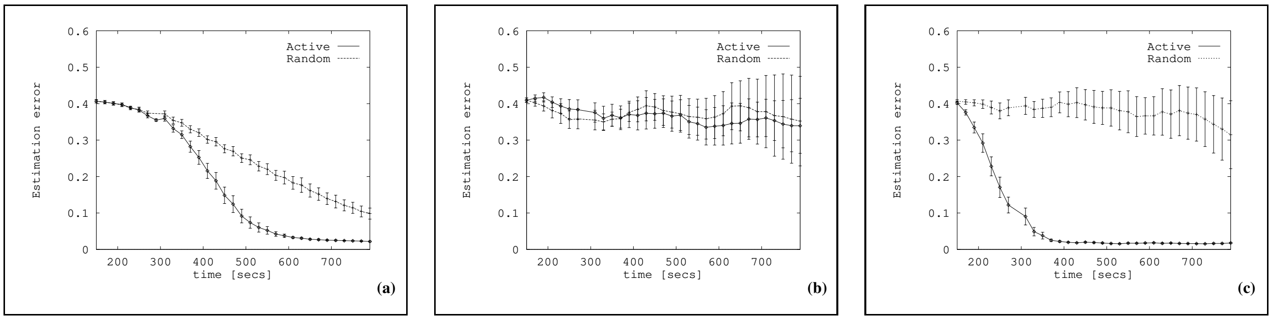

Fig. 9. Estimation error when using (a) only ultrasound, (b) only laser-range finder, and (c) both.

dom. This passive method was compared to our active approach, where sensors are according to Eq. (13). To evaluate the difference between the two approaches we compute the error in localization measured by the L_1 norm, weighted by $Bel(l)$. In the first experiment the robot was only allowed to choose among its 24 ultrasound sensors. The results are depicted in Figure 9 (a). This figure plots the localization error as a function of the time, averaged over 12 runs, along with their 95% confidence intervals (bars). Apparently, the error decreases significantly faster when sensors are selected actively (solid line). This result clearly demonstrates the benefit of active sensing.

Figure 9 (b) depicts the corresponding results for the laser-range finder. In this case active sensing is not superior to choosing a pointing direction randomly. Obviously none of the methods is able to correctly determine the position of the robot, which is due to the inability of the laser-range finder to detect features breaking the symmetry of the corridor. During this kind of experiments we always observed two local maxima in the position probability distribution, one representing the true location of the robot and one representing the position mirrored by $180°$.

In the final experiment the robot was allowed to choose among both, ultrasound sensors and laser-range finders. The result is depicted in Figure 9 (c). Here again, active sensing significantly improves the efficiency of the localization process. In addition to this, active sensing performs much better when choosing from both sensors (Figure 9 (c)) than when being restricted to only one kind of sensor (Figure 9 (a) and Figure 9 (b)). In order to estimate the certainty of the two approaches of being at the true location during this class of experiments we summed up the probabilities assigned to positions closer than 40 cm to the true location of the robot. These probabilities are depicted in Figure 10. As can be seen here active sensing is certain of being at the true location after less than 400 seconds. The random strategy on the other hand is not able to uniquely determine the position of the robot within the scope of the experiment.

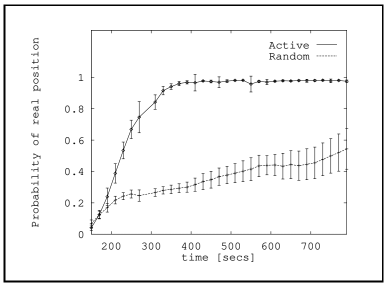

Fig. 10. Probability assigned to correct position.

To shed light onto the question as to why active localization performs significantly better than the passive method we analyzed the sensor readings that our active approach considered during localization. The points depicted in Figure 11 have been generated by storing the sensor measurements considered for localization and plotting their end points relative to the true position of the robot at that time.

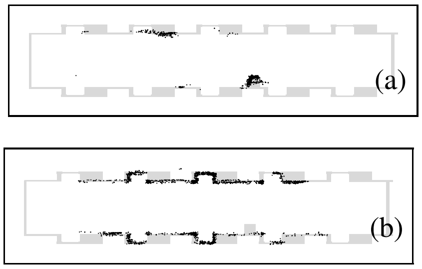

Fig. 11. End points of actively selected (a) ultrasound and (b) laser sensings.

Please note that the points in Figure 11 (a) and Figure 11 (b) stem from *one* experiment where the robot was allowed to choose among both sensors. Here the advantage of our method becomes obvious: the less accurate ultrasound sensors are only used to scan the box and its symmetric position (upper left cloud of points in Figure 11 (a)) in order to break the symmetry of the corridor. In most other cases the laser sensors were preferred due to their higher accuracy (c.F. 11(b)). This also explains why active sensing when using both sensors is significantly better than active sensing when using only one kind of sensor: by selecting the right kind of sensor our approach combines the accuracy of the laser-range finder with the ability of the ultrasound sensors to disambiguate the position estimation.

6 Conclusions

This paper advocates a new, active approach to mobile robot localization. In active localization, the robot controls its various effectors so as to most efficiently localize itself. In essence, actions are generated by maximizing the expected decrease of uncertainty, measured by entropy. This basic principle has been applied to two active localization problems: *active navigation*, and *active sensing*. In the case of active navigation, expected costs are incorporated into the action selection. Both approaches have been verified empirically using our RWI B21 mobile robot.

The key results of the experimental comparison are:

(1) The efficiency of localization is increased when actions are selected by minimizing entropy. This is the case for both active navigation and active sensing. In some cases, the active component enabled a robot to localize itself where the passive counterpart failed.
(2) The relative advantage of active localization is particularly large if the environment possesses relatively few features that enable a robot to unambiguously determine its location.

Despite these encouraging results, there are some limitations that deserve future research. One of the key limitations arises from the algorithmic complexity of the entropy prediction (compare Eq. (7)). While some algorithmic tricks made the computation of entropy feasible within the complexity bounds of our environment, more research is needed to scale the approach to environments that are significantly larger (e.g., 1000m$\times$1000m). A second limitation arises from the greediness of action selection. In principle, the problem of optimal exploration is NP hard, and there exist situations where greedy solutions will fail. However, in none of our experiments we ever observed that the robot was unable to localize itself using our greedy approach, something that quite frequently happened with the passive counterpart.

References

[1] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, 1982.

[2] M. Beetz, W. Burgard, D. Fox, and A.B. Cremers. Integrating active localization into

high-level robot control systems. *Robotics and Autonomous Systems, Special Issue for SIRS'97*, to appear.

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

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

[5] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proc. of the 21th German Conference on Artificial Intelligence, Germany*. Springer Verlag, 1997.

[6] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the Thirteenth National Conference on Artificial Intelligence*, 1996.

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

[8] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. A hybrid collision avoidance method for mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, Belgium, 1998. to appear.

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

[10] J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

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

[12] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. Technical report, Brown University, 1995.

[13] J. Kleinberg. The localization problem for mobile robots. In *Proc. of the 35th IEEE Symposium on Foundations of Computer Science*, 1994.

[14] S. Koenig. The complexity of real-time search. Technical Report CMU-CS-92-145, Carnegie Mellon University, April 1992.

[15] B. Kuipers and Y.T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8 1981.

[16] M.L. Littman, T.L. Dean, and L.P. Kaelbling. On the complexity of solving markov decision problems. In *Proc. of the Eleventh International Conference on Uncertainty in Artificial Intelligence*, 1995.

[17] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.

[18] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*, chapter 17. Number 0-13-103805-2 in Series in Artificial Intelligence. Prentice Hall, 1995.

[19] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence*, 1995.

[20] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Journal of Artificial Intelligence*. to appear.

[21] S. Thrun. The role of exploration in learning control. In D. A. White and D. A. Sofge,

editors, *Handbook of intelligent control: neural, fuzzy and adaptive approaches*. Van Nostrand Reinhold, Florence, Kentucky 41022, 1992.

[22] 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. MIT/AAAI Press, Cambridge, MA, 1998.