A Bayesian Approach to Landmark Discovery and Active Perception in Mobile Robot Navigation

Sebastian Thrun

May 1996

CMU-CS-96-122

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

To operate successfully in indoor environments, mobile robots must be able to localize themselves. Over the past few years, localization based on landmarks has become increasingly popular. Virtually all existing approaches to landmark-based navigation, however, rely on the human designer to decide what constitutes appropriate landmarks.

This paper presents an approach that enables mobile robots to select their landmarks by themselves. Landmarks are chosen based on their utility for localization. This is done by training neural network landmark detectors so as to minimize the a posteriori localization error that the robot is expected to make after querying its sensors. An empirical study illustrates that self-selected landmarks are superior to landmarks carefully selected by a human. The Bayesian approach is also applied to control the direction of the robot's camera, and empirical data demonstrates the appropriateness of this approach for active perception.

The author is also affiliated with the Computer Science Department III of the University of Bonn, Germany, where part of this research was carried out.

This research is sponsored in part by the National Science Foundation under award IRI-9313367, and by the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, and the Defense Advanced Research Projects Agency (DARPA) under grant number F33615-93-1-1330. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of NSF, Wright Laboratory or the United States Government.

Keywords: active perception, active vision, artificial neural networks, Bayesian analysis, exploration, landmarks, mobile robots, navigation, probabilistic navigation, sensor fusion

1 Introduction

For autonomous robots to operate successfully, they must know where they are. In recent years, landmark-based approaches have become popular for mobile robot localization. While the term "landmark" is not consistently defined in the literature, there seems to be a consensus that landmarks correspond to distinct spatial configurations of the environment which can be used as "reference points" for localization and navigation.

In recent years, landmark-based localization has been successfully employed in numerous mobile robot systems (see *e.g.*, [1, 4, 14, 18, 19, 25, 26, 31, 33, 35, 42]). A recent paper by Feng and colleagues [10] provides an excellent overview of different approaches to landmark-based localization. Many of the approaches reviewed there require special landmarks such as bar-code reflectors [9], reflecting tape, ultrasonic beacons, or visual patterns that are easy to recognize such as, *e.g.*, black rectangles with white dots [2]. Some of the more recent approaches use more natural landmarks for localization, which do not require special modifications of the environment. For example, landmarks in [19] correspond to certain gateways, doors and other vertical objects, detected with sonar sensors and pairs of camera images. Another approach [35] compiles multiple sonar scans into a local evidence grid [8, 23], from which geometric features such as different types of openings are extracted. The TRC HelpMate, which is one of the few commercially available service robots, uses ceiling lights as landmarks for localization [17]. Ceiling lights are stationary and easy to detect. In all these approaches, however, the landmarks themselves and the corresponding strategy for their recognition is prescribed by a human designer, and most of these systems rely on a narrow set of pre-defined landmarks.

A key open problem in landmark-based localization is the problem of automatically discovering good landmarks. Ideally, for landmarks to be as useful as possible, one wants them to be (1) stationary, (2) reliably recognizable, (3) sufficiently unique, and (4) there must be enough of them, so that they can be observed frequently. In addition, (5) landmarks should be well-suited for different types of localization problems, such as *initial self-localization*, which is the problem of guessing the initial robot location, and *position tracking*, which refers to the problem of compensating slippage and drift while the robot is moving. These problems, although related, often require different types of landmarks.

The problem of identifying landmarks is generally difficult and far from being solved. It is common practice that a human designer selects the landmarks. In some approaches, the human hard-codes a set of routines that can recognize whether or not a landmark is visible. In other approaches, supervised learning is employed to learn landmark recognizers—here the human designer provides the target labels for supervised learning. There are at least three shortcomings to both these approaches: First, selecting landmarks requires that the human is knowledgeable about the characteristics of the robot's sensors, and the environment in which the robot operates. As a consequence, it is often not straightforward to adjust a landmark-based system to new sensors, or new environments. Second, humans might be fooled by introspection. Since the human sensory apparatus differs from that of mobile robots, landmarks that appear to be appropriate for human orientation are not necessarily appropriate for robots. Finally, when the environment changes (*e.g.*, walls are painted in a different color, objects are moved, or the illumination changes), such static approaches to landmark recognition tend not to adjust well to new conditions, thus lead to suboptimal results or, in the extreme case, seize to work. Approaches that allow robots to automatically learn their landmarks are therefore preferable.

This paper presents an approach that allows a robot to select landmarks by itself, and to learn its own landmark recognizers. It does so by training a set of neural networks, each of which maps sensor input to a single value estimating the presence or absence of a particular landmark. In principle, the robot can choose any landmarks that can be recognized by neural networks. To "discover" landmarks, the networks are trained so as to minimize the average error in robot localization. More specifically, they are trained by minimizing the average a posteriori error in localization, which the robot is expected to make after it queries its sensors. As a result, the robot selects landmarks that are generally useful for localization (hence fulfill the criteria listed above).

The approach has been evaluated in an office environment, using a mobile robot equipped with sonar sensors and a color camera mounted on a pan/tilt unit. The key results of this paper can be summarized as follows:

1. The burden of selecting appropriate landmarks is eliminated.
2. Our approach consistently outperforms our current supervised learning approach, in which the human hand-selects landmarks and trains neural networks to recognize them.
3. If the robot is allowed to direct its camera (active perception), it can localize itself faster and more accurately than with a static camera configuration (passive perception).

The remainder of this paper is organized as follows. Section 2 introduces a general probabilistic model of robot motion and landmark-based localization which has been adopted from recent literature. Section 3 derives the landmark learning algorithm. Algorithms for active navigation and perception are described in Section 4, followed by an empirical evaluation of these algorithms using our mobile robot (Section 5). Finally, Section 6 summarizes the main results obtained in this paper and discusses open issues and future research.

2 A Probabilistic Model of Robot Localization

This sections lays out the groundwork for the landmark discovery approach presented in the next section. It provides a rigorous probabilistic account on robot motion, landmark recognition and localization.

In a nutshell, landmark-based localization works as follows:

1. In regular time intervals, the robot queries its sensors to check if one or more landmarks can be observed.
2. The result of these queries are used to refine the robot's internal believe as to where in the world it might be. The absence of a landmark is often as informative as its presence.
3. When the robot moves, its internal belief is updated accordingly. Since robot motion is inaccurate, it increases the robot's uncertainty.

Below, we will make three conditional independence assumption, which are essential for deriving an incremental update rule. These assumptions are equivalent to the assumption that the robot operates in a partially observable Markov environment [7], in which the only "state" is the location of the robot. The Markov assumption is commonly made in robot localization and navigation.

2.1 Robot Motion

Landmark-Based localization can best be described in probabilistic terms. Let l denote the location of the robot within a global reference frame. For mobile robots, l typically consists of the robot's x and y coordinates, along with its heading direction θ. While physically, a robot always has a unique location l at any point in time, internally it only has a belief concerning where it might be. This belief will be described by a probability density over all locations $l \in L$, denoted by

$$\hat{P}(l) \ . \tag{1}$$

Here L denotes the space of all locations. The problem of localization, phrased in general terms, is to approximate as closely as possible the "true" distribution of the robot location, which has a single peak at the "true" location and is zero elsewhere. Below, proximity will be defined as a weighted error.

Each motion command (*e.g.*, translation, rotation) changes the location of the robot. Expressed in probabilistic terms, a motion command $a \in A$ (A is the space of all motion commands) is described by a transition density

$$P_a(l \mid \tilde{l}). \tag{2}$$

P_a specifies the probability that the robot is l, given that it was previously at $\tilde{l}$ and that it just executed action a. If the robot would *not* use its sensors, it would gradually loose information as to where it is due to slippage and drift (*i.e.*, the entropy of $\hat{P}(l)$ would increase). Incorporating landmark information counteracts this effect, since landmarks convey information about the robot's location.

2.2 Landmarks

Suppose the robot is able to recognize n different landmarks. Each landmark detector maps a sensor measurement (*e.g.*, a sonar scan, a camera image) to a value in $\{0, 1\}$, depending on whether or not the robot believes that the i-th landmark is "visible." Obviously, for any sensible choice of landmark detectors, chances to observe a landmark f_i depend on the location l. Let

$$P(f_i|l) \tag{3}$$

denote the probability that the i-th landmark f_i is observed when the robot is at a location l. $P(f_i|l)$ is defined for all $f_i \in \{0,1\}$ and all $l \in L$. Although a landmark detector may be a deterministic function of the sensor input, $P(f_i|l)$ is generally non-deterministic, due to randomness (noise) in perception.

If the robot possesses n different landmark detectors, it observes n different values at any point in time, denoted by $(f_1, f_2, \dots, f_n) \in \{0,1\}^n$. Since each landmark detector outputs a binary value $f_i \in \{0,1\}$, there are (at most) 2^n such landmark vectors f. Assuming that different landmark detectors are conditionally independent[1], the total probability of observing $f \in \{0,1\}^n$ at l is the product of the marginal probabilities

$$P(f|l) \quad = \quad \prod_{i=1}^{n} P(f_i|l) \tag{4}$$

2.3 Robot Localization

The computational process of robot localization can now be formalized as follows. Initially, before consulting its sensors, the robot has some *prior belief* as to where it might be (uncertainty). This prior is denoted by $P(l)$. For example, in the absence of any more specific information, $P(l)$ may be distributed uniformly over all locations $l \in L$.

For reasons of simplicity, let us assume that at any point in time the robot executes an action a, senses, and, as a result, obtains a landmark vector f. Let $a^{(1)}, a^{(2)}, \cdots$ denotes the sequence of actions, and $f^{(1)}, f^{(2)}, \cdots$ the sequence of landmark vectors. The robot's belief after taking the t-th step is denoted by

$$P(l|f^{(1)} \cdots f^{(t)}, a^{(1)} \cdots a^{(t)}). \tag{5}$$

According to Bayes rule,

$$\begin{aligned} & P(l|f^{(1)} \cdots f^{(t)}, a^{(1)} \cdots a^{(t)}) \\ & = \frac{P(f^{(t)}|l, f^{(1)} \cdots f^{(t-1)}, a^{(1)} \cdots a^{(t)}) \; P(l|f^{(1)} \cdots f^{(t-1)}, a^{(1)} \cdots a^{(t)})}{P(f^{(t)}|f^{(1)} \cdots f^{(t-1)}, a^{(1)} \cdots a^{(t)})}. \end{aligned} \tag{6}$$

Assuming that given the true robot location l, the t-th landmark vector $f^{(t)}$ is independent of previous landmark vectors $f^{(1)} \cdots f^{(t-1)}$ and previous actions $a^{(1)} \cdots a^{(t-1)}$ (in other words: assuming independent noise in landmark recognition and robot motion—an assumption that follows directly from the Markov assumption), (6) can be simplified to yield the important formula [27]

$$P(l|f^{(1)} \cdots f^{(t)}, a^{(1)} \cdots a^{(t)}) \quad = \quad \frac{P(f^{(t)}|l) \; P(l|f^{(1)} \cdots f^{(t-1)}, a^{(1)} \cdots a^{(t)})}{P(f^{(t)}|f^{(1)} \cdots f^{(t-1)}, a^{(1)} \cdots a^{(t)})} \tag{7}$$

[1] More specifically, it is assumed that if one knows the location of the robot l, knowledge of $n-1$ landmark detectors does not allow to make any more accurate predictions of the outcome of the n-th, for any subset of $n-1$ landmarks. In other words, it is assumed that the noise in landmark recognition is independent.

The denominator on the right hand side of (7) is normalizer which ensures that the density integrates to 1. It is obtained as follows:

$$P(f^{(t)}|f^{(1)}\cdots f^{(t-1)},a^{(1)}\cdots a^{(t)})) \quad = \quad \int_L P(f^{(t)}|l)\;P(l|f^{(1)}\cdots f^{(t-1)},a^{(1)}\cdots a^{(t)})\;dl \tag{8}$$

For processing the t-th action, $a^{(t)}$, the transition density $P_a(l|\tilde{l})$ is used:

$$P(l|f^{(1)}\cdots f^{(t-1)},a^{(1)}\cdots a^{(t)}) \quad = \quad \int_L P_a(l|\tilde{l})\;P(\tilde{l}|f^{(1)}\cdots f^{(t-1)},a^{(1)}\cdots a^{(t-1)})\;d\tilde{l} \tag{9}$$

Put verbally, the probability of being at l is the probability of previously having been at $\tilde{l}$, multiplied by the probability that action $a^{(t)}$ would carry the robot to location l (and integrated over all "previous" locations $\tilde{l}$).

2.4 Incremental Algorithm

Notice that both density estimations (7) and (9) can be transformed into an *incremental* form. This follows from the fact that the density after the t-th observation (left hand side of (7)) is obtained from the density just before making that observation. Likewise, the density after performing action $a^{(t)}$ (left hand side of (9)) is directly obtained from the density just before executing $a^{(t)}$. The incremental nature of (7) and (9) allows us to state a compact algorithm for maintaining and updating the probability density of the robot location. To indicate the incremental nature of the belief density, the "current" belief will be denoted $\hat{P}(l)$.

1. Initialization: $\hat{P}(l) \longleftarrow P(l)$
2. For each observed landmark vector f do:

$$\hat{P}(l) \quad \longleftarrow \quad P(f|l)\;\hat{P}(l) \tag{10}$$

$$\hat{P}(l) \quad \longleftarrow \quad \hat{P}(l)\;\left[\int_L \hat{P}(l)\;dl\right]^{-1} \qquad \text{(normalization)} \tag{11}$$

3. For each robot motion a do:

$$\hat{P}(l) \quad \longleftarrow \quad \int_L P_a(l|\tilde{l})\;\hat{P}(\tilde{l})\;d\tilde{l} \tag{12}$$

This algorithmic scheme subsumes various probabilistic algorithms published in the recent literature on landmark-based localization and navigation (see *e.g.*, [4, 26, 35]). Notice that it requires knowledge about three probability densities: $P(l)$, $P_a(l|\tilde{l})$, and $P(f|l)$. Recall that the initial estimate $P(l)$ is usually the uniform probability distribution. The transition probability $P_a(l|\tilde{l})$ describes the effect of the robot's actions, and is assumed to be known (in practice it usually suffices to know a pessimistic approximation of $P_a(l|\tilde{l})$). The probability $P(f|l)$ is usually learned from examples, unless an exact model of the robot's environment and its sensors is available. $P(f|l)$ is often represented by a piecewise constant function [3, 4, 5, 18, 24, 26, 35, 38, 39], or a parameterized density such as a Gaussian or a mixture thereof [12, 30, 36, 37].

Figure 1: Landmark-based localization—an illustrative example.

Figure 1 gives a graphical example that illustrates landmark-based localization. Initially, the location of the robot is unknown—thus, $\hat{P}(l)$ is uniformly distributed (Figure 1a). The robot queries its sensors and finds out that it is next to a door. This information alone does not suffice to determines its position uniquely—partially because there might be a small chance that its landmark detectors are wrong, partially because there are multiple doors. As a result $\hat{P}(l)$ is large for door locations and small everywhere else (Figure 1b). Next, the robot moves forward, in response to which its density $\hat{P}(l)$ is shifted and slightly flattened, reflecting the uncertainty $P_a(l|\tilde{l})$ introduced by robot motion (Figure 1c). The robot now queries its sensors again, and finds out that again it is next to a door. The resulting density (Figure 1d) has now a single peak and is fairly accurate—the robot "knows" with high accuracy where it is.

2.5 Estimating a Single Location

In practice, it is often desirable to determine a unique estimate of the robot location, instead of an entire density $\hat{P}(l)$. For the sake of completeness, this section briefly describes two standard estimators, which

Figure 2: Maximum likelihood and Bayes estimator.

are commonly used in the statistical literature:

$$
\begin{aligned}
\text{maximum likelihood:} \qquad & l^* &=& \ \operatorname*{argmax}_{l} \hat{P}(l) \\
\text{Bayes estimator:} \qquad & l^* &=& \ \int_L l \ \hat{P}(l) \ dl
\end{aligned}
\qquad (13)
$$

The maximum likelihood estimator selects the location l^* which maximizes the likelihood $\hat{P}(l)$ (hence its name). If several locations tie, one is chosen at random. The Bayesian estimator, on the other hand, selects the location l that is best on average. In other words, it returns the location which minimizes the square deviation from the "true" location, if the latter is distributed according to $\hat{P}(l)$. Notice that the average error inferred by the maximum likelihood estimator is, in general, larger than that of the Bayesian estimator.

It is well-known that both estimators can be problematic, depending on the nature of the density $\hat{P}(l)$ [40]. In the situation depicted in Figure 2a, the maximum likelihood estimator would return the location of the spike on the left, since it is the most likely robot location, despite the fact that almost all probability mass is found on the right side of the diagram. In the situation depicted in Figure 2b, the Bayesian estimator would return the location between both spikes, which minimizes the average error, even though its likelihood might be zero. The approach described in this paper represents locations by entire probability densities.

This completes the derivation of a probabilistic framework to landmark-based localization. Of particular interest here is the assumption that the n landmark detectors are pre-wired. In the next section, we will drop this assumption and propose a novel approach that allows a robot to chose its own landmarks, by learning landmark detectors.

3 Learning Landmarks

This section describes the approach to landmark learning with artificial neural networks. The key idea is to select landmarks based on their utility for localization.

To do so, this section first derives a formula that measures the a posteriori localization error that a robot is expected to make when it is allowed to query its sensors. By minimizing this error with gradient descent in the parameter space of the landmark detectors (which, in the approach presented here, are realized by neural networks), the robot learns landmark detectors which are most informative for the task of localization. Notice that this approach does not rely on a human to determine appropriate landmarks. Instead, the robot chooses its own landmarks, through the process of minimizing the expected localization error. In an empirical evaluation, which follows this section, it will be demonstrated that this approach outperforms our current supervised learning approach, in which a human selects the landmarks and trains the neural networks in a supervised fashion.

3.1 The Average Error

Suppose the robot is at location l. After single sensor snapshot, the *Bayesian a posteriori error* (average localization error) is governed by

$$E(l) \;=\; \int_L \sum_{f=(0,\ldots,0)}^{(1,\ldots,1)} ||l-\hat{l}||\; P(f|l)\; P(\hat{l}|f)\; d\hat{l} \tag{14}$$

Here $||\cdot||$ denotes a norm[2] which measures the deviation of the "true" location l, and the estimated location $\hat{l}$. $P(f|l)$ measures the likelihood that the robot observes the landmark f at l, and $P(\hat{l}|f)$ denotes the likelihood with which the robot believes to be at $\hat{l}$ when observing f. $E(l)$ can be transformed using Bayes rule:

$$E(l) \;=\; \int_L \sum_{f=(0,\ldots,0)}^{(1,\ldots,1)} ||l-\hat{l}||\; P(f|l)\; P(f|\hat{l})\; \hat{P}(\hat{l})\; P(f)^{-1}\; d\hat{l} \tag{15}$$

Here $\hat{P}(\hat{l})$ is the *a priori uncertainty* in the location, which exists prior to querying the robot's sensors. If there were no uncertainty (*i.e.*, if $\hat{P}(\hat{l})$ was centered at a single location), there would be no localization problem, hence there would be no need to use landmark information.

$E(l)$ measures the expected error for a particular location l. Averaging $E(l)$ over all locations l yields the *Bayesian a posteriori localization error*, denoted by E:

$$E \;=\; \int_L E(l)\; P(l)\; dl \tag{16}$$

$$\stackrel{(15)}{=} \int_L \int_L \sum_{f=(0,\ldots,0)}^{(1,\ldots,1)} ||l-\hat{l}||\; P(f|l)\; P(f|\hat{l})\; P(l)\; \hat{P}(\hat{l})\; P(f)^{-1}\; d\hat{l}\; dl \tag{17}$$

Substituting $P(f|l)$ by $\prod_{i=1}^{n} P(f_i|l)$ (*cf.* Equation (4)) and re-ordering some of the terms yields:

[2] The L_1 norm was used throughout the experiments.

$$E = \int_L \int_L ||l - \hat{l}|| \, P(l) \, \hat{P}(\hat{l}) \, \cdot$$
$$\cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \left(\prod_{i=1}^{n} P(f_i|l) \, P(f_i|\hat{l}) \right) \, P(f)^{-1} \, d\hat{l} \, dl \tag{18}$$

The error E is central to the landmark learning approach. Notice that E contains the following terms, which are integrated over all "true" locations l, all believed locations $\hat{l}$, and all landmark vectors f:

1. The first term, $||l - \hat{l}||$, measures the error between the true and the believed location.
2. $P(l)$ reflects the a priori chances of the robot to be at location l. We will generally assume that all locations l are equally likely, *i.e.*, $P(l)$ is uniformly distributed.
3. $\hat{P}(\hat{l})$, specifies the a priori uncertainty in the location as discussed above.
4. $P(f_i|l)$, and $P(f_i|\hat{l})$ measures the probability of observing the i-th landmark at l, and $\hat{l}$, respectively.
5. Finally, $P(f)^{-1}$ is a normalizer which can be computed as follows:

$$P(f)^{-1} = \left[\int_L P(f|\tilde{l}) \, \hat{P}(\tilde{l}) \, d\tilde{l} \right]^{-1} = \left[\int_L \left(\prod_{i=1}^{n} P(f_i|\tilde{l}) \right) \, \hat{P}(\tilde{l}) \, d\tilde{l} \right]^{-1} \tag{19}$$

E enables the robot to compare different sets of landmark detectors with each other: The smaller E, the better the sets of landmark detectors. Hence, minimizing E is objective of the approach presented here. Notice that E (and hence the "optimal" landmark detectors, which minimize E) is a function of the uncertainty $\hat{P}(\hat{l})$. It therefore can happen that a set of landmark detectors which is optimal under one uncertainty performs poorly under another.

Notice that all densities in (18) are of the type $\hat{P}(\hat{l})$, $P(l)$, and $P(f_i|l)$. Expressions of the first two types are either priors, or, as discussed in the previous section, can be computed incrementally. Expressions of the sort $P(f_i|l)$ can be approximated based on data, which will be discussed in more detail below (Section 3.2).

3.2 Approximating E

The key idea of landmark discovery is to train neural networks to minimize E. The rationale behind this approach is straightforward: The smaller E, the more useful the landmark detectors for the task of localization.

However, while E measures the "true" Bayesian localization error, it can not be computed in any but the most trivial situations, basically because the probabilities $P(f_i|l)$ are unknown. However, it can be approximated with examples. More specifically, the robot is assumed to be given a set of examples

$$X = \{\langle l, s \rangle\}. \tag{20}$$

X consists of sensor measurements, denoted by s, which are labeled by the location l where the measurement was taken. Such examples are easy to obtain by driving the robot around and recording its location. Neural network landmark detectors will be denoted by

$$g_i : S \longrightarrow [0,1] \quad \text{for } i = 1, \ldots, n \;. \tag{21}$$

They map sensor measurements s (camera image, sonar scan) to landmark values in $[0, 1]$. Thus, the data set X can be used to provide samples that characterize the conditional probability $P(f_i|l)$. $\forall\{\langle l, s\rangle\} \in X$:

$$P(f_i|l) \;\approx\; \begin{cases} g_i(s) & \text{if } f_i = 1 \\ 1 - g_i(s) & \text{if } f_i = 0 \end{cases} \tag{22}$$

In other words, the output of the i-th network for an example $\{\langle l, s\rangle\} \in X$, $g_i(s)$, is interpreted as the probability that the i-th landmark is visible at location l.

We are now ready to approximate the error E (*cf.* (18) and (19)) based on the data set X:

$$\begin{aligned} \tilde{E} \;=\; & \sum_{\langle l,s\rangle \in X} \sum_{\langle \hat{l},\hat{s}\rangle \in X} ||l - \hat{l}||\; P(l)\; \hat{P}(\hat{l}) \cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \\ & \left(\prod_{i=1}^{n} P(f_i|l)\; P(f_i|\hat{l})\right) \underbrace{\left[\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \left(\prod_{i=1}^{n} P(f_i|\tilde{l})\right) \hat{P}(\tilde{l})\right]^{-1}}_{P(f)^{-1}} \end{aligned} \tag{23}$$

Equation (23) follows directly from (*cf.* (18) and (19)). Notice that $\tilde{E}$ converges to E as the size of the data set goes to infinity.

3.3 The Learning Algorithm

The neural network feature recognizers are trained with gradient descent to directly minimize $\tilde{E}$. This is done by iteratively adjusting the internal parameters of the i-th neural networks (*i.e.*, their *weights* and *biases*, denoted below by $w_{i\mu\nu}$, *cf.* [32]) in proportion to the negative gradients of $\tilde{E}$:

$$w_{i\mu\nu} \;\longleftarrow\; w_{i\mu\nu} - \eta\, \frac{\partial \tilde{E}}{\partial w_{i\mu\nu}} \tag{24}$$

Here $\eta > 0$ is a learning rate, which is commonly used in gradient descent. Computing the gradient (24) is a technical matter, as both $\tilde{E}$ and artificial neural networks are differentiable:

$$\frac{\partial \tilde{E}}{\partial w_{i\mu\nu}} \;=\; \sum_{\langle \bar{l},\bar{s}\rangle \in X} \frac{\partial E}{\partial g_i(\bar{s})}\, \frac{\partial g_i(\bar{s})}{\partial w_{i\mu\nu}} \tag{25}$$

The second gradient on the right hand side of Equation (25) is the regular output-weight gradient used in the Back-propagation algorithm, whose derivation is omitted here (see *e.g.*, [13, 32, 41] and most textbooks on neural network learning). The first gradient in (25) can be computed as follows:

$$\begin{aligned}
\frac{\partial \tilde{E}}{\partial g_i(\bar{s})} &\overset{(18)}{=} \sum_{\langle l,s\rangle \in X} \sum_{\langle \hat{l},\hat{s}\rangle \in X} ||l-\hat{l}||\; P(l)\; \hat{P}(\hat{l}) \; \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \\
&\qquad \frac{\partial}{\partial g_i(\bar{s})} \left[\left(\prod_{i=1}^{n} P(f_i|l)\; P(f_i|\hat{l}) \right) \; P(f)^{-1} \right] \\
&= \sum_{\langle l,s\rangle \in X} \sum_{\langle \hat{l},\hat{s}\rangle \in X} ||l-\hat{l}||\; P(l)\; \hat{P}(\hat{l}) \; \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \prod_{j\neq i} P(f_j|l)\; P(f_j|\hat{l}) \cdot \qquad (26) \\
&\qquad \cdot \left[\frac{\delta_{l,\bar{l}}\; P(f_i|\hat{l}) + \delta_{\hat{l},\bar{l}}\; P(f_i|l)}{\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \prod_{j=1}^{n} P(f_j|\tilde{l})} - \frac{P(f_i|l)\; P(f_i|\hat{l}) \prod_{j\neq i} P(f_j|\bar{l})\; \hat{P}(\bar{l})}{\left(\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \prod_{j=1}^{n} P(f_j|\tilde{l}) \right)^2} \right] (2\delta_{f_i,1} - 1)
\end{aligned}$$

Here $\delta_{x,y}$ denotes the Kronecker symbol, which is 1 if $x = y$ and 0 if $x \neq y$. $P(f_j|l)$ is computed according to Equation (22). Figure 3 shows the landmark learning algorithm, and summarizes the main formulas derived in this and the previous section. The gradient descent update is repeated until a termination criterion is reached (*e.g.*, early stopping using a cross-validation set, or pseudo-convergence of E), just like in regular Back-propagation [13].

To summarize, E is the expected localization error after observing a single sensor measurements. The neural network landmark detectors are trained so as to minimize E based on examples. Notice that this training scheme differs from supervised learning in that no target values are generated for the neural network landmark detectors. Instead, their characteristics emerge as a side-effect of minimizing E. Notice that E and thus the resulting landmark detectors depend on the uncertainty $\hat{P}(\hat{l})$. Below, when presenting some of our experimental results, it will be shown that in cases in which the margin of uncertainty is small, quite different landmarks will be selected than if the margin of uncertainty is large. However, while the landmark detectors have to be *trained* for a particular $\hat{P}(\hat{l})$, they can be *used* to estimate the location for arbitrary uncertainties. It is therefore helpful but not necessary to train different sets of landmark detectors for different a priori uncertainties.

1. **Initialization:** Initialize the parameters $w_{i\mu\nu}$ of each network with small random values.
2. **Iterate:**

2.1 $\forall \, \langle l, s \rangle \in X$: compute the conditional probabilities

$$P(f_i|l) \quad = \quad \begin{cases} g_i(s) & \text{if } f_i = 1 \\ 1 - g_i(s) & \text{if } f_i = 0 \end{cases} \tag{27}$$

where $g_i(s)$ is the output of the i-th network for input s (*cf.* (22)).

2.2 Compute the error $\tilde{E}$ (*cf.* (23))

$$\tilde{E} \quad = \quad \sum_{\langle l,s\rangle \in X} \; \sum_{\langle \hat{l},\hat{s}\rangle \in X} ||l - \hat{l}|| \; P(l) \; \hat{P}(\hat{l}) \; \cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \left(\prod_{i=1}^{n} P(f_i|l) \; P(f_i|\hat{l}) \right) \left[\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \left(\prod_{i=1}^{n} P(f_i|\tilde{l}) \right) \; \hat{P}(\tilde{l}) \right]^{-1} \tag{28}$$

2.3 $\forall$ network parameters $w_{i,\mu,\nu}$ compute

$$\begin{aligned} \frac{\partial \tilde{E}}{\partial w_{i\mu\nu}} \quad = \quad & \sum_{\langle \bar{l},\bar{s}\rangle \in X} \frac{\partial g_i(\bar{s})}{\partial w_{i\mu\nu}} \sum_{\langle l,s\rangle \in X} \; \sum_{\langle \hat{l},\hat{s}\rangle \in X} ||l - \hat{l}|| \; P(l) \; \hat{P}(\hat{l}) \; \cdot \\ & \cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \prod_{j \neq i} P(f_j|l) \; P(f_j|\hat{l}) \; (2\delta_{f_i,1} - 1) \; \cdot \\ & \cdot \left[\frac{\delta_{l,\bar{l}} \; P(f_i|\hat{l}) + \delta_{\hat{l},\bar{l}} \; P(f_i|l)}{\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \; \prod_{j=1}^{n} P(f_j|\tilde{l})} - \frac{P(f_i|l) \; P(f_i|\hat{l}) \; \prod_{j \neq i} P(f_j|\bar{l}) \; \hat{P}(\bar{l})}{\left(\sum_{\langle \tilde{l},\tilde{s}\rangle \in X} \; \prod_{j=1}^{n} P(f_j|\tilde{l}) \right)^2} \right] \end{aligned} \tag{29}$$

The gradients $\dfrac{\partial g_i(\bar{s})}{\partial w_{i\mu\nu}}$ are obtained with Back-propagation (*cf.* (25) and (27)).

2.4 $\forall$ network parameters $w_{i,\mu,\nu}$ update (*cf.* (24))

$$w_{i\mu\nu} \quad \longleftarrow \quad w_{i\mu\nu} - \eta \; \frac{\partial \tilde{E}}{\partial w_{i\mu\nu}} \tag{30}$$

Figure 3: The landmark learning algorithm.

4 Active Perception and Active Navigation

The expected a posteriori localization error E can also be used for controlling the robot's sensors and actions, so as to *actively* minimize the localization error. This section distinguishes two cases—active perception and active navigation—both of which rely on the same principle of greedily minimizing E.

4.1 Active Perception

To control the robots sensors, let us assume a (finite) set of different sensor configurations, denoted by $C = \{c_1, c_2, \ldots, c_m\}$. For example, a mobile robot might direct its camera to perceive different aspects of its environment (active vision). For simplicity, let us assume each sensor configuration has its own set of landmark recognizers. Then, the density $P(f_i|l)$, which measures the probability of observing a feature f_i at location l, is a function of the configuration $c \in C$. Henceforth, let us denote these densities by $P_c(f_i|l)$. The expected a posteriori localization error for configuration c is given by

$$
\begin{aligned}
E_c \;=\; & \int_L \int_L ||l - \hat{l}||\; \hat{P}(l)\; \hat{P}(\hat{l}) \;\cdot \\
& \cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \left(\prod_{i=1}^{n} P_c(f_i|l)\; P_c(f_i|\hat{l}) \right) \; P_c(f)^{-1}\; d\hat{l}\; dl\;, \qquad (31)
\end{aligned}
$$

with

$$
P_c(f)^{-1} \;=\; \left[\int_L P_c(f|\tilde{l})\; \hat{P}(\tilde{l})\; d\tilde{l} \right]^{-1} \;=\; \left[\int_L \left(\prod_{i=1}^{n} P_c(f_i|\tilde{l}) \right) \; \hat{P}(\tilde{l})\; d\tilde{l} \right]^{-1} \qquad (32)
$$

Both these equations are equivalent to those given in (18) and (19), except that the conditional densities $P(f_i|l)$ are now indexed by the subscript c. Notice that $\hat{P}(\cdot)$ in (31) denotes the "actual" uncertainty of the robot (as defined in Section 2.4). A greedy approach to active perception would be to select c so as to minimize E_c:

$$
c^* \;=\; \underset{c \in C}{\operatorname{argmax}}\; E_c \qquad (33)
$$

In the unlikely event that multiple sensor configurations tie, one is chosen at random.

By controlling the robot's sensors through minimizing E_c, the robot always directs its sensors so that the next sensor input is expected to be most informative, *i.e.*, is expected to reduce the a posteriori localization error the most. The approach is greedy, since it investigates only a single sensor measurement, instead of the entire sequence of measurements.

Notice that by making c an explicit input of each feature detector network g_i, it is possible to extend this scheme to infinitely many sensor configurations.

4.2 Active Navigation

Active navigation follows the same principle as active perception. In a nutshell, the robot selects its motion commands so that it minimizes the expected localization error E. The derivation of the control equation is straightforward. In active navigation, the internal belief upon executing action a is obtained by updating $\hat{P}(l)$ (*cf.* (12)):

$$\hat{P}(l) \quad = \quad \int_L P_a(l|\tilde{l})\ \hat{P}(\tilde{l})\ d\tilde{l} \tag{34}$$

Hence, the error E_a with

$$\begin{aligned} E_a \quad = \quad & \int_L \int_L \ ||l - \hat{l}|| \ \int_L P_a(l|\tilde{l})\ \hat{P}(\tilde{l})\ P_a(\hat{l}|\tilde{l})\ \hat{P}(\tilde{l})\ d\tilde{l}\ \cdot \\ & \cdot \sum_{f_1=0}^{1} \sum_{f_2=0}^{1} \cdots \sum_{f_n=0}^{1} \left(\prod_{i=1}^{n} P_c(f_i|l)\ P_c(f_i|\hat{l}) \right)\ P_c(f)^{-1}\ d\hat{l}\ dl\ , \end{aligned} \tag{35}$$

measures the expected a posteriori localization error, which is expected to be made after executing action a and taking a single sensor measurement. The motion direction that is greedily optimal for localization is then obtained by minimizing E_a:

$$a^* \quad = \quad \underset{a \in A}{\text{argmax}}\ E_a \tag{36}$$

5 Results

This section describes the main empirical results obtained with the landmark learning approach advocated in this paper. All results were obtained using the mobile robot AMELIA shown in Figure 4. The two primary results of our empirical study are:

1. Self-selected landmarks allows the robot to localize itself more accurately than human-selected landmarks, if the latter ones are trained using regular supervised learning.

2. Our approach to active perception, in which the robot is allowed to control the direction of its camera, is superior to passive perception.

This section also characterizes the impact of the uncertainty assumption on the landmark selection, and the interplay of multiple landmark networks that are trained simultaneously.

5.1 Experimental Setup

5.1.1 Testbed

The AMELIA robot (Figure 4) is equipped with a color camera mounted on a pan/tilt unit on top of the robot, and a circular array of 24 sonar proximity sensors. Sonar sensor return approximate echo distances to nearby obstacles, along with noise.

Figure 5a depicts a hand-drawn map of our testing environment. The environment contains two windows (at both corners), various doors, an elevator, a few trash-bins, and several walkways. Data was collected in multiple episodes (runs). To simplify the data collection, each run begun at a designated start location (point (A) in Figure 5a), and was terminated when the robot reached the foyer (point (H)). During each run the robot moved autonomously with approximately 15 cm/sec, controlled by its local obstacle avoidance routine [11, 34]. Figure 5b shows the path taken in three runs, along with an occupancy map constructed using the techniques described in [23, 39]. The length of each path is approximately 89 meters.

Generally speaking, the kinematic configuration of the robot is three-dimensional (it is often expressed by two Cartesian coordinates x and y, and the heading direction θ). Notice, however, that in our testbed the robot is not free to move arbitrarily in this three-dimensional space—instead it is forced to follow a narrow corridor. Simplified speaking, the robot moves on a one-dimensional manifold in its configuration space. Consequently, in our experiments the location of the robot was modeled by a single (one-dimensional) value, l, which measured the distance of the current location to the starting point. Data was collected automatically. When collecting the data, locations l were measured by cumulative dead-reckoning; no additional effort was made to correct for errors in the odometry of the robot. The reader may notice that the one-dimensional representation of l has two practical advantages over the more general, three-dimensional representation: It decreases the computational complexity of the algorithm considerably, and it reduces the amount of data necessary for successful learning. However, representing locations with a single value injects additional (non-Markovian) "noise" into the localization, since in practice the robot does not follow the exact same trajectory, so that multiple configurations in the true configuration space are projected onto a single value.

Figure 4: AMELIA, the robot used in our research.

5.1.2 Data and Representations

Data was collected in a total of twelve runs, with three different camera configurations. In four episodes the camera was pointed towards the left (denoted by c_{left}), in four additional episodes the camera was pointed up (denoted by c_{up}), and in the remaining four episodes the camera was pointed towards the right of the robot (denoted by c_{right}):

camera configuration	c_{left}	c_{up}	c_{right}
pan angle	45° left	straight ahead	45° right
tilt angle	straight	30° up	straight
number snapshots	3,110	3,473	3,232

Example images and sonar scans are shown in Figure 6. The letters labeling each row correspond to the marked locations in Figure 5a. Sonar scans are shown in the left column. Here the circle in the center depicts the robot from a bird's eye perspective. Each of the 24 cones surrounding the robot visualizes the distance to the nearest obstacle, measured by a single sonar sensor. The three cameras images in each row are camera images correspond to the different camera configurations c_{left}, c_{up}, and c_{right}.

To compensate some of the daytime- and view-dependent variations, images were pre-processed by normalizing the pixel mean and the variance within each image. Subsequently, each image was subdivided into ten equally-sized rows, and ten equally-sized columns. For each of these rows and columns, the following seven characteristic image features were computed:

- **average brightness**,

Figure 5: (a) Wean Hall, and (b) three of the twelve runs used in this study, along with an occupancy grid map constructed from sonar scans. The letters in (a) indicate where the example images were taken.

- **average color** (separate values for each of the three color channels), and
- **texture information:** the average absolute difference of the RGB/values of any two adjacent pixels (in a sub-sampled image of size 60 by 64, computed separately for each color channel).

In addition, 24 sonar measurements were provided, resulting in a total of $7 \cdot 20{+}24{=}164$ sensory features that were used as input values for the landmark detector networks. During the course of this research, we experimented with a variety of different image encodings, none of which appeared to have a significant impact on the quality of the results. Examples of image encodings (shown for rightmost image only) are depicted in the right column of Figure 6.

Figure 6: Examples of sonar scans, images (looking left, up and right) and image encodings.

5.1.3 Training

In all our experiments, layered multi-layer perceptrons with sigmoidal activation functions were used to detect landmarks [32]. These networks contained 164 input units, 6 hidden units, and one output unit. No effort was made to optimize the network structure.

The landmark learning algorithm summarized in Figure 3 is the exact gradient descent update algorithm. However, computing the gradient (Equation (27)) is computationally expensive. To keep the training times manageable even with large training sets, a modified training scheme was employed, which iterated the following four steps:

1. First, the network outputs $g_i(s)$ were computed for each training example $\langle s, l \rangle \in X$.
2. Subsequently, the gradients of $\tilde{E}$ with respect to the network outputs $g_i(s)$ were computed (*cf.* (27)).
3. The gradients were used to generate "pseudo-patterns" for each training example $\langle s, l \rangle \in X$:

$$\left\langle \, s \; , \;\; g_i(s) - \frac{\tilde{E}}{g_i(s)} \, \right\rangle \tag{37}$$

4. These patterns were approximated using 100 epochs of regular Back-Propagation, using a learning rate of 0.0001, a momentum of 0.9 and an approximate version of conjugate gradient descent [13].

This algorithm approximates gradient descent. It differs from gradient descent in that the exact gradient of $\tilde{E}$ is only computed occasionally, *i.e.*, every 100 training epochs. The advantage of this algorithm is its speed: Approximately 90% of the computational time is spent in the second step of the algorithm, whereas the Back-Propagation refinement requires less than 10%. Using this algorithm, typical training times on a SUN Ultra-Sparc were between 2 hours (small uncertainty, one network) and 4 days (global uncertainty, 4 networks). Notice that training time could have been reduced further by approximating the gradient, using only parts of the training set (on-line learning, or stochastic gradient descent [13]). As documented below, we did not observe any significant over-fitting, in none of our experiments. We attribute this to the fact that data is plentiful. Thus, instead of using cross-validation to determine the stopping time, that training was terminated after a fixed number of training epochs.

5.1.4 Testing

Unless otherwise noted, all results provided in this section were obtained for the third camera configuration (c_{right}), basically because these four runs were recorded first. In all experiments, two of these four runs were used for training the landmark networks, and the two remaining ones were used for evaluation. In an effort to evaluate the a posteriori localization error for a particular set of landmark detectors properly, one of the two evaluation runs was used to provide the "current snapshot" (expression $\langle l, s \rangle$ in Equation (23)). The other evaluation run provided the "reference labels" for estimating location (expression $\langle \hat{l}, \hat{s} \rangle$ in Equation (23)). This separation of the evaluation data is of fundamental importance, because subsequent snapshots within in a single run are usually similar, and thus may not be independent.

Notice that the landmark learning algorithm optimizes networks for a specific a priori uncertainty. In all our experiments, we only report results obtained with different *uniform* uncertainties, with varying width (Gaussian uncertainties give very similar results). Notice that the uncertainty in training does not necessarily have to be the same as in evaluation. We will refer to the uncertainty used in training as the "training uncertainty," and the one used in the evaluation as "testing uncertainty." When evaluating the trained landmark detectors, sometimes different a priori uncertainties are used, to investigate the robustness of the approach.

As noticed in Section 3.2, general probability densities cannot be represented on digital computers. In our experiments, they are approximated discretely. The approximation scheme used here directly follows from the approximation described in Section 3.2, Equation (23): $\hat{P}$ is calculated only for data $\langle l, s \rangle \in X$, where X is the evaluation set that provides the location labels. Such an approximation provides the highest resolution possible given the data—investigating more compact representations is beyond the scope of this paper.

Several diagrams in this paper show the output of landmark networks separately for the four different runs after training (*cf.* Figures 7, 9, 12, 13, and 14). Every diagram consists of for graphs, each of which corresponds to a particular data set. The top two graphs correspond to the evaluation sets (current snapshot and reference label), and the bottom two graphs to the training set. Each of the sub-graphs depicts the output of one (or more) neural networks for snapshots taken at different locations l. The black lines underneath each graph indicate the exact location at which the snapshots were taken. As can be seen by the spacing of these lines, the time required for each snapshot varied due to delays in the Ethernet transmission.

Other diagrams (Figures 8, 10, 11, and 15) show the results of evaluating a particular set of landmark networks. Unfortunately, the absolute a posteriori localization error $\tilde{E}$ depends crucially on the prior uncertainty $\hat{P}$, so that different absolute errors are barely comparable. To make these results comparable with each other, we will exclusively show the relative error ratio before and after sensing. More specifically, "performance" in the context of localization is defined as the quotient

$$1 \quad - \quad \frac{\text{a posteriori localization error}}{\text{prior localization error}} \tag{38}$$

which is typically measured in percent. Unless explicitly stated, all performance results reported here were obtained using the evaluation sets, following the testing methodology described above.

5.2 Human-Selected Landmarks and Supervised Learning

To compare the approach presented in this paper to other methods to landmark navigation in which a human expert hand-selects the landmarks, we first trained a landmark network in a supervised manner. To do so, we manually labeled the training sets by whether or not the image contained a significant fraction of a door. Doors, which frequently are visible when the cameras points towards the right (*i.e.*, configuration c_{right}), appear to be natural landmarks that are particularly well-suited for the fine-grained localization of mobile robots. In fact, in previous research carried out in our lab, doors were used as the sole visual landmarks for localization in the same environment, since they were assumed to be the most helpful landmarks (doors are considerably easy to recognize and stationary, and they play an important role in human orientation).

Figure 7 shows the output of the network after training. The network was trained on the two bottom datasets. Here the network almost perfectly approximated the target label in the training set. The dataset in the top row was used for testing the localization accuracy, using location labels provided by the run exhibited in the second row. As can be seen from Figure 7, the neural network landmark detector sharply

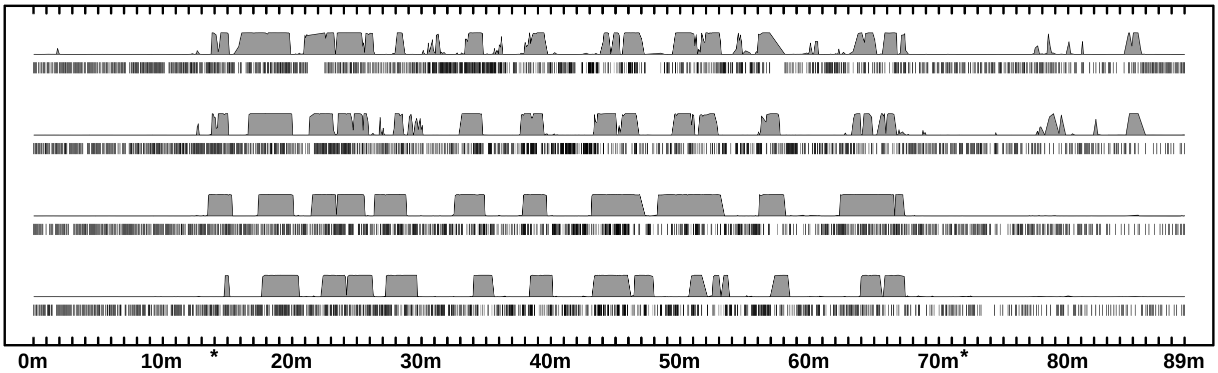


Figure 7: Supervised learning: network output and training patterns. See text.

discriminates between door and non-door sensor scans. The differences between different runs are due to variations in the sensor values, caused by errors in dead-reckoning, changes in the environment (such as people that sometimes appeared in the field of view), and the projection of the three-dimensional kinematic configuration to a one-dimensional manifold.

The utility of this landmark detection network for localization was measured using the two evaluation sets, following the methodology described above. Figure 8 depicts the empirical estimation results, averaged over 826 locations (*i.e.*, every location in the testing run), and for uniform uncertainty priors with different widths. As can be seen there, a single sensor snapshot reduces the localization error by an average of 4.35% if the a priori uncertainty (before querying the sensors) is uniformly distributed in $[-1\text{m}, 1\text{m}]$ (leftmost bar). If the a priori uncertainty is uniformly distributed in $[-2\text{m}, 2\text{m}]$, the reduction is almost twice as large: 8.34%. For uncertainties with larger entropy, the supervised landmark detector becomes less useful. In the extreme, where the a priori location is completely unknown and, thus, the uncertainty is globally uniformly distributed (rightmost bar), a single sensor snapshot reduces the a posteriori localization error by only 2.16%. This comes at little surprise, since information concerning the visibility of a door is not particularly helpful if the location of the robot is globally unknown (and the robot is only allowed to take a single snapshot).

5.3 Self-Selected Landmarks

Figure 9 depicts the output of the landmark detector network trained with the approach advocated in this paper. Each of the three diagrams in Figure 9 displays the results obtained for a different (uniform) training uncertainty $\hat{P}$: (a) uniform in $[-2\text{m}, 2\text{m}]$, (b) uniform in $[-10\text{m}, 10\text{m}]$, and (c) globally uniform. These results clearly illustrate the dependence of self-selected landmarks on the training uncertainty: In the top diagram, where the a priori uncertainty in the robot's location is considerably small (uniform in $[-2\text{m}, 2\text{m}]$), the output of the landmark detector changes with high frequency as the robot travels down the hallway. Some of the landmarks selected here correspond to doors, others to darker regions in the hallway and/or openings in the wall. For larger margins of uncertainty (Figure 9b&c), the robot selects different, more global landmarks, *i.e.*, the output of the network changes less frequently. In the extreme case of global uncertainty (Figure 9c), the only landmark selected by the robot is (the absence of) an orange wall, which characterizes the first 14 meters of each run until the robot makes its first left turn. These findings illustrate the first key result of the empirical study: The landmarks selected by the robot depend on the uncertainty distribution for which they were trained—there is no such thing as a uniquely

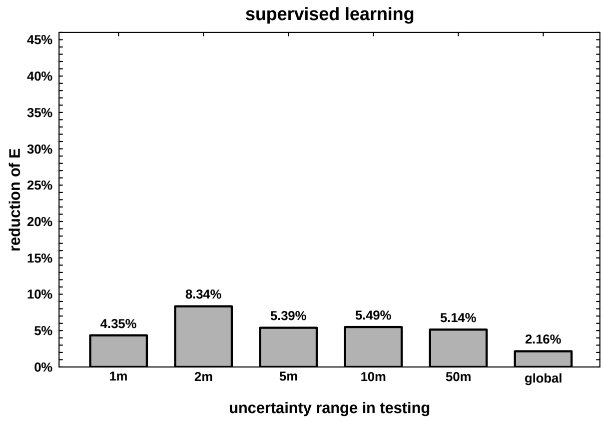


Figure 8: Performance results for supervised learning.

best landmark.

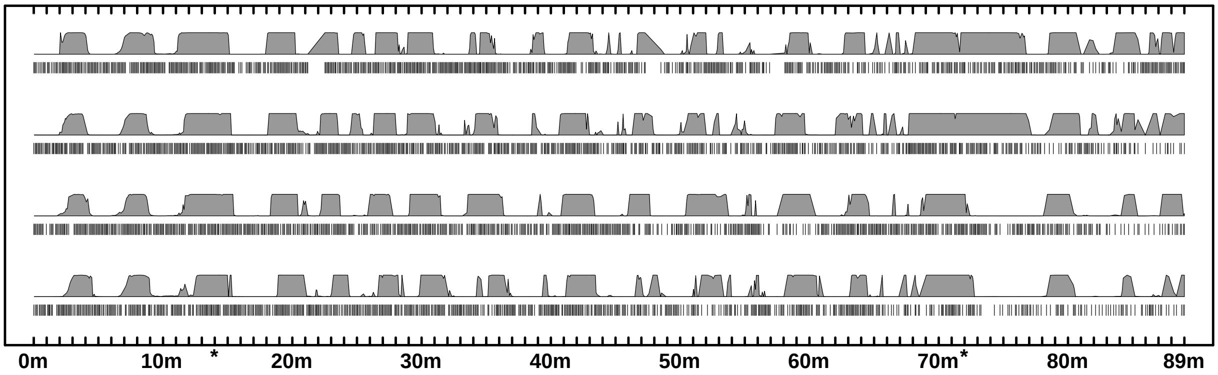
To characterize the appropriateness of the different landmark detectors for localization, we computed empirically the reduction of uncertainty for an independent test set, using the exact same data and following the same procedure as in the evaluation of the supervised approach. Figure 10 depicts training curves and average results for the three different networks discussed above. Figure 10a1, for example, shows the error reduction of the network trained on the uniform prior $[-2\text{m}, 2\text{m}]$ as a function of the number of training iterations (*cf.* Figure 9a). The bold curve shows the reduction of the localization error evaluated on the training set. The dashed line shows the same quantity, measured on the independent evaluation sets, using the same uncertainty prior as for training. As can be seen from this curve, the final error reduction, after 150 training iterations, is 14.9%. The other curves in Figure 10a1 depict the average error reduction for different uncertainty distributions. For example, when the network is tested under an uncertainty uniform in $[-1\text{m}, 1\text{m}]$ (notice that it is *trained* for uniform uncertainty in $[-2\text{m}, 2\text{m}]$), the final error reduction after 150 training iterations is only approximately 6.65%. This is because this network has been optimized for a different uncertainty. Notice that there is no noticeable over-fitting effect during training.

Figure 10a2 surveys the final performance results after training, taken from Figure 10a1. All bars shown here were obtained using the independent evaluation sets—the performance on the training set is omitted here. Figures 10b1 and 10b2 show the same results for the network trained under uniform uncertainty in $[-10\text{m}, 10\text{m}]$, and Figures 10c1 and 10c2 show the results obtained for the network trained under globally uniform uncertainty. These results, too, confirm the second key result of the empirical evaluation: Each network performs best under the uncertainty it was trained for. However, when applied under different uncertainties, the networks still manage to reduce the error.

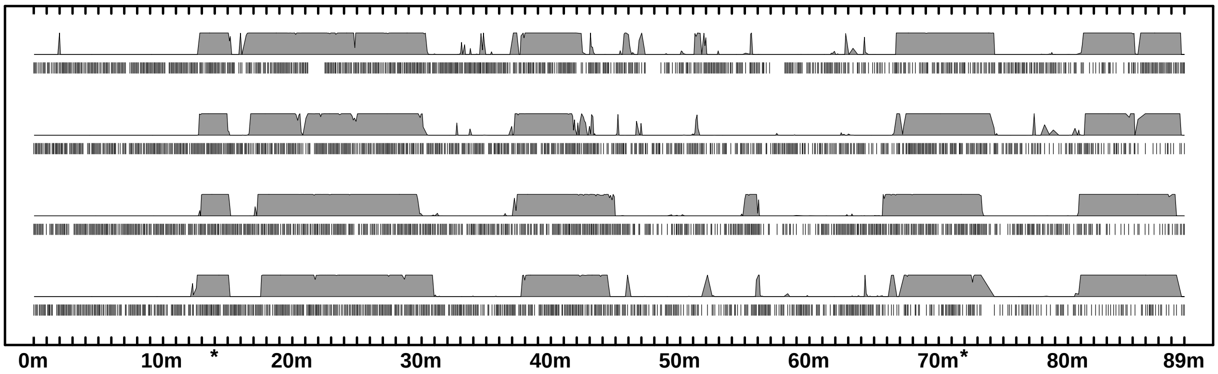
5.4 Comparison

When comparing the human-selected landmarks with the ones that were selected automatically, one notices commonalities and differences. Some of the landmarks in Figure 9a (this network appears to be most similar to the networks trained with supervised learning) indeed correspond to doors. However, closer examination of the output characteristics unveils that due to unevenly spaced floor lights, our

(a) uncertainty: $[-2\text{m}, 2\text{m}]$



(b) uncertainty: $[-10\text{m}, 10\text{m}]$



(c) uncertainty: uniform

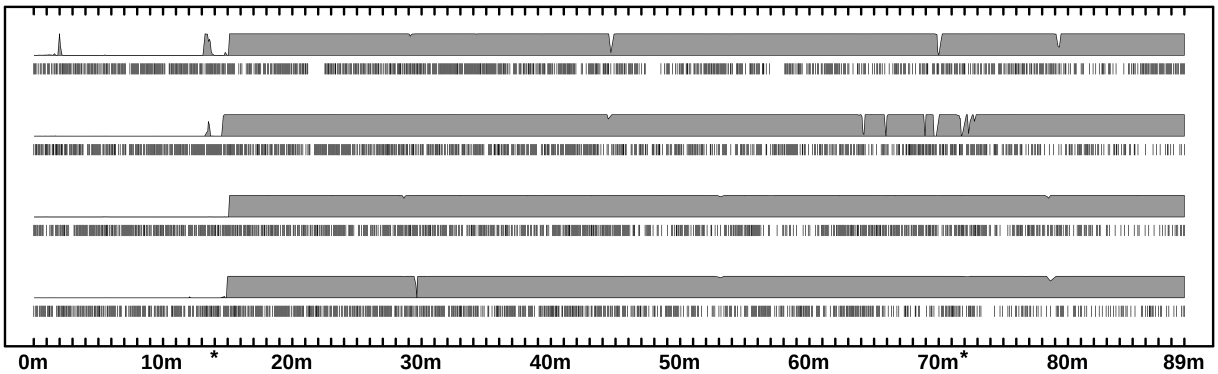


Figure 9: Landmarks selected by the robot, for $n = 1$ network and different uncertainty distributions $\hat{P}(\hat{l})$ (priors).

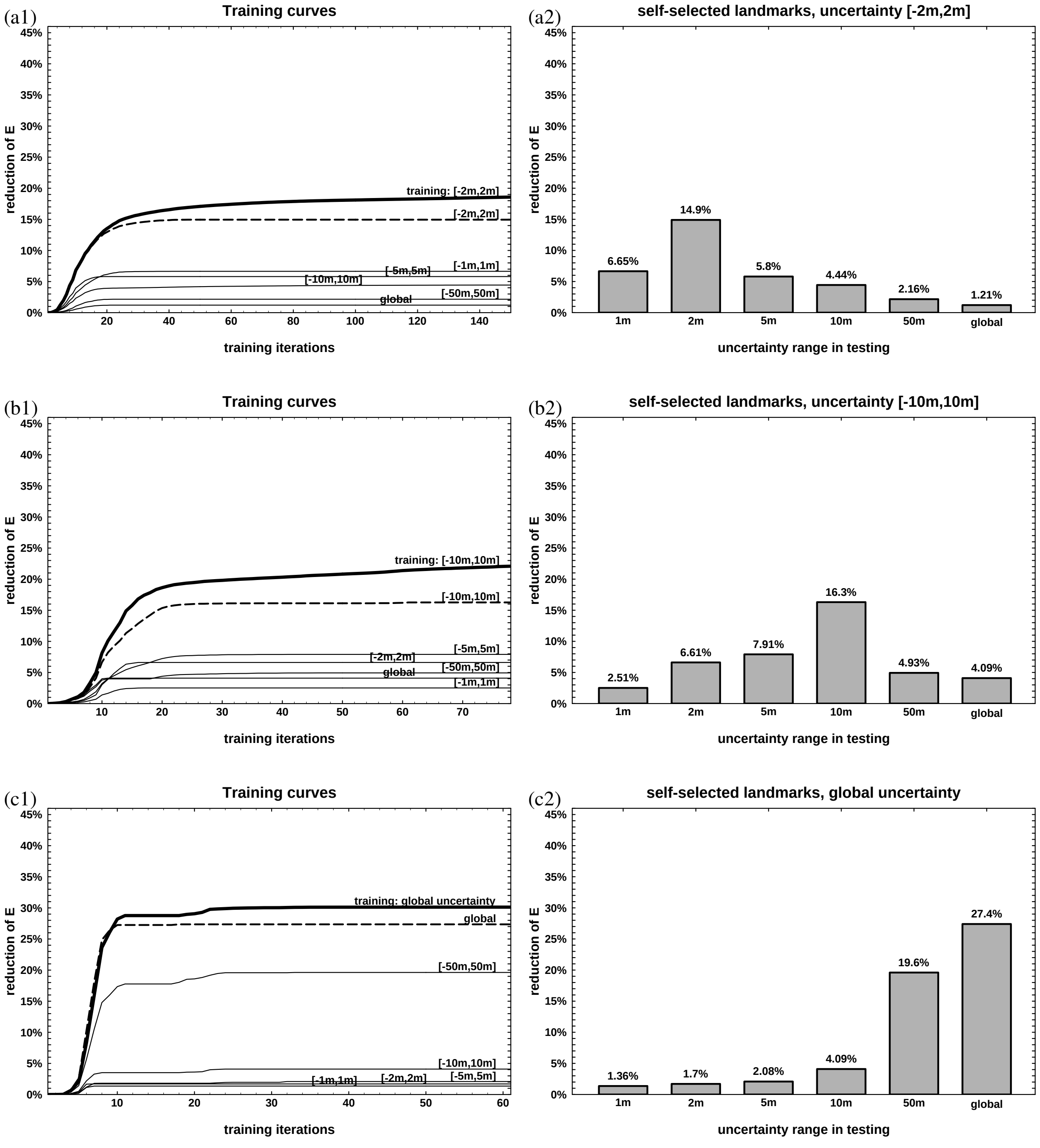


Figure 10: Results for self-selected landmarks.

testing environment possesses dark regions for which the encoded sensor input is very similar to that of doors. These dark regions are barely noticeable for humans, but for robots they form very reliable landmarks. Consequently, some of the landmarks of the network shown in Figure 9a correspond to these regions. Moreover, the network also managed to discover landmarks in regions that do not contain doors. Here adjacent hallways (which also appear darker than their surrounding) are generally used as landmarks, among other things.

It is most interesting to compare the error reduction of the self-selected landmarks with that obtained with human-selected landmarks. Each diagram in Figure 11 shows the comparison of the different networks—trained with supervised learning and with different training uncertainties—for one particular testing uncertainty. For example, Figure 11a illustrates that in cases where the (testing) uncertainty is uniformly distributed in $[-1\mathrm{m}, 1\mathrm{m}]$, the supervised approach reduces the error by 4.35% (*cf.* first bar in Figure 8). Here the networks trained for the uncertainties $[-1\mathrm{m}, 1\mathrm{m}]$ and $[-2\mathrm{m}, 2\mathrm{m}]$ are both superior: 7.02% and 6.65%. As is easy to be seen from this and the other diagrams, the network which was trained on the testing uncertainty reduces the error the most.

The comparison of supervised learning with the results obtained for self-selected landmarks proliferates the third key results of this study: Self-selected landmarks are uniformly superior to the human-selected landmarks. If the prior uncertainty is uniform in $[-2\mathrm{m}, 2\mathrm{m}]$, the corresponding self-selected landmarks reduce the error by 14.9%, which is 1.79 better than the reduction obtained with the supervised approach (8.34%, *cf.* Figure 11b). For more global uncertainties the relative difference between both approaches is even more dramatic: If the robot is globally uncertain, the error reduction for the self-selected landmarks is 27.4%, which is by a factor of 12.7 larger than the error reduction obtained with the hand-selected landmarks (2.16%). At first glance, this comparison is somewhat unbalanced, since the same supervised network is compared with different self-selecting networks, each of which has been optimized for a particular uncertainty distribution. However, since in practice the uncertainty distribution is accessible, new networks can be trained without further external information. Training new networks in a supervised manner, in contrast, would require that a human provides new target labels, *i.e.*, labels that specify which landmarks are appropriate under this new uncertainty. In addition, the reader may notice that even for the distribution where the supervised approach performs best (which is $[-2\mathrm{m}, 2\mathrm{m}]$, *cf.* Figure 8), the self-selected landmarks are superior (Figure 11b).

5.5 Multiple Landmarks

Thus far, all empirical results assume only a single landmark network. The landmark learning algorithm facilitates the simultaneous training of multiple landmark detectors, so that more than just a single bit of information can be extracted from each sensor snapshot.

Figures 12, 13, and 14 show the output characteristics for n=2, n=3, and n=4 networks, respectively. To facilitate the comparison, all these networks were trained using the uniform uncertainty $[-2\mathrm{m}, 2\mathrm{m}]$ (just like the network shown in Figure 9a). Here, too, each figure is sub-dived into four parts, which correspond to the four different runs that comprise the dataset. Each of these parts depicts the output characteristics of all n networks as a function of the location (sampled over the corresponding data set).

It is interesting to notice that each network specializes to different, mostly uncorrelated landmarks. This comes at little surprise, since the less dependent the individual landmark detectors are, the more information is extracted from sensor snapshots. Table 1 depicts the empirical correlation coefficient for the output of the four landmark networks exhibited in Figure 14. The strongest correlation, measured for

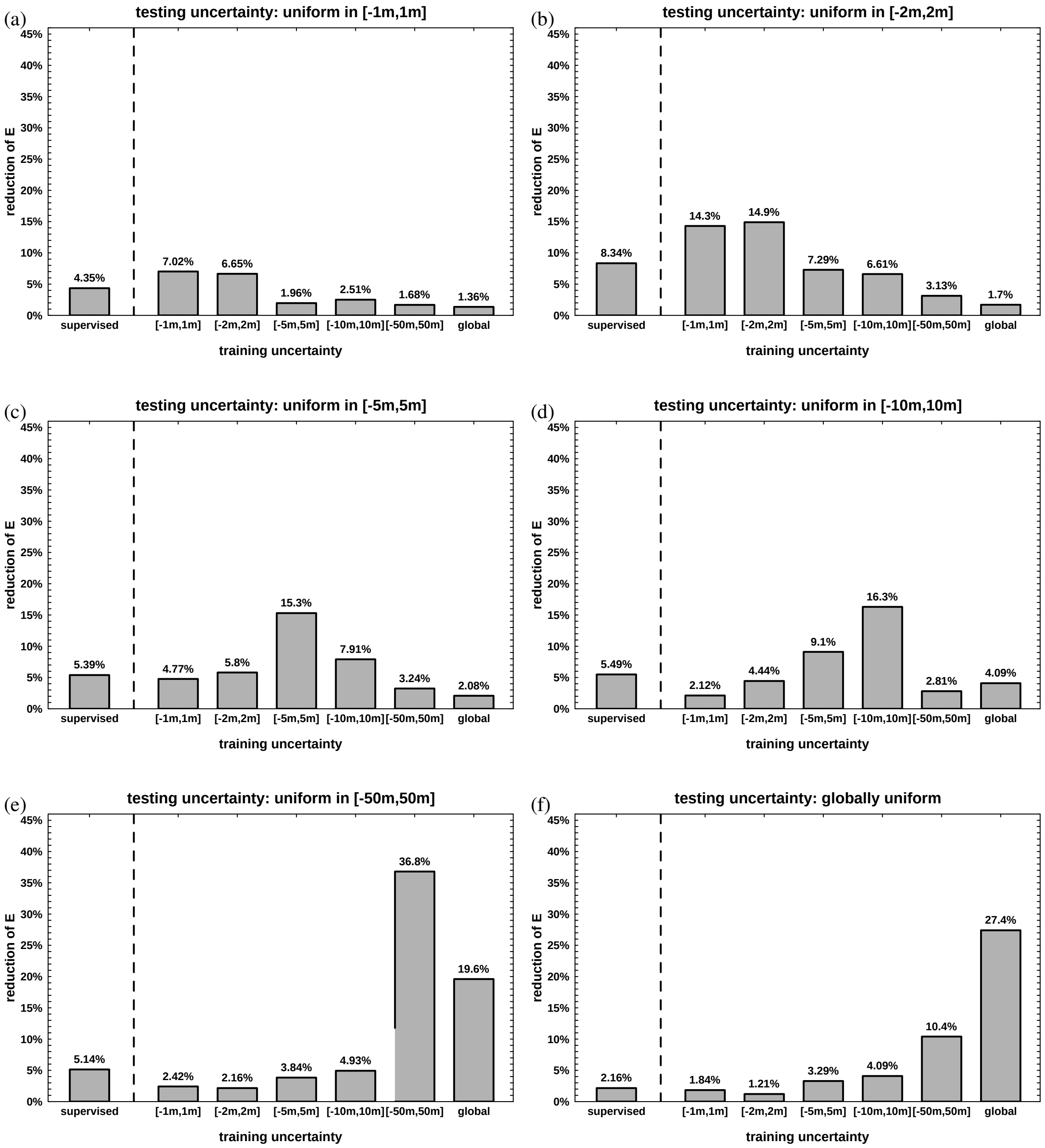


Figure 11: Performance comparison: supervised learning and self-selected landmarks, for different training and testing uncertainties.

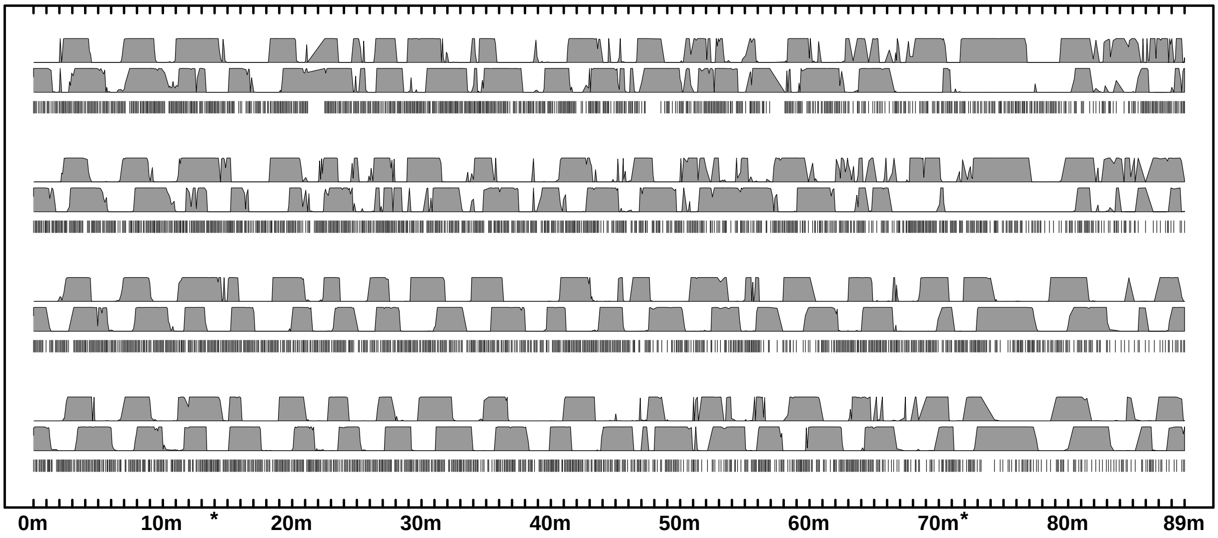


Figure 12: Landmarks selected by the robot with $n = 2$ networks.

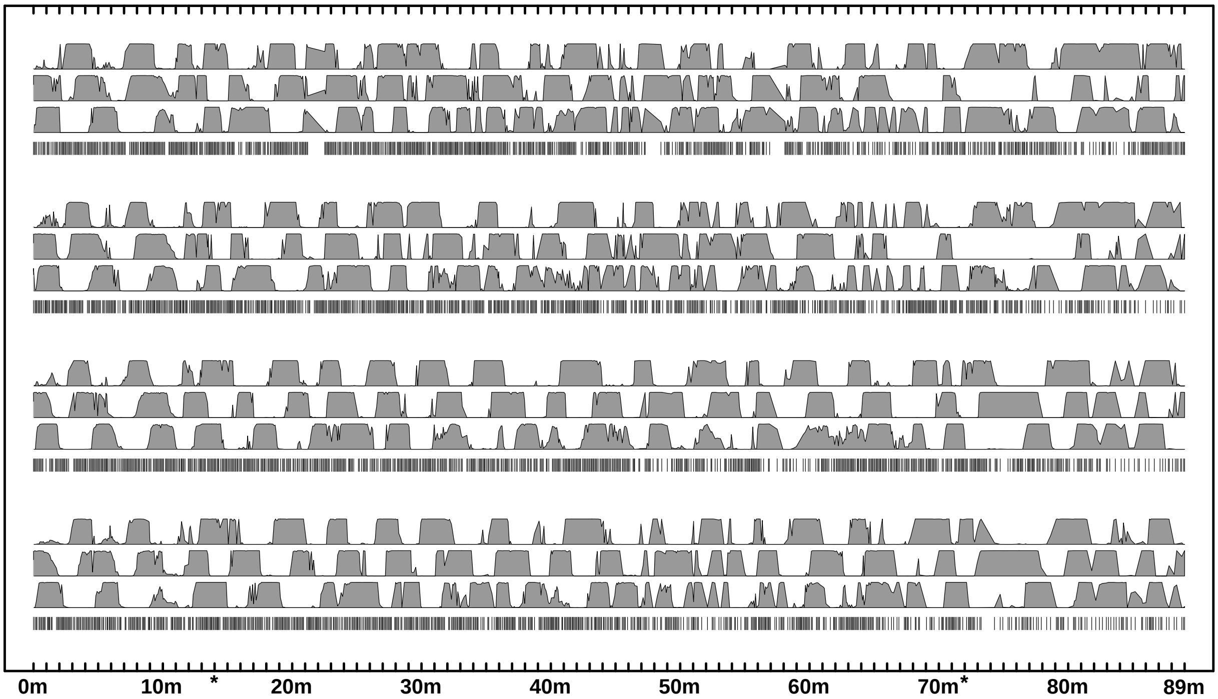


Figure 13: Landmarks selected by the robot with $n = 3$ networks.

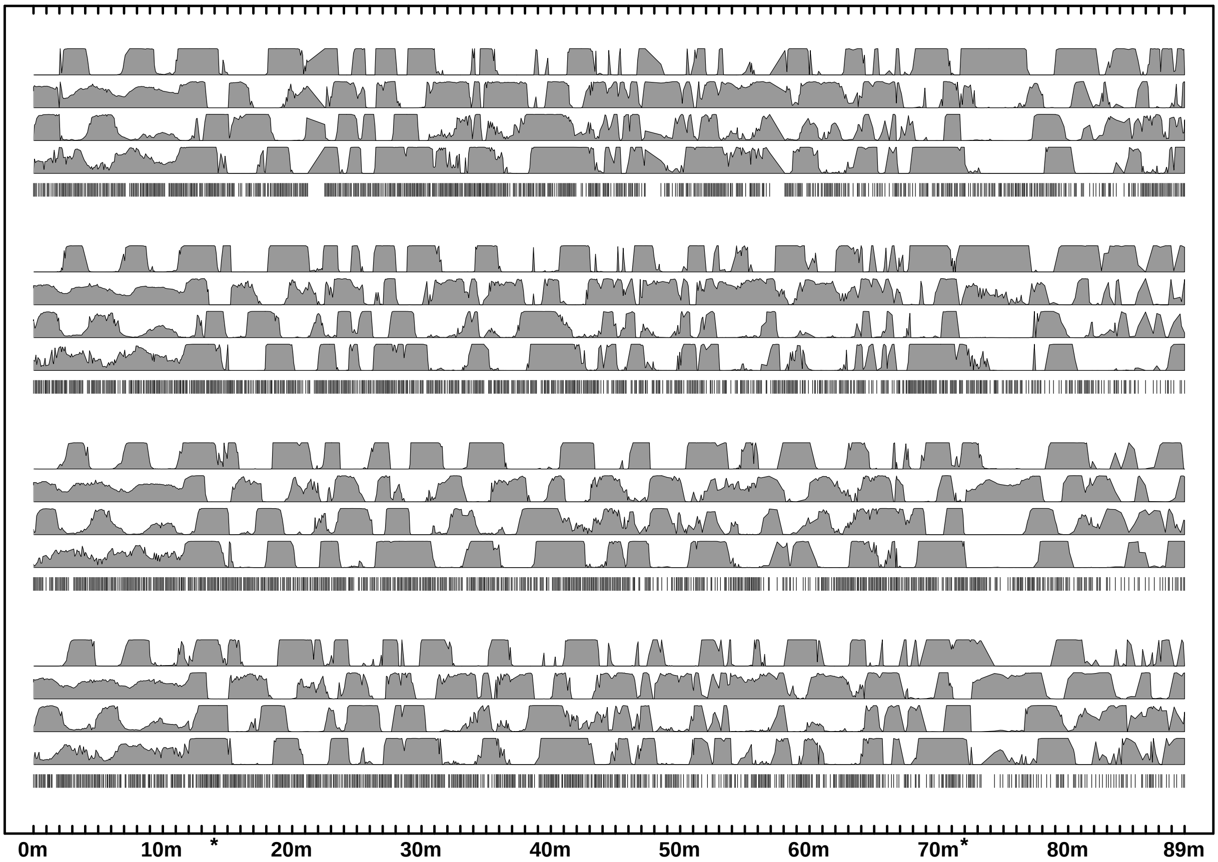


Figure 14: Landmarks selected by the robot with $n = 4$ networks.

	g_1	g_2	g_3	g_4
g_1	1	-0.251	-0.378	0.279
g_2	-0.251	1	0.007	-0.164
g_3	-0.378	0.007	1	0.138
g_4	0.279	-0.164	0.138	1

Table 1: Correlations of network outputs.

the testing set, is 0.378. Most of the networks are only weakly correlated, *i.e.*, the correlation coefficient is close to zero. While ideally, one would expect stochastic independence—which would imply that the correlation between any pair of networks would be zero—in practice such landmarks might not exist. We interpret the results in Table 1 as an indication that the approach indeed manages to discover different landmarks; an interpretation that is confirmed by comparing the characteristic curves of each landmark detector.

Most of all, our believes are supported by the performance comparison shown in Figure 15. With the exception of the supervised results, each network investigated here was trained for uniform uncertainty in $[-2\text{m}, 2\text{m}]$, and tested under various other uncertainties, which are specified at the top of each diagram. These results are intriguing. The more networks are used, the (significantly) larger the error reduction. For example, as shown in Figure 15b, the self-selected approach with with n=4 networks reduces the error by as much as 41.7%, which outperforms both single-net approaches (supervised and self-selected) by a factor of 5.0, or 2.8, respectively. The results for other testing uncertainties are similar, showing that with larger values of n the self-selective approach becomes even more superior. To summarize, the fourth key result reported here is that multiple networks are superior to a single landmark detector network. Notice that no additional information is required to train and use multiple networks. We conjecture that with limited amounts data, there exists an "optimal" number of networks beyond which the performance degrades (due to over-fitting).

Not shown here are various results obtained for training uncertainties other than uniform distributions in $[-2\text{m}, 2\text{m}]$, which basically confirm the findings reported here. For brevity we also omitted similar results obtained for the two other camera configurations, c_{left} and c_{up}, which also are well in tune with the results reported here.

5.6 Active Perception

In the remainder of this section we will provide some empirical results obtained for active perception. To investigate active perception, data for three different camera configurations c_{left}, c_{up} and c_{right} was used. For each of these configurations, a separate network was trained, using a uniform uncertainty distribution in $[-2\text{m}, 2\text{m}]$.

Robot motion is inaccurate. To model phenomena such as drift and slippage, we employed the simplistic model that robot motion introduces 10% uniformly distributed randomness into the location of the robot. More specifically, after moving forward by Δl starting at location $\tilde{l}$, the probability density of the resulting location l was assumed to be distributed uniformly, with the center at $\tilde{l} + \Delta l$ and the range $0.1 \cdot \Delta l$. The conditional density is formally written as

$$P_{\Delta l}(l \mid \tilde{l}) \quad = \quad \begin{cases} 0 & \text{if} \quad l < \tilde{l} + 0.95\Delta l \\ 10 \cdot \Delta l^{-1} & \text{if} \quad \tilde{l} + 0.95\Delta l \leq l \leq \tilde{l} + 1.05\Delta l \\ 0 & \text{if} \quad l > \tilde{l} + 1.05\Delta l \end{cases} \tag{39}$$

This model is extremely simplistic and, moreover, overly conservative, *i.e.*, it intentionally overstates the true randomness in robot motion. To some extent, overly conservative models counterbalance the damaging effect of the Markov assumption which, strictly speaking, does not hold true for robot motion. No effort was made to build a more realistic model.

In our experiments, the robot moves continuously. After each $\Delta l = 0.5$ meter of autonomous robot motion, the density $\hat{P}$ is updated according to the incremental update algorithm derived in Section 2

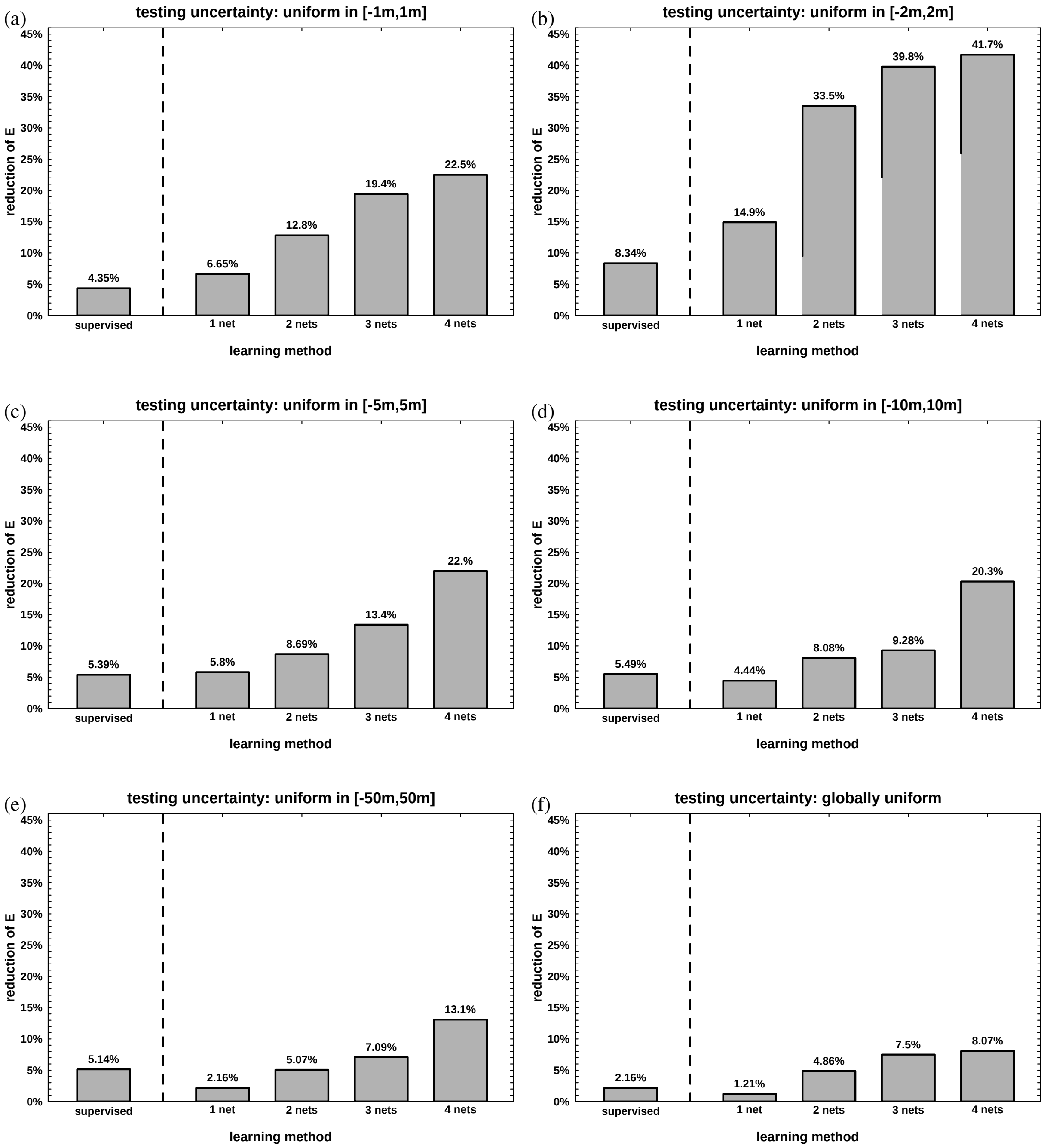


Figure 15: Performance comparison: supervised learning self-selected landmarks for different numbers of networks. During training, the uncertainty is distributed uniformly in $[-2\text{m}, 2\text{m}]$.

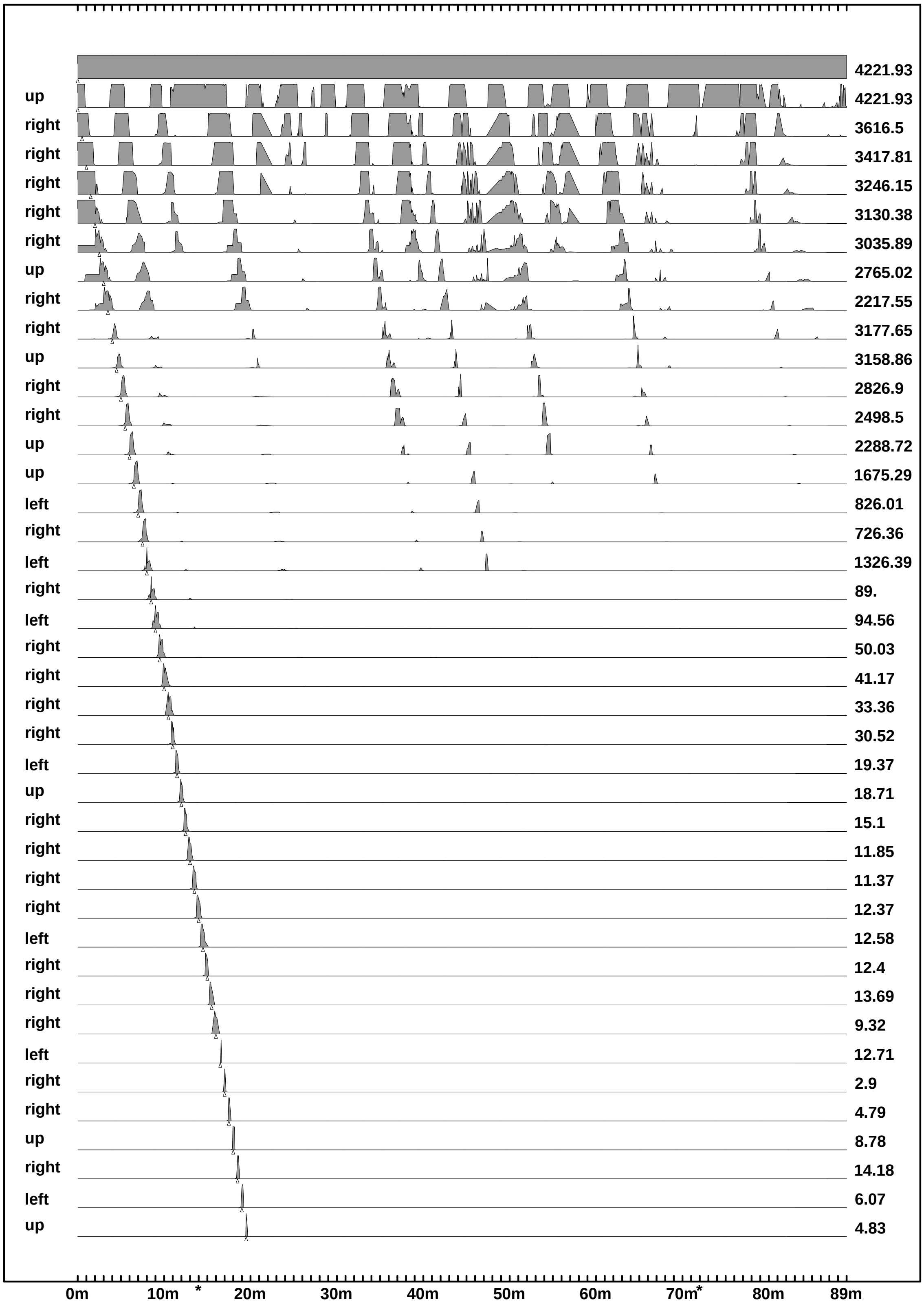


Figure 16: Localization with active perception.

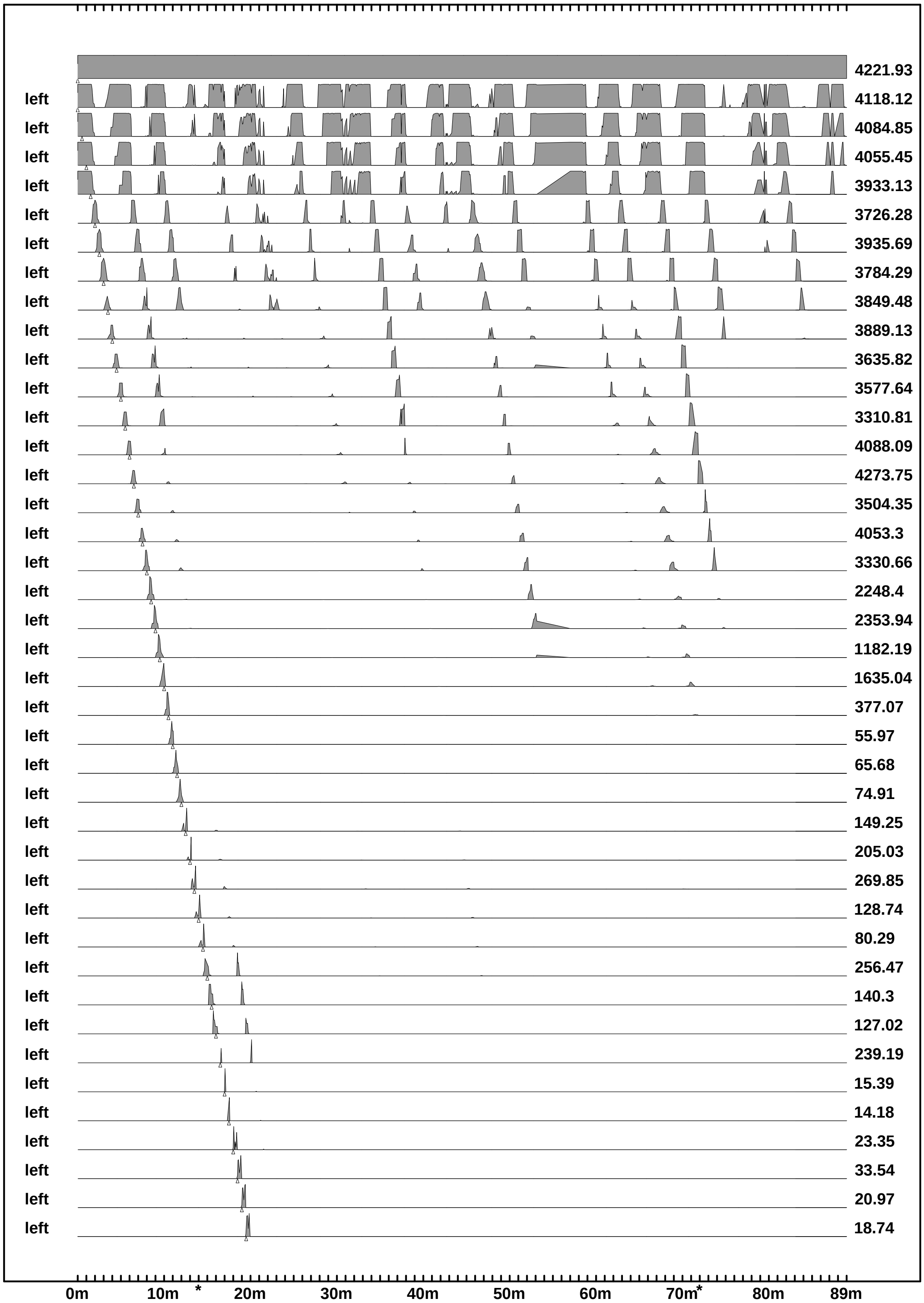


Figure 17: Localization with passive perception: Looking left.

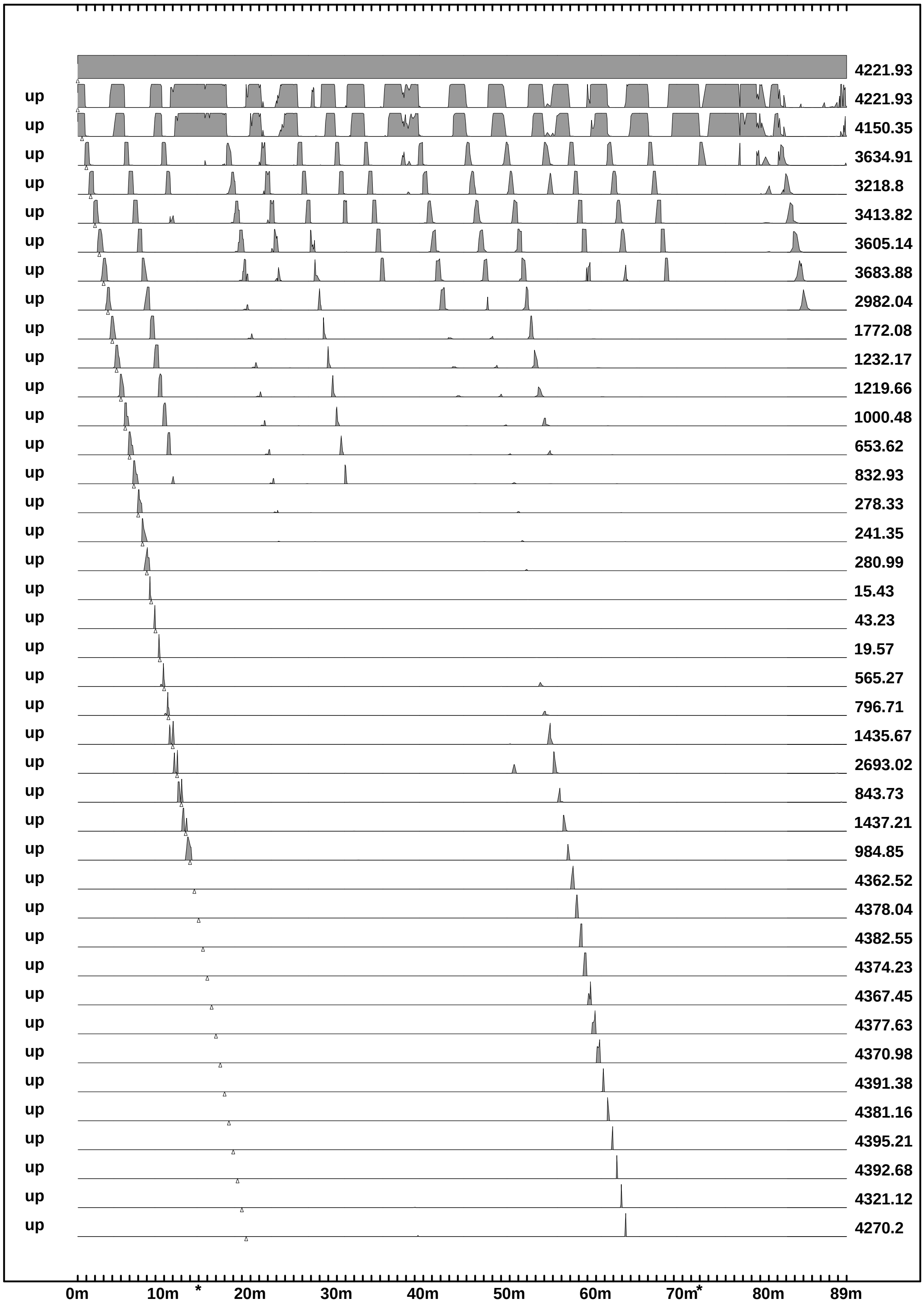


Figure 18: Localization with passive perception: Looking up.

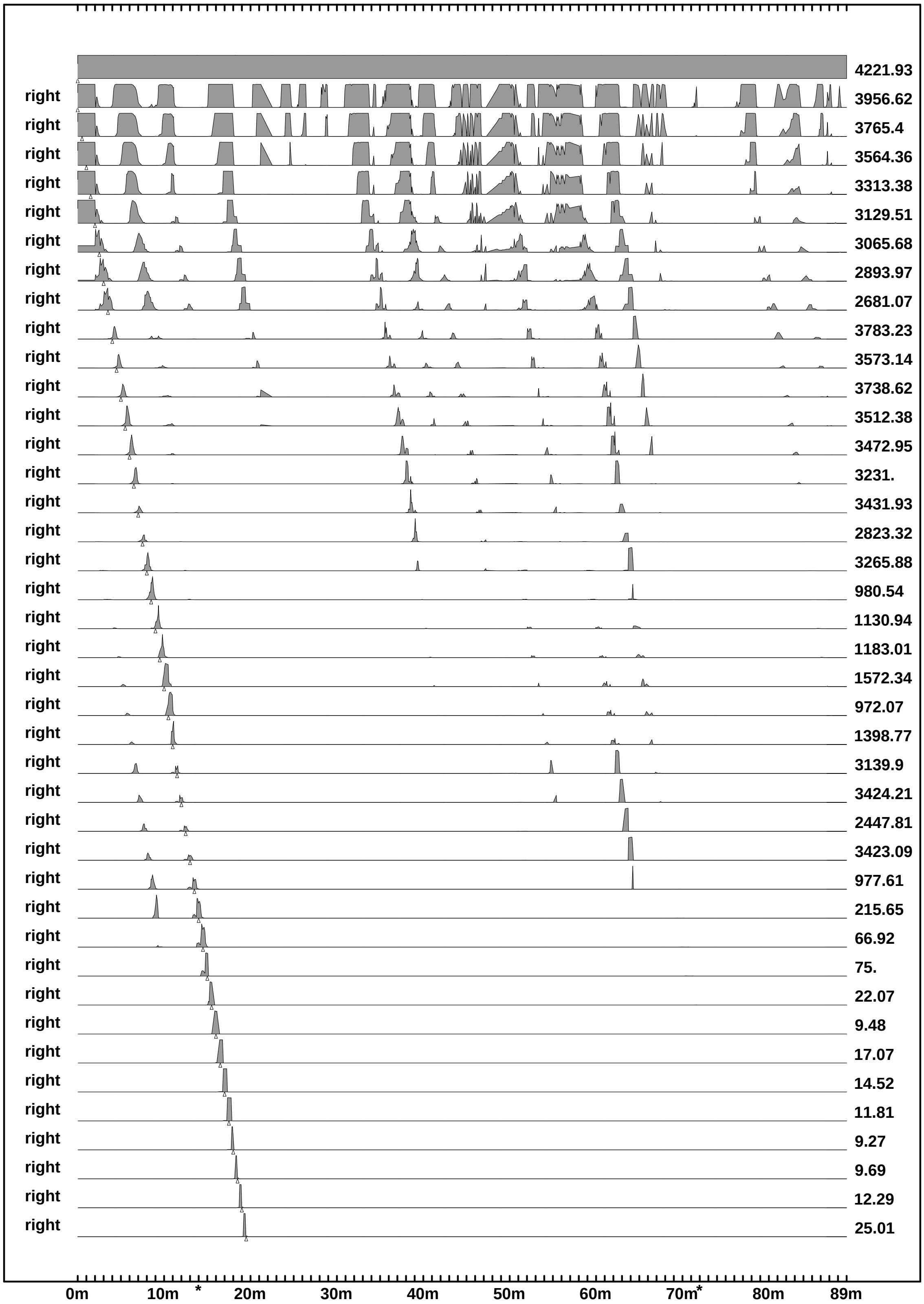


Figure 19: Localization with passive perception: Looking right.

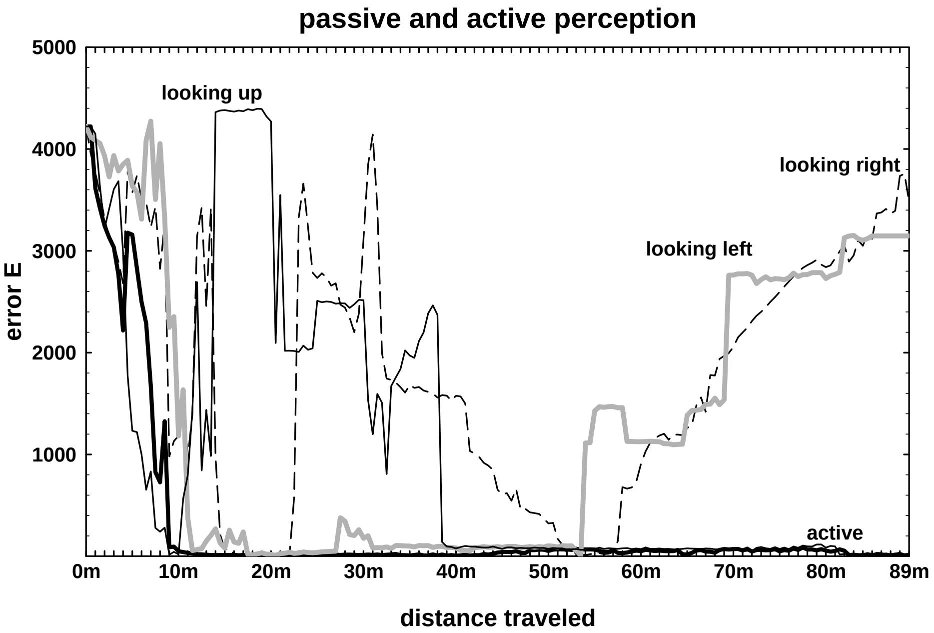


Figure 20: Localization error for passive and active perception. Notice that the error for active perception (thick solid line) stays close to zero after approx. 9 meters of robot motion.

(*cf.* (12)). While the robot is moving, it determines its next sensor configuration by maximizing E_c (*cf.* (33)). At the end of each 0.5 meter segment, the robot takes and processes a single sensor snapshot. The landmark information extracted from this snapshot is used to refine $\hat{P}$ (*cf.* (10) and (11)).

Figure 16 depicts the uncertainty distribution $\hat{P}$ as the robot moves through the corridor. Each of the grayly-shaded curves depicts the uncertainty at a particular point in time. The small triangle under each diagram depicts the "true" location. Initially, the location is assumed to be unknown, thus its uncertainty is globally uniform (*cf.* first row in Figure 16). The robot now decides to direct its camera towards the ceiling (*i.e.*, $c^* = c_{\text{up}}$) and takes the first sensor snapshot, resulting in a multi-modal density with reduced entropy (second row in Figure 16—notice that for the reader's convenience all densities in this figure are normalized so that they have the same amplitude). For the second snapshot the robot directs its camera towards the right. As the robot progresses, the sensor configuration is changed frequently and the uncertainty increases further, until eventually, after approximately 9 meters of continuous robot motion (and 18 sensor measurements), the robot reliably knows where it is. In all subsequent moves, the robot accurately maintains the correct location estimation. Notice that Figure 16 depicts only the first 20 meters of the entire corridor.

The accuracy of the estimation depends crucially on the ability to direct the robot's sensors. Figures 17, 18, and 19 show densities for fixed camera configurations (passive perception). All experiments illustrate that the robot occasionally looses track of its location, triggered by systematic, time-dependent variations in its perceptions. Figure 20 surveys the localization error for all four experiments as a function of total travel distance (and hence number of sensor measurements). As can be seen there, the active approach (bold line) quickly, after approx. 9 meters, settles to the right location (*i.e.*, small error) and tracks the location accurately. All passive approaches, in contrast, are occasionally off. These findings clearly illustrate the fifth and final key result in this study: active perception is superior to passive perception in landmark-based localization. It is superior in two aspects: it converges faster to the correct location, and it exhibits increased robustness in the long run.

6 Discussion

This paper presents an approach to landmark-based localization for mobile robots. The key problem addressed in this paper is how to automatically choose the best landmarks, out of the myriad of possible choices. The approach presented here allows a robot to select its landmarks automatically, based on a single criterion: the utility for localization. More specifically, landmarks are recognized by artificial neural networks, which are trained so as to minimize the Bayesian a posteriori localization error. As a result, landmarks (and the "programs" that recognize them) are learned automatically, without the necessity to provide further information. The idea of minimizing the Bayesian error is also successfully applied to problems of active perception and active navigation, in which the robot actively controls its sensors and actuators so as to determine its location as accurately as possible.

The key results of this paper are:

1. Unlike in previous approaches, no human is required to specify appropriate landmarks. The robot is capable of selecting its landmarks and learning the corresponding landmark detectors.

2. Self-selected landmarks are uniformly superior to our current best human-selected visual landmarks. For example, in situations in which these human-selected landmarks work best (uniform uncertainty in $[-2\mathrm{m}, 2\mathrm{m}]$), self-selected landmarks (with $n = 4$ networks) work 5.0 times better. In situations where our hand-selected landmarks work poorly (*e.g.*, global uncertainty), robot-selected landmarks were found to be 12.7 times as informative as human-selected ones.

3. Active perception is superior to passive perception. Empirical results demonstrated that by moving the camera so as to maximize the reduction of the a posteriori localization error, the robot can more accurately localize itself and track its location as it is moving around.

We attribute the finding that the robot-selected landmarks work so much better than our hand-selected ones primarily to three reasons: First, the utility of landmarks depends on the a priori uncertainty in the location. To select the right landmark, the human must have an understanding of the uncertainty (which of course varies) and its role in localization. Second, robot sensors are quite different from human senses. Thus, it is generally questionable if landmarks that are appropriate for human orientation are appropriate for robots as well—the experimental findings reported here certainly provide evidence to the contrary. Third and finally, if landmark detectors are learned, the built-in bias of the function approximator imposes limitations as to what can be learned from examples. Thus, for a human to select good landmarks, he/she must be knowledgeable about the particular data encoding, and the properties of the function approximator at hand.

Despite these encouraging results, there is a variety of open questions and limitations that warrant discussion and future research.

- **Overcoming the conditional independence assumptions.** The reader should notice that the general probabilistic equations for localization, on which this work is based, hinges on three conditional independence assumptions (Markov assumptions), which in fact are commonly made in approaches to landmark-based localization:

  1. Given the true location of the robot l, multiple observations are conditionally independent. Notice that this is trivially the case if observations are deterministic functions of the robot

location. However, if the environment possesses state (*e.g.*, a human which walks by and which is visible in multiple sensor snapshots), this assumption will usually be violated.

2. Given the true location l, the output of the individual landmark detectors is stochastically independent. If landmark detectors are deterministic (as is the case in this paper, where they are realized by artificial neural networks), this assumption is logically implied by the first conditional independence assumption.
3. The probabilistic effects of robot actions are (conditionally) independent. Notice that this is usually not the case for fast moving robots, unless an explicit model of the robot dynamics is taken into account, However, by using an overly conservative model of robot motion (as employed in our experiments), many of the damaging implications of the assumption can be avoided in practice.

These assumptions are often referred to as Markov assumption. A recent approach by Kortenkamp and Weymouth [19], who argue for integrating different sensors such as cameras and sonar sensors, relies on a fourth conditional independence assumption: Given the true location l, the randomness in the sonar measurements are independent of the randomness in the camera images. If this assumption holds true, both sensor modalities can be interpreted independently and then integrated using Bayes rule. This assumption is not made in the current approach, in which both sensors are interpreted by a single, monolithic neural network. Obviously, if the environment is dynamic and, for example, people may be in the field of view of both sensors, this assumption is violated (as will be the first independence assumption listed here). However, by updating location densities in a conservative way (as done in [19]), we believe that many of the pitfalls arising from this assumption can be avoided in practice.

It is generally desirable to find algorithms that overcome, at least in part, the conditional independence assumptions. Often, conditional dependence is caused by hidden state. For example, an (unmodeled) human that appears in multiple subsequent camera images introduces dependence. An approach for reducing the effects of hidden state is to estimate it directly, and augment the state space correspondingly. There are various methods for estimating hidden state, such as Hidden Markov Models [6, 22, 29], recurrent neural networks [28, 43], Kalman filters [16, 37], or tree-based or instance-based approaches [21]. The obvious disadvantage of such an approach would be the increased size of the state space, which might cause severe data gathering and computational efficiency problems. To the best of of knowledge, none of the existing probabilistic approaches to mobile robot localization is capable of estimating hidden state.

- **Computational complexity.** The current approach is computationally expensive. In our implementation, the computation of the Bayesian error (*cf.* (23) or (28)) and its derivative (*cf.* (27) or (30)) is done somewhat more efficiently than suggested by the derivation of the update formulas, by re-shuffling some of the terms. Nevertheless, computing the full Bayesian error is in $O(2^n|X|^2)$, where n is the number of landmark networks, and $|X|$ is the training set size. The gradient of the Bayesian error (23) requires time in $O(2^n n|X|^3)$. While the nature of the uncertainty prior has a strong impact on the constant factor in the computational complexity, training networks using the correct gradient is infeasible for large numbers of networks. In fact, in the training scheme employed here we modified gradient descent to cache the gradient, leading to a significant (but constant) speed-up. Fast computation is particularly important for active perception and active navigation, where decisions have to be made in real-time. While with a single network, as employed in our experiments, the perceptual configuration can well be computed in real-time, time limitations prohibit using more than a single network in the current implementation.

Reducing the computation time is a key goal of future research. At first glance, that training time can be reduced almost arbitrarily by approximating the gradient, using only parts of the training set (on-line learning, or other stochastic gradient descent schemes [13]). However, by doing so the approximate gradient deviates from the true gradient, which can cause severe convergence problems.

A second strategy for reducing the computational load would be to represent densities more compactly. Currently, densities are spanned over each available data points, which, in a way, provides the highest resolution that is supported by the data. Others have used parameterized densities such as Kalman filters [37], or more coarse-grained models of densities such as evenly spaced grids, where the size of each grid cell might be 10 centimeters [4] or as much as 1 meter [18, 35]. Such representations are computationally easier to handle, yet they impose intrinsic limitations in the resolution of densities (which in turn introduces an intrinsic error into the fine-grain localization). We believe that the issue of representation is mostly orthogonal to the issues of landmark discovery addressed in this paper, and we deliberately chose the current representation in order to evaluate the effect of landmark selection on the localization error as objectively as possible.

- **Multi-step active control.** Another limitation of the current approach is the greediness of active perception and active navigation. The approaches investigated here select actions and sensor configurations so as to maximize the information gain in the immediate next sensor snapshot. Strictly speaking, to sense and act optimally, multiple sensor snapshots must be taken into account, and the active perception/navigation problem becomes a complex planning problem in the space of all density functions (belief space).

  Even if the state space were discrete (which is naturally not the case for mobile robot locations), determining optimal actions by planning in belief space is known to be hopelessly intractable, despite some remarkable recent progress [15, 20]. In our past research, we have used greedy methods very successfully for robot exploration [38, 39] albeit the fact that they generate suboptimal control—however, greedy methods were used to acquire models of the environment, not to determine the location of the robot (the location was estimated with a maximum likelihood estimator and then assumed to be correct). Bridging the gap between optimal multi-step control and greedy control is an important open problem for any approach that controls physical systems under uncertainty.

- **Multi-step optimization.** A similar limitation applies to the way landmark networks are trained. At present, neural networks are trained so as to minimize the a posteriori localization error, which is made after processing the immediate next sensor snapshot. Mathematically, it would be straightforward to extent this approach to multiple sensor readings. However, if more than a single sensor snapshot is taken into account, the Bayesian error of a set of landmark detectors becomes a function of the program that controls the robot. In other words, landmark detectors cannot be learned without taking robot control into account. This appears to be undesirable, since additional knowledge about the control strategies is necessary, and landmark networks that are optimal for one navigation strategy might be sub-optimal for another one. Moreover, it is unclear how much can be gained by taking sequences of sensor readings into account during learning.

- **Correcting dead-reckoning errors in the training set.** In an effort to minimize the information provided by a human, locations in the training set were estimated based on cumulative dead-reckoning. Obviously, dead-reckoning is erroneous, and with more accurate position estimations the results can most likely be improved. However, the dead-reckoning errors in the training set were considerably small in our experiments, and as our results indicate, the landmark learning algorithm appears to be considerably robust to these error.

The reader should notice that the error E can be used to re-estimate the location of each individual data point. Given a belief distribution of the true location (instead of a single location label), the location itself can be estimated by minimizing E, just like the weights in the neural network. However, one has to be careful when moving data points around, since their location is crucial for the outcome of learning.

One of the interesting properties of the landmark learning algorithm is that it minimizes a single quantity: the weighted a posteriori error in localization. In the introduction to this paper we listed five desired properties for landmarks: they should be (1) stationary, (2) reliably recognizable, (3) sufficiently unique, (4) there should be enough of them, and (5) they should relate to the prior uncertainty. In fact, most of these criteria establish natural trade-offs: The more landmarks the robot is willing to recognize, for example, the more likely it is to confuse them. The approach presented here provides a rationale, single solution as to how to trade off these properties: Minimizing the a posteriori error automatically trades-off all of them, by exclusively assessing he "utility" of landmarks for localization. The framework presented here integrates multiple sensor modalities, and is rich enough to provide rigorous answers to problems of localization, landmark learning, and active perception and navigation. Moreover, if the environment changes, learning enables robots to automatically adjust to the conditions, and, if necessary, to discover new landmarks.

Acknowledgments

The author wishes to express his gratitude to the RHINO and the XAVIER mobile robot groups, particularly Wolfram Burgard, Dieter Fox, and Reid Simmons, for stimulating discussions.

References

[1] Betke, M. and Gurvits, L. *Mobile Robot Localization using Landmarks*. no. SCR-94-TR-474, Siemens Corporate Research, Pinceton, December 1993. *will also appear in the IEEE Transactions on Robotics and Automation*.

[2] Borenstein, J. *The Nursing Robot System*. Technion, Haifa, Israel, June 1987.

[3] Buhmann, J., Burgard, W., Cremers, A. B., Fox, D., Hofmann, T., Schneider, F., Strikos, J., and Thrun, S. *The Mobile Robot Rhino*. **AI Magazine**, vol. 16 (1995)

[4] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. *Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids*. in: **Proceedings of the Thirteenth National Conference on Artificial Intelligence**, AAAI. AAAI Press/MIT Press, Menlo Park, 1996.

[5] Burgard, W., Fox, D., Hennig, D., and Schmidt, T. *Position Tracking with Position Probability Grids*. University of Bonn, Computer Science Department III, Bonn, Germany, 1996. *internal report*.

[6] Chrisman, L. *Reinforcement Learning with Perceptual Aliasing: The perceptual distinction approach*. in: **Proceedings of 1992 AAAI Conference**. AAAI Press / The MIT Press, Menlo Park, CA, 1992.

[7] Chung, K. **Markov chains with stationary transition probabilities**. Springer Publisher, Berlin, 1960.

[8] Elfes, A. *Sonar-Based Real-World Mapping and Navigation*. **IEEE Journal of Robotics and Automation**, vol. RA-3 (1987), pp. 249–265.

[9] Everett, H., Gage, D., Gilbreth, G., Laird, R., and Smurlo, R. *Real-World Issues in Warehouse Navigation*. November 1994, *Volume 2352*.

[10] Feng, L., Borenstein, J., and Everett, H. *"Where am I?" Sensors and Methods for Autonomous Mobile Robot Positioning*. no. UM-MEAM-94-12, University of Michigan, Ann Arbor, MI, December 1994.

[11] Fox, D., Burgard, W., and Thrun, S. *The Dynamic Window Approach to Collision Avoidance*. **IEEE Robotics and Automation**, vol. (1996), *to appear, also appeared as Technical Report IAI-TR-95-13, University of Bonn, 1995.*

[12] Gelb, A. **Applied Optimal Estimation**. MIT Press, 1974.

[13] Hertz, J., Krogh, A., and Palmer, R. G. **Introduction to the theory of neural computation**. Addison-Wesley Pub. Co., Redwood City, California, 1991.

[14] Hinkel, R. and Knieriemen, T. *Environment Perception with a Laser Radar in a Fast Moving Robot*. in: **Proceedings of Symposium on Robot Control**, edited by . Karlsruhe, Germany, 1988, pp. 68.1–68.7.

[15] Kaelbling, L., Littman, M., and Cassandra, A. *Planning and Acting in Partially Observable Stochastic Domains*. Computer Science Department, Brown University, Providence, RI, 1996.

[16] Kalman, R. E. *A new Approach to Linear Filtering and Prediction Problems*. **Trans. ASME, Journal of Basic Engineering**, vol. 82 (1960), pp. 35–45.

[17] King, S. and Weiman, C. *HelpMate Autunomous Mobile Robot Navigation System*. November 1990, pp. 190–198. *Volume 2352*.

[18] Koenig, S. and Simmons, R. *Passive Distance Learning for Robot Navigation*. in: **Proceedings of the Thirteenth International Conference on Machine Learning**, edited by L. Saitta. 1996

[19] Kortenkamp, D. and Weymouth, T. *Topological mapping for mobile robots using a combination of sonar and vision sensing*. in: **Proceedings of the Twelfth National Conference on Artificial Intelligence**, AAAI. AAAI Press/MIT Press, Menlo Park, 1994, pp. 979–984.

[20] Littman, M., Cassandra, A., and Kaelbling, L. *Learning Policies for Partially Observable Environments: Scaling Up*. in: **Proceedings of the Twelfth International Conference on Machine Learning**, edited by A. Prieditis and S. Russell. 1995.

[21] McCallum, R. A. *Instance-Based State Identification for Reinforcement Learning*. in: **Advances in Neural Information Processing Systems 7**, edited by G. Tesauro, D. Touretzky, and T. Leen. MIT Press, Cambridge, MA, 1995, *To appear*.

[22] McCallum, R. A. *Overcoming Incomplete Perception with Utile Distinction Memory*. in: **Proceedings of the Tenth International Conference on Machine Learning**, edited by P. E. Utgoff. Morgan Kaufmann, San Mateo, CA, 1993, pp. 190–196.

[23] Moravec, H. P. *Sensor Fusion in Certainty Grids for Mobile Robots*. **AI Magazine**, vol. (1988), pp. 61–74.

[24] Moravec, H. and Martin, M. *Robot Navigation by 3D Spatial Evidence Grids*. Mobile Robot Laboratory, Robotics Institute, Carnegie Mellon University, 1994.

[25] Neven, H. and G., S. *Dynamics parametrically controlled by image correlations organize robot navigation*. **Biological Cybernetics**, 1995. *to appear*.

[26] Nourbakhsh, I., Powers, R., and Birchfield, S. *DERVISH an Office-Navigating Robot*. **AI Magazine**, vol. 16 (1995), pp. 53–60.

[27] Pearl, J. **Probabilistic reasoning in intelligent systems: networks of plausible inference**. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[28] Pearlmutter, B. A. *Learning State Space Trajectories in Recurrent Neural Networks*. **Neural Computation**, vol. 1 (1989), pp. 263–269. *Also appeared as: Technical Report CMU-CS-88-191, Carnegie Mellon University, 1988.*

[29] Rabiner, L. R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. in: **Proceedings of the IEEE**, IEEE, edited by . 1989. *IEEE Log Number 8825949*.

[30] Rencken, W. *Autonomous Sonar Navigation in Indoor, unknown and Unstructired Environments*. in: **Intelligent Robots And Systems**, edited by V. Graefe. Elsevier, 1995.

[31] Rencken, W. *Concurrent Localisation and Map Building for Mobile Robots Using Ultrasonic Sensors*. in: **Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems**, edited by . Yokohama, Japan, 1993, pp. 2129–2197.

[32] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. *Learning Internal Representations by Error Propagation*. in: **Parallel Distributed Processing. Vol. I + II**, edited by D. E. Rumelhart and J. L. McClelland. MIT Press, 1986.

[33] Schiele, B. and Crowley, J. *A Comparison of Position Estimation Techniques Using Occupancy Grids*. in: **Proceedings of the 1994 IEEE International Conference on Robotics and Automation**. San Diego, CA, 1994, pp. 1628–1634.

[34] Simmons, R. *The Curvature-Velocity Method for Local Obstacle Avoidance*. in: **Proceedings of the International Conference on Robotics and Automation**, edited by . Minneapolis MN, 1996,

[35] Simmons, R. and Koenig, S. *Probabilistic Robot Navigation in Partially Observable Environments*. in: **Proceedings of IJCAI-95**, IJCAI, Inc. Montreal, Canada, 1995.

[36] Smith, R., Self, M., and Cheeseman, P. *Estimating Uncertain Spatial Relationships in Robotics*. in: **Autonomous Robot Vehnicles**, edited by I. Cox and G. Wilfong. Springer-Verlag, 1990, pp. 167–193.

[37] Smith, R. C. and Cheeseman, P. *On the Representation and Estimation of Spatial Uncertainty*. no. TR 4760 & 7239, SRI, 1985.

[38] Thrun, S. *Exploration and Model Building in Mobile Robot Domains*. in: **Proceedings of the ICNN-93**, IEEE Neural Network Council, edited by E. Ruspini. San Francisco, CA, 1993, pp. 175–180.

[39] Thrun, S. and Bücken, A. *Integrating Grid-Based and Topological Maps for Mobile Robot Navigation*. in: **Proceedings of the Thirteenth National Conference on Artificial Intelligence**, AAAI. AAAI Press/MIT Press, Menlo Park, 1996.

[40] Vapnik, V. **Estimations of dependences based on statistical data**. Springer Publisher, 1982.

[41] Wasserman, P. D. **Neural computing: theory and practice**. Von Nostrand Reinhold, New York, 1989.

[42] Weiß, G., Wetzler, C., and von Puttkamer, E. *Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans*. in: **Proceedings of the International Conference on Intelligent Robots and Systems**. 1994, pp. 595–601.

[43] Williams, R. J. and Zipser, D. *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*. **Neural Computation**, vol. 1 (1989), pp. 270–280. *Also appeared as: Technical Report ICS Report 8805, Institute for Cognitive Science, University of California, San Diego, CA, 1988*.