ISSN 1361 - 6161

THE UNIVERSITY of MANCHESTER

Computer Science
University of Manchester

Automatic Selection of Landmarks for Mobile Robot Navigation

Ulrich Nehmzow, David Gelder
and Tom Duckett

Department of Computer Science

University of Manchester

Technical Report Series

UMCS-00-7-1

Automatic Selection of Landmarks for Mobile Robot Navigation*

Ulrich Nehmzow, David Gelder and Tom Duckett†

Department of Computer Science
University of Manchester
Oxford Road, Manchester, UK.
`ulrich@cs.man.ac.uk`

31.7.2000

---

*Refereed by: Ian Pratt-Hartmann

†Now with Department of Technology, University of Örebro, S - 70182 Örebro.

Abstract

Landmark-based navigation systems for mobile robots typically map locations within the environment at regular intervals. Regular mapping in fixed intervals, however, has two main weaknesses:

1. Highly conspicuous and recognisable landmarks may be missed because they lie in between sampling points, and
2. commonplace landmarks with little locative information get mapped, and result in early saturation of maps and mapping of irrelevant data.

This report describes a landmark selection mechanism based on an acquired internal model of "usual" relationships between perceptions subsequent in time. Wherever perceptions differ from the expected sensation, the location is selected as a landmark for mapping.

The report describes the landmark selection mechanism in detail, and supplies evidence that landmarks are selected persistently on multiple traversal of the same environment.

1 Introduction

1.1 Motivation for Automatic Landmark Selection

1.1.1 Dead Reckoning versus Landmark-Based Navigation

Navigation is one of the most fundamental competences of any mobile agent. It facilitates purposive, goal-directed motion, and is therefore the core component of mobile robot applications such as inspection, transportation, surveillance, or guidance.

Because of incorrigible drift errors, odometry-based navigation such as path integration (dead reckoning) can only be used over very short distances, and robot navigation systems that operate over realistic distances of several hundred metres route length have to use different methods for localisation and path planning.

The most common approach to such middle scale navigation is that of using perceptual landmarks. A number of systems have been presented at Manchester [Duckett and Nehmzow, 1998, Nehmzow and Owen, 2000, Nehmzow, 1999] and worldwide [Kurz, 1996, Yamauchi and Beer, 1996, Zimmer, 1995, Franz et al., 1998] [Gutmann and Konolige, 1999] that perform such navigation.

1.1.2 Disadvantages of Conventional Landmark-Based Navigation Systems

A typical landmark-based navigation system observes the environment in regular intervals, and places perceptual landmarks perceived at those regular-spaced locations onto a map. There are a number of disadvantages to such systems.

First of all, many perceptual landmarks look alike (e.g. many sections of a corridor appear identical to a robot's sensors). Placing them on the map wastes map space, and either leads to unwieldy maps, or maps that saturate after the exploration of only a small area.

Secondly, placing a landmark on the map merely by virtue of being located at a sampling point means that almost certainly very conspicuous (and therefore useful) landmarks will be missed, because they happen to lie between sampling points. In short, sampling at regular, user-defined intervals has definitive weaknesses.

And yet, sampling at regular intervals is the most commonly taken approach; firstly, because it is simple to implement, secondly and more importantly, because alternative selection mechanisms so far did not exist.

This report describes a mechanism for the automatic selection of landmarks by the mobile robot itself as it explores its environment. Using the mechanism, the robot selects conspicuous landmarks, which can then be mapped, while it ignores those landmarks which carry little navigational value, because of their common perceptual appearance. User guidance is not needed at any stage in this process. The main point made here is not that our landmark selection mechanism selects the *best* landmarks, but that it selects conspicuous landmarks *consistently* in several traversals of the same environment.

1.2 Landmark Selection Method Used

The basic idea behind the approach presented here is that of a filtering process. We postulate that a "good" perceptual landmark to map for navigational purposes is one that does not conform to a common model of perception, i.e. a landmark that "sticks out" and looks markedly different to commonly perceived landmarks.

To determine such perceptual landmarks, we train an artificial neural network to learn a model of the robot's "typical" perception, by making a prediction of a future perception at time $t+1$, given the current perception at time t and the knowledge that the robot is moving forwards in a straight line. What the network learns is a model of how sensory perceptions (sonar perceptions in our case) at times t and $t+1$ relate to one another.

Any location where the model's prediction is poor differs from the usual relationship between successive perceptions, and is therefore selected for mapping. Such locations are obviously often inconspicuous to the human observer, but the important aspect here is that they do not conform to the learned model of the robot's perceptions, and are therefore "unusual".

Because the robot learns a model of the relationship between successive perceptions, using its own sensors and not a model of some specific environment, this approach is environment-independent. In other words, once the model has been learned, the "landmark filter" can be used without further modification in any other environment.

1.3 Related Work

1.3.1 Landmark-based Navigation

The navigation concept for which this landmark selection mechanism is to be used is that of building topological maps, containing perceptual landmarks and the spatial relationship between these landmarks.

Franz and Mallot [Franz and Mallot, 2000] provide a good overview of biomimetic navigation systems that use topological maps (graphs) for navigation.

Owen [Owen, 2000] and Duckett [Duckett, 2000] have presented robot navigation systems which use this mechanism. Artificial neural networks (self-organising feature maps, restricted Coulomb energy networks, or adaptive resonance theory networks) cluster the robot's sonar perceptions of the world, and map them and their spatial relationships. Landmarks are added quasi-continuously, in that perceptions are clustered and mapped continuously as the robot moves through the environment. Whether landmarks are added to the map or not is largely determined by the perceptions alone (whenever a new perception is encountered, it is added to the map, if not already present). In Owen's system there is an additional mechanism that prevents very small areas or areas that are very close to each other from being mapped.

Kurz [Kurz, 1996] presents a robot navigation system that constructs a graph representation of the environment from sensor data, also using self-organising feature maps or restricted Coulomb energy networks. Again, whenever perceptions change, they are mapped, in a quasi-continuous manner.

The *ELDEN* system [Yamauchi and Beer, 1996] uses a so-called adaptive place network consisting of place units, corresponding to a region in Cartesian space, and their topological relationships. Place units are created every 100 ms, i.e. at regular spatial intervals.

The *ALICE* system [Zimmer, 1995] again uses topological maps, updating the map quasi-continuously whenever a sensory perception differs sufficiently (above a user-defined threshold) from all previously mapped perceptions.

All of these approaches have in common that they map sensory perceptions and their topological relationships, and use those for self-localisation and path planning. Landmarks are either added to the map in a quasi-continuous manner, by determining whether the current sensory perception is sufficiently different from any perception mapped already, or at regular spatial or temporal intervals, or by a combination of both methods. No specific focus is placed on using a pre-selector that differentiates between common perceptions and unusual one, before the actual mapping stage. In this sense, the work presented here is an addition to existing methods of topological map building.

1.3.2 Autonomous Landmark Selection

The advantages of landmark-based navigation are obvious, and there are many instances of such robot navigation systems (see above). Many of these systems have in common that the landmarks used are selected *a priory* by the operator, and mapped for later use. The *ServiceBot* of the GMD [Pauly et al., 1999] is a good example of such a robot; this robot pre-processes the raw sensor data, using *a priori* knowledge, before straight lines and corners are selected as landmarks for mapping. Both the *a priori* knowledge and the fact that straight lines and corners are selected are contributions from the human operator.

There are two main disadvantages to letting the user select landmarks. First, he may overlook landmarks that are easily detectable by the robot, and secondly, he may choose landmarks which are hard to detect by the robot. These two points are arguments in favour of *autonomous* selection of landmarks by the robot.

The problem of autonomous selection of landmarks for autonomous mobile robot navigation has been investigated by a number of researchers.

Thrun [Thrun, 1998] addressed the problem of learning to select an optimal set of landmarks for performing self-localisation in one specific environment, where the landmarks consisted of a projection from the robot's raw sensory perceptions (camera images) onto vectors in a lower-dimensional space. This optimisation was performed by minimising the robot's error in self-localisation. Thrun showed that his method produced better performance than localisation using designer-determined landmarks including doors and ceiling lights. A similar but computationally cheaper technique was developed by Vlassis *et al.* [Vlassis et al., 2000], who showed that their optimisation method produced better results than principal component analysis (PCA). Our approach differs in that we do not carry out any analysis of the utility of the landmarks selected, instead using a self-acquired model of "typical" sequences of perceptions which is independent of any particular task or environment.

Neither of the above approaches is suitable for *on-line* learning of landmarks, because they rely on an acquired model of a previously explored environment. Zimmer [Zimmer, 1996] considered the problem of selecting landmarks in a topological map through a process of "life-long learning", where the robot's map was continuously adapted on-line during exploration. This approach used global statistical information, based on comparison of accumulated error statistics at each of the nodes, to decide where to add and delete nodes in the map. A related idea can be found in Bourque & Dudek [Bourque and Dudek, 2000], who addressed the "vacation snapshot" problem of deciding in which locations to take camera images in order to obtain a set of images which best represents an entire environment. This approach kept running statistics on what is a "typical" perception, together with backtracking to previously visited locations which were subsequently found to be "atypical". Our method differs from these approaches in that we only use *local* sensory information to decide when to add landmarks to the map.

The closest relative of our approach is that of Simhon & Dudek [Simhon and Dudek, 1998], which also uses only local information. They addressed the problem of deciding the best locations in which to create local metric maps, using a distinctiveness function based on the robot's ability to localise itself within such a map. However, this approach relies heavily on the designer's intuition in selecting an appropriate distinctiveness function, and is suitable only for selecting appropriate landmarks for self-localisation. By contrast, our approach is task-independent, and should be useful for other applications besides navigation, such as detecting "interesting" sensory perceptions for inspection and surveillance tasks.

2 Experimental Setup

2.1 The Manchester Mobile Robot *FortyTwo*

Manchester's mobile robot *FortyTwo* is shown in figure 1.

Figure 1: THE MANCHESTER MOBILE ROBOT FORTYTWO.

It is a Nomad 200 mobile robot, equipped with 16 ultrasonic sensors, 16 infrared sensors, 20 tactile sensors, a fluxgate compass and a camera. For the experiments described in this report, sonar sensors were used.

2.2 The Bremen Wheelchair Robot

The semi-autonomous wheelchair *Rolland* (see figure 2) is used at the University of Bremen to investigate user-support measures for disabled people. The wheelchair is equipped with sonar sensors to detect obstacles.

Figure 2: THE SEMI-AUTONOMOUS WHEELCHAIR *Rolland* OF THE UNIVERSITY OF BREMEN.

The wheelchair was used in the preliminary experiments described in [Duckett and Nehmzow, 1999], which are summarised briefly in the next section.

3 Preliminary Experiments with the Bremen Wheelchair Robot

These experiments aimed at getting the robot to learn its *own* model of the time series of sonar readings using an artificial neural network. A many-to-many mapping was used, whereby the input to the network consisted of a whole set of sensor readings at time t, and the target output consisted of the next set of readings at time $t+1$. By training the network to perform the prediction task, the associations between inputs and outputs are learned from the distribution of the training data, rather than being pre-installed by the system designer.

The robot used for these experiments was the Bremen semi-autonomous navigating wheelchair discussed in section 2.2. A subset of the robot's sonar sensors were used for landmark-selection, which were located on the right-hand side of the robot. The robot's motion was constrained so that it could only move forwards in a more or less straight line.

For these experiments, the robot's sonar sensor data was recorded in four areas of the MZH building at Bremen University and logged into a data file at 10 cm intervals as it was driven around by the human operator.

The idea was to enable the network to generalise over its input space, and thus to learn a general sensor model rather than a specific model of one particular environment (see [Zimmer, 1997] for a discussion of general versus specific world models). A single perceptron was used with six inputs and six outputs, plus a bias input which was set to -1 throughout. The raw sonar readings were normalised to lie between 0 and 1 before input to the network. The network's task was to predict the sonar reading at time $t+1$, given the reading at t.

In order to eliminate noise, a filtering mechanism was used. This mechanism is discussed in section 5.2.

Results obtained were mixed, but promising enough to conduct the work discussed in this report. In particular, in one environment four peaks were observed consistently at roughly the same distances in each of three trials. There was clear correspondence between these four peaks and the positions of four sets of lift doors located along the route traversed.

4 Experiments with Manchester's *FortyTwo*

4.1 Environment and Conduct of Experiments

As the mechanism discussed here aims to detect location-specific deviations from a learned norm, simulation would be an unsuitable tool, because standard numerical models such as the one available for the Nomad 200 robot assume uniform properties of all objects in the environment. It is therefore imperative that real robot data be used.

However, because there is no closed loop between perception and motor action, it is not necessary to run all experiments on the robot. It is sufficient to log real robot data, and to process it off line.

We conducted two major sets of experiments: one, using logged data and evaluating our approach off line, and one, implementing the landmark filter on *FortyTwo*, and observing the landmark selection behaviour on the actual robot, moving in the real world. This second set of experiment was to confirm the positive results of the first batch of off-line experiments.

Both sets of experiments were conducted in the corridors of the Computer Science Department at Manchester University, an environment typical for office corridors, containing long passages, junctions, doors and items like fire extinguishers or notice boards.

4.2 Experimental Procedure for the Off-Line Experiments: Data Collection

As stated above, the landmark selection mechanism discussed here requires no direct motor response to sensory perception, as it only processes sensor signals as they are perceived. The first set of experiments, therefore, was conducted by first collecting real sonar data in two corridors of the Department of Computer Science, and subsequently selecting landmarks off line, using that data.

Readings of sonar sensors 10 to 14 (which are on the right hand side of the robot, see figure 3) were logged in two different corridors ("environment 1" and "environment 2")

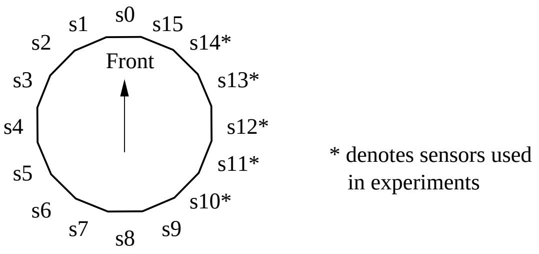

Figure 3: Arrangement of FortyTwo's Sonar Sensors.

of the Department of Computer Science at Manchester, readings were taken every 25 cm, and eight runs of 25 m length each were recorded in each environment. Thus, sixteen sets of data (two times eight) were obtained.

To ensure that data sampling points coincided, a control program was used that resulted in very similar trajectories, if the robot was started from identical starting locations. To ensure the latter, markings on the floor were used. To ensure the former, a forcefield approach was used which kept the robot in the middle of the corridor: each sonar reading was interpreted as a repelling spring force, and the resulting steering command executed by the robot was the force vector sum of all sixteen sonar forcefields. That readings were taken every 25 cm was achieved by using the internal odometry.

5 Automatic Selection of Landmarks by Means of Prediction: Method

The idea for landmark selection discussed in this paper is to make a prediction of a forthcoming sensory perception, based on the current sensory perception and an acquired model of the temporal relationship between perceptions, and to flag all those locations as landmarks that do not conform to this model.

The mechanism consists of the following stages:

- Model acquisition phase:
  1. Learning the temporal relationship between sensory perception at time t and $t + 1$, using a modified single layer Perceptron.
- Model application and landmark selection phase:
  1. Predicting the sensory perception at $t + 1$, using the model and the perception at t,
  2. Computing the sum squared error ϵ of the network's prediction,
  3. Smoothing the error curve by integrating over the last ten error values,
  4. Median filtering of the resulting error curve to remove outliers,
  5. Further smoothing by introducing a momentum term that removes smaller peaks in the error curve, and finally
  6. Detecting maxima in the error curve, which coincide with bad predictions, i.e. locations that provide "unusual" perceptions.

5.1 Modeling the Temporal Relationship between Subsequent Sensory Perceptions

In order to identify those locations in the robot's environment that do not conform to a "standard" model, that model needs to be defined first.

One possible approach to modeling sensory perception would be to establish a hand-coded numerical model, similar to generic numerical models used in robot simulation (e.g. the Nomad simulator [Nomadic Technologies, year]. Such models, however, are

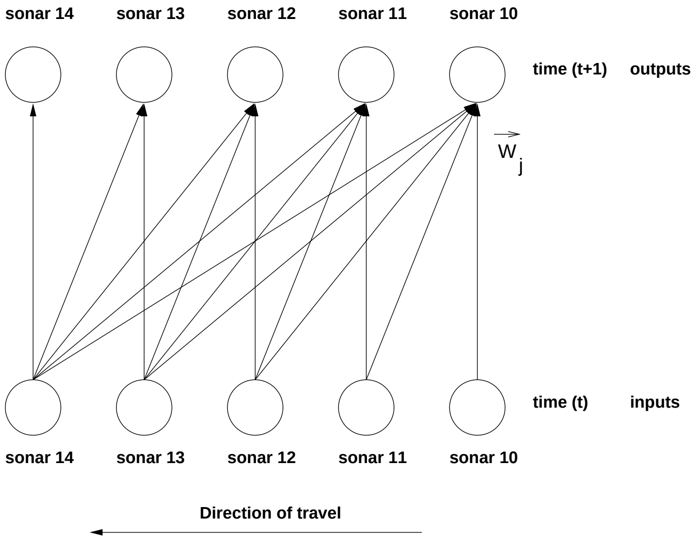

Figure 4: NETWORK ARCHITECTURE FOR THE PERCEPTION-PREDICTING PERCEPTRON.

of necessity so simplistic that they do not cover idiosyncrasies such as sonar crosstalk, specular reflection or noise, i.e. the very features the landmark filter is intended to detect. They are, therefore, unsuitable for prediction of sensory perception to a level of accuracy as would be required for automated landmark selection.

We therefore decided to let the robot itself acquire the model of the temporal relationship between sensory perceptions, associating sensory perceptions at time t with those at time $t+1$ in an associative artificial neural network. Preliminary experiments with different variants of a multi-layer Perceptron and a single-layer Perceptron proved that the latter performed best, at the lowest computational cost [Duckett and Nehmzow, 1999].

5.1.1 The Modified Perceptron Used for Predicting Sensory Perceptions

In the experiments presented here, *FortyTwo* moved along a corridor-following trajectory, using the sonar "forcefield" to maintain a stable and centred path. During this motion, the robot logged the sensory perceptions of the sonar sensors on its right hand side (sensors 10 to 14 in figure 3).

We chose a single-layer Perceptron with five input units (plus one bias input constantly clamped to '+1') and five output units, the inputs being the five sonar perceptions at time t, the outputs being the five subsequent sonar perceptions of the same five sensors at time $t+1$. This network is shown in figure 4.

This Perceptron is *not* a standard Perceptron with full connectivity, but has a reduced connectivity as shown in figure 4 to reflect the fact that the robot travels only forwards, and that therefore perceptions are first made by sonar 14, and lastly by sonar 10. Therefore, somewhat intuitively, the reduced connectivity inhibits the influence of rear sensors on the prediction of sensors which are further forward.

5.1.2 Network Training

The output o_j of a node j is given by equation 1.

$$o_j = f(\vec{w}_j \cdot \vec{\imath}), \tag{1}$$

where $\vec{w}_j$ is the weight vector of unit j, $\vec{\imath}$ the input vector to the network, and f is the sigmoid transfer function given in equation 2.

$$f(x) = \frac{1}{1+e^{-2x}}. \tag{2}$$

After presentation of a training pair (input & output), the new weight vector $\vec{w}_j(t+1)$ values can be calculated by equations 3 and 4.

$$\vec{w}_j(t+1) = \vec{w}_j(t) + \Delta\vec{w}_j(t), \tag{3}$$

$$\Delta\vec{w}_j(t) = \eta(\tau_j - o_j) \cdot \vec{\imath}, \tag{4}$$

where η is the learning rate and τ_j is the target value for output unit j.

The error ϵ of the network's performance, which was subsequently used to select landmarks for mapping, was defined as the summed squared error over all input patterns i, as given in equation 5.

$$\epsilon = \sum_i (d_i - y_i)^2. \tag{5}$$

d_i is the actually observed sensory perception at $t+1$, given a particular input pattern $\vec{\imath}$ at time t, and y_i is the network prediction of that perception at $t+1$.

To train the Perceptron in the off-line experiments, two different training procedures were used. These are described in the following section.

5.1.3 Network Training Procedure

The two times eight data sets were divided into three data sets: training, validation and test set. The data from the training set was used to train the Perceptron, using equations 3 and 4. While training was in progress, the independent validation set was used to compute the momentary network error. There will be a stage in the training process where this error decreases no further — this is the point where training is stopped. The purpose of using an independent validation set for determining the length of the training period is to avoid overtraining of the network [Bishop, 1995]. Finally, the test set was used to examine the trained network's performance on data that the net was never presented with. It was from the test set that landmarks were selected by the mechanism presented in this report.

The sixteen data sets were used in two different training arrangements:

- Training arrangement 1: training on one data set from environment 1; validation on one set from environment 2; testing on the remaining seven data sets from environment 1.
- Training arrangement 2: training on one data set from environment 2; validation on one data set from environment 1; testing on the remaining seven data sets from environment 2.

Subsequently, the arrangement which performed most effectively was selected for further testing. The training data was presented in input-output pairs (sonar values at time t and time $t+1$ respectively), where the interval between t and $t+1$ corresponds to the 25 cm interval between samples.

Using the validation data to determine the optimal number of training steps, it turned out that for both training arrangements the number of learning steps chosen was 17.

Figure 5 gives an example of the prediction error ϵ of the perceptron.

5.2 Filtering of Network Output and Selection of Landmarks

The unprocessed prediction error output of the network, as shown for instance in figure 5, is noisy, and contains many minor peaks that are so deep in the noise level that their selection as landmarks is not desired.

We therefore applied a number of filtering processes to the raw prediction error signal, before detecting peaks and thus selected landmarks. These filtering stages are discussed now.

A general overview of the entire mechanism can be seen in figure 6.

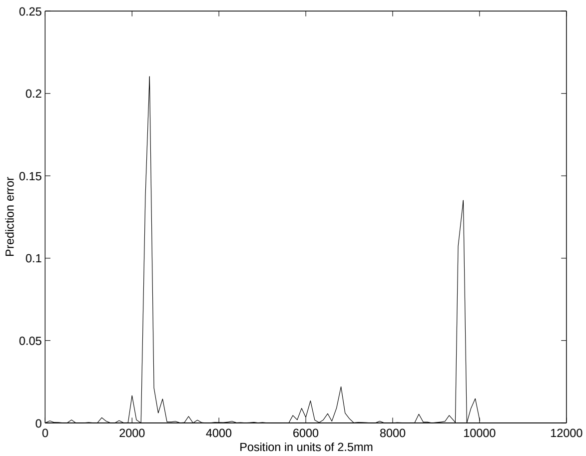

Figure 5: Perceptron stage output from test run 1 in environment 1.

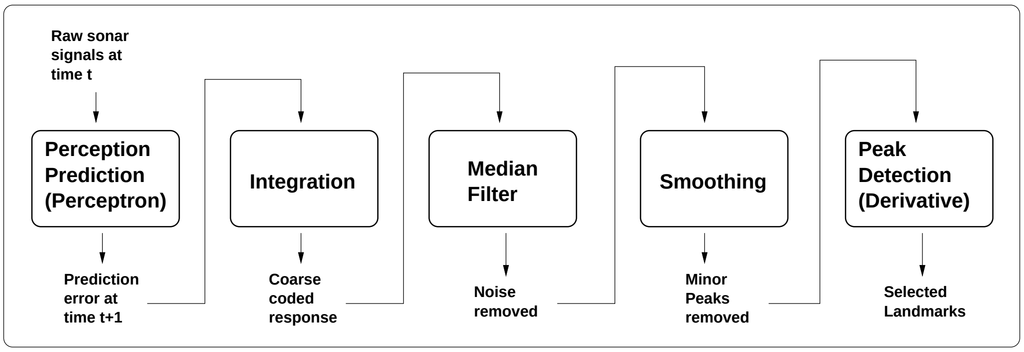

Figure 6: Overview of the 5-stage filtering mechanism.

5.2.1 Integration

The first filtering mechanism, an integration over time, was added to attenuate spurious noise from the perceptron stage. Integration is performed by simply summing up the last ten values of the raw prediction error.

Equation 6 indicates how the integrated signal F_2 is obtained.

$$F_2(t) = \sum_{i}^{10} F_1(t - i) \tag{6}$$

Figure 7 shows how the signal given in figure 5 is modified by the integration stage.

5.2.2 Median Filter

The median filter was added to reduce noise further, by removing outliers left over from the integration stage. This is performed by storing and sorting the last nine values from stage two. The median F_3 is simply derived by selecting the middle value from the sorted range of those nine value.

Figure 8 shows how the signal given in figure 7 changes through median filtering.

5.2.3 Smoothing

Looking at figure 8, one can see that the peaks are now clearly indicated, but have become fairly broad. In order to sharpen the locus of peaks, we added a momentum term that replaces each signal value $F_3(t)$ by 80% of the signal value $F_3(t-1)$, plus 20% of the current value $F_3(t)$. Equation 7 describes this filter, with λ_s chosen to be $\lambda_s = 0.2$.

$$F_4(t) = \lambda_s F_3(t) + (1 - \lambda_s) F_4(t - 1) \tag{7}$$

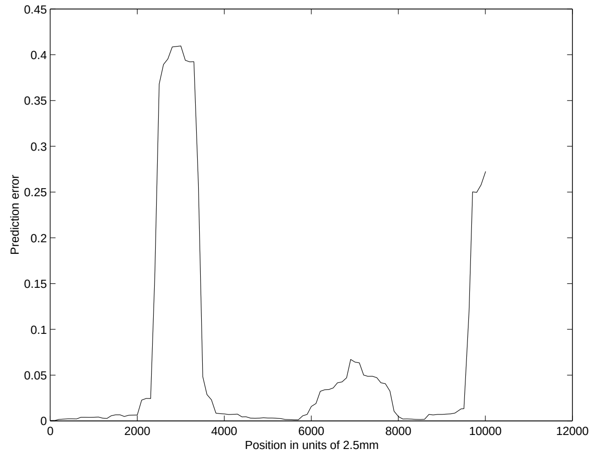

Figure 7: Integration stage output from test run 1 in environment 1. The input signal is shown in figure 5.

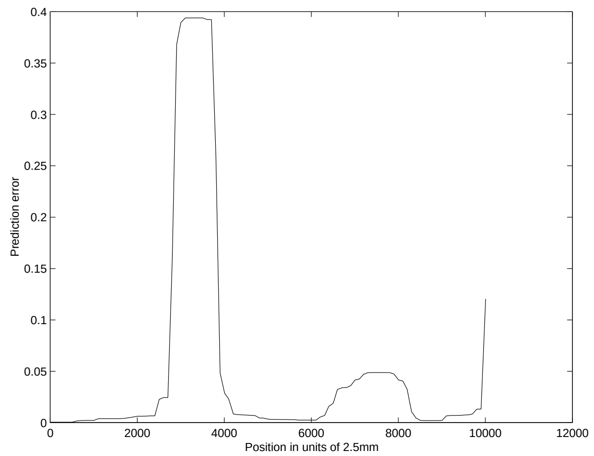

Figure 8: Median Filter stage output from test run 1 in environment 1. The input signal is shown in figure 7.

Figure 9 gives the resulting curve, with momentum applied to the curve given in figure 8.

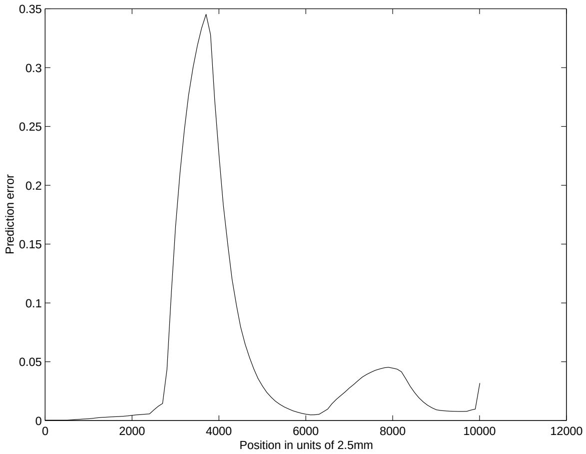

Figure 9: SMOOTHING STAGE OUTPUT FROM TEST RUN 1 IN ENVIRONMENT 1. THE INPUT SIGNAL IS SHOWN IN FIGURE 8.

5.2.4 Peak Detection and Landmark Selection

The original, noisy prediction error curve (figure 5) is now in a good shape to be analysed for landmark selection. This is done by locating all peaks in the curve.

Peak detection is done by approximating first and second derivative of the curve F_4, and selecting all its zero crossings where the second derivative is below zero (see equation 8).

$$F_4'(x) = \frac{dF_4(x)}{dx} \approx \frac{F_4(x_2) - F_4(x_1)}{x_2 - x_1} \tag{8}$$

Figure 10 shows the application of this stage to the curve given in figure 9, with the two detected peaks indicated by vertical lines. These three locations would be selected as landmarks to be mapped.

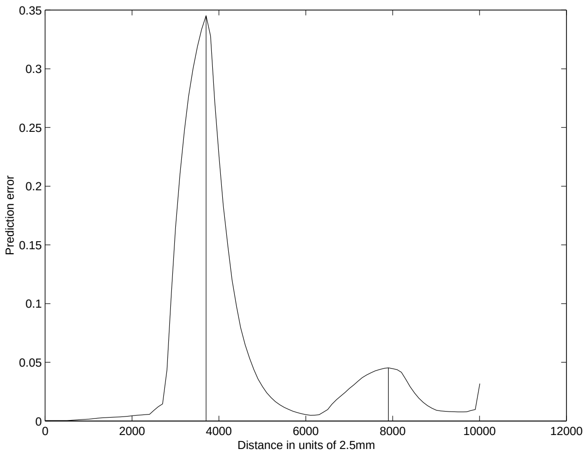

Figure 10: PEAK DETECTION STAGE OUTPUT FROM TEST RUN 1 IN ENVIRONMENT 1. THE INPUT SIGNAL IS SHOWN IN FIGURE 9.

5.3 Summary of Prediction, Filtering and Landmark Selection Mechanism

A five stage filtering mechanism has been developed in which significantly unusual features in the environment can be detected by the comparison of consecutive sonar readings taken along the robot's trajectory.

The array of sonar values is presented to the pattern associator at each step. The perceptron tries to predict the next values, generating a noisy response which is the sum-squared error of the prediction. What the artificial neural network has learned is not the features of the training environment, but the temporal relationship between subsequent perceptions of the sonar sensors used in our chosen array. This reduces the dependency on the environment which means that the prediction mechanism should be effective in new, unseen environments (see chapter 6).

The subsequent filtering stages of integration, median filtering and smoothing manipulate the raw prediction error to simplify identifying peaks via an approximation of first derivatives.

6 Experimental Results

We conducted two different sets of experiments. First, an off-line evaluation of the proposed landmark-selection mechanism was performed, using logged data obtained through *FortyTwo's* exploration of a corridor environment.

These off-line experiments confirmed the initial findings obtained with *Rolland* [Duckett and Nehmzow, 1999], i.e. that the proposed mechanism selects landmarks consistently.

In a second set of experiments, an on-line version of our landmark-selection mechanism was implemented on *FortyTwo*. The robot was then sent to explore different corridor environments within the Department, and the selected landmarks were logged for analysis.

The results of both sets of experiments are discussed in the following subsections.

6.1 Off-line Evaluation using "Real World Data"

Figures 11 and 12 show the landmarks selected in environments 1 and 2 in seven independent runs through the respective environment. As stated in section 5.1.3, two different training procedures were tested, which results in the 14 results shown for each environment.

6.1.1 Analysis of Off-Line Results Obtained in Environment 1

Looking at figure 11, one can see that landmarks at about 4000 distance units[1] and 8000 distance units along corridor 1 are selected in 13 out of the 14 cases. These two landmarks are therefore selected with very high reliability, independent of the network training regime adopted.

If the robot is trained on data from environment 1, a third landmark at about 1800 units is selected in 6 out of 7 cases. This landmark is never seen when the robot is trained on environment 2. This means that the landmark at 1800 units would not normally have been selected under the practical application conditions as we envisage them, because we argue that the robot will use a general, acquired perception model that has been learned outside the environment in which landmarks are to be selected.

Apart from the two very reliably detected landmarks, and the third landmark that is detected if the robot is trained in the same environment as the one in which it selects landmarks, there are a number of spurious peaks round about 6000 units. This is undesired, in that it means that either a landmark gets mapped that is never encountered again, or a detected landmark is not found on the map. However, the former case simply reduces the efficiency of the map through mapping some irrelevant information, while

[1] One unit is 0.254 cm, owing to the American implementation of *FortyTwo's* odometry.

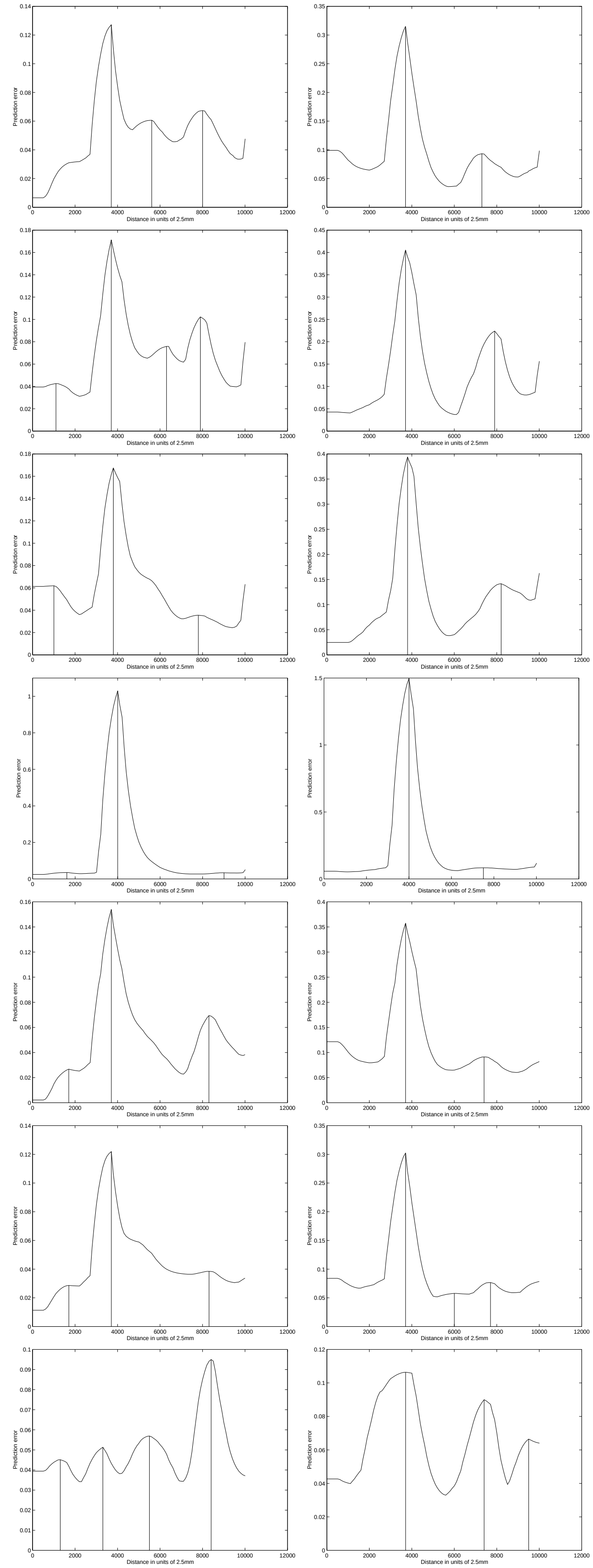

Figure 11: Peaks detected (i.e. landmarks selected) in environment 1 over 7 runs after training on data from environment 1 (left column) and environment 2 (right column). The selected landmarks are indicated by vertical lines, abscissa values indicate the robot's position, ordinate values the prediction error.

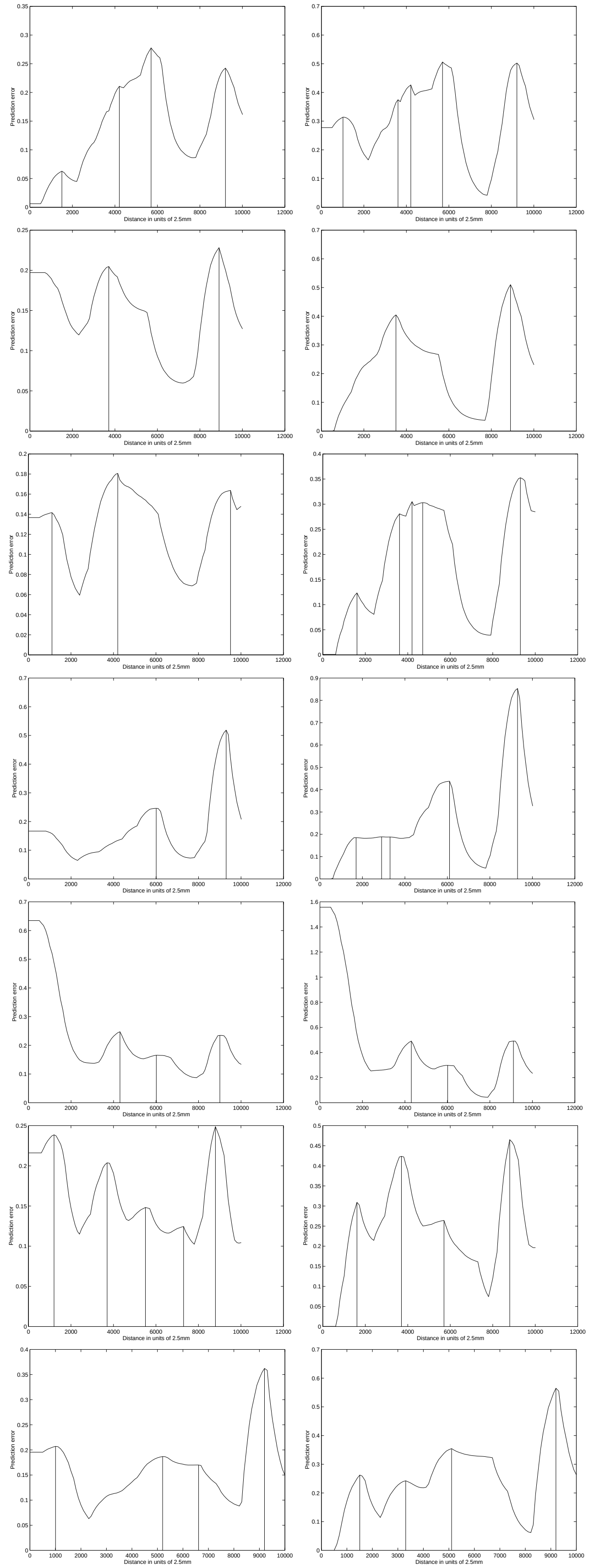

Figure 12: Peaks detected (i.e. landmarks selected) in environment 2 over 7 runs after training on data from environment 1 (left column) and environment 2 (right column). The selected landmarks are indicated by vertical lines, abscissa values indicate the robot's position, ordinate values the prediction error.

the latter case can be dealt with by ignoring selected landmarks that are not found on the map.

6.1.2 Analysis of Off-Line Results Obtained in Environment 2

Looking at figure 12, one can see that landmarks at about 1500, 3500, 5000 and 8000 distance units are selected repeatedly and reliably (in 41 cases out of 56), and that again a small number of spurious landmarks are selected (for instance at about 3000 units and 9000 units), as was the case for environment 1.

Overall, however, there is a sufficiently high reliability in landmark selection to merit the implementation on a mobile robot, for on-line evaluation. The results obtained in on-line evaluation experiments are described in subsection 6.2.

6.1.3 Correlation Between Selected Landmarks and Physical Features of the Robot's Environment

We argue that landmark selection for navigational purposes should be performed autonomously by the robot, because the robot's sensory perception and a human's perception of the same environment differ so drastically that the human is likely to a) select landmarks that are difficult to detect by the robot, and b) to overlook landmarks that are easily detectable by the machine (problem of perceptual discrepancy).

Nevertheless, it is interesting to analyse the robot's selection of landmarks with respect to the landmark's position in the explored environment. Figures 13 and 14 show environmental features (rooms with their associated room numbers shown at the top, doors and physical items in the corridor) and the prediction error observed at various locations.

Bear in mind that there is a delayed relationship between a physical feature and a selected landmark, because the various signal processing stages of the algorithm introduce delay.

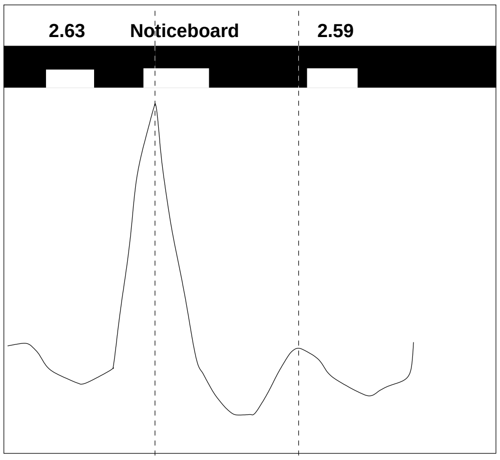

Figure 13: Correlation between Features in Training Environment 1 and Responses of the Landmark Selection Mechanism. Selected landmarks are indicated by vertical lines.

Analysing these figures, it is possible to associate some selected landmarks with physical features such as door edges of edges of a notice board, but other selected landmarks have no clear explanation to the human observer. This observation confirms our argument of perceptual discrepancy between human and robot.

6.2 On-Line Evaluation Using the Robot

Given the results described in subsection 6.1, we were interested to evaluate the landmark selection mechanism *on-line*, selecting landmarks while the robot was exploring its environment. As before, the criterion for successful performance would be the reliability

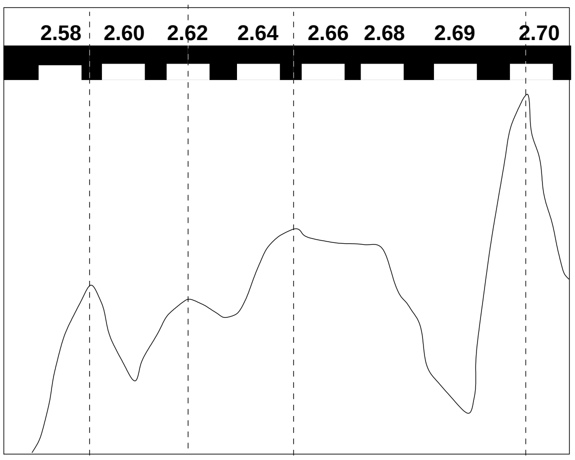

Figure 14: Correlation between Features in Training Environment 2 and Responses of the Landmark Selection Mechanism. Selected landmarks are indicated by vertical lines.

and persistence with which landmarks were selected in consecutive runs, rather than the conspicuity of a landmark to the human observer.

We therefore implemented a real time version of the landmark selection algorithm of section 5 on *FortyTwo*, and sent the robot to select landmarks in four different environments ("environment 3" to "environment 6"). Each environment was traversed 12 times, along closely similar routes (achieved by a "forcefield" corridor following program).

Whenever a landmark was selected by the algorithm, the observer was notified of this fact by a voice message, and manually entered the location into a floor plan of the corridor. The four results are shown in figures 15, 16, 17 and 18.

6.3 Analysis of On-Line Results

6.3.1 Results Obtained in Environment 3

Results obtained in environment 3 are shown in figure 15.

There is one clear landmark (detected in 10 out of 12 runs) outside room 2.32, and one elongated cluster of landmarks near room 2.38. The landmark outside room 2.30 is selected in 6 out of 12 runs, and there are spurious landmarks outside room 2.28.

This indicates that certain landmarks are detected reliably and repeatedly. More interesting, however, is the fact that there are long stretches of corridor where no landmarks at all are selected. Here, the acquired model of "normal" perceptions is so reliable that large areas of the environment are reliably classified as not worthwhile being mapped, which obviously allows the construction of more efficient maps.

6.3.2 Results Obtained in Environment 4

The results obtained in environment 4 are shown in figure 16.

Here, the landmark outside room 2.50 is detected in 12 out of 12 runs, the one outside room 2.44 in 8 out of 12 runs, and the one outside room 2.46 in 9 out of 12 runs. There are also two landmarks that are rarely detected, outside room 2.46 and room 2.48.

6.3.3 Results Obtained in Environment 5

The results obtained in environment 5 are shown in figure 17.

These results show very close clusters with a high repetition rate, they are very close to what we would consider optimal behaviour. The landmark outside room 2.72 is selected in 12 out of 12 runs, as is the landmark outside room 2.76. The landmark outside room 2.75 is selected in 9 out of 12 runs, and only the landmark at room 2.74 has a low repetition rate of 4 out 0f 12 runs.

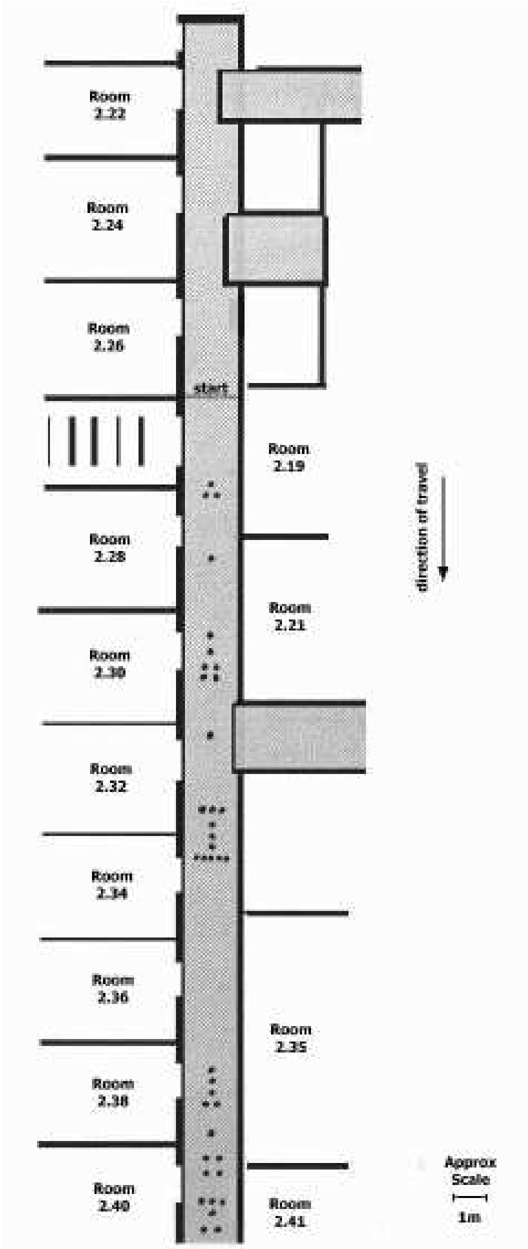

Figure 15: Test environment 3 showing peak occurrences over 12 runs. Note the long stretches without selected landmarks, indicating areas which would not be mapped, and thus indicating a gain in map building efficiency.

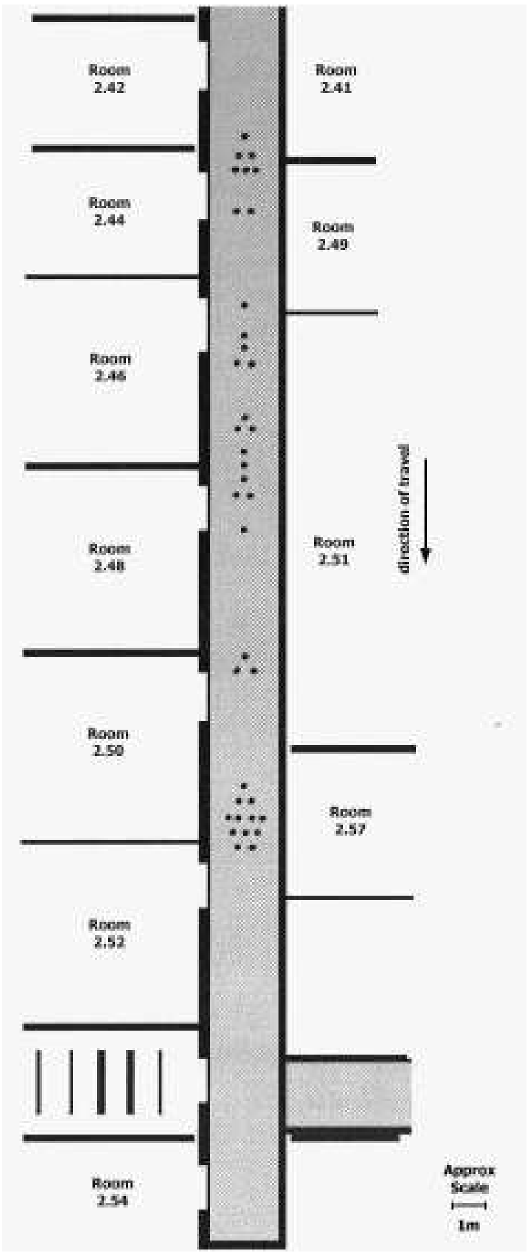

Figure 16: Test environment 4 showing peak occurrences over 12 runs.

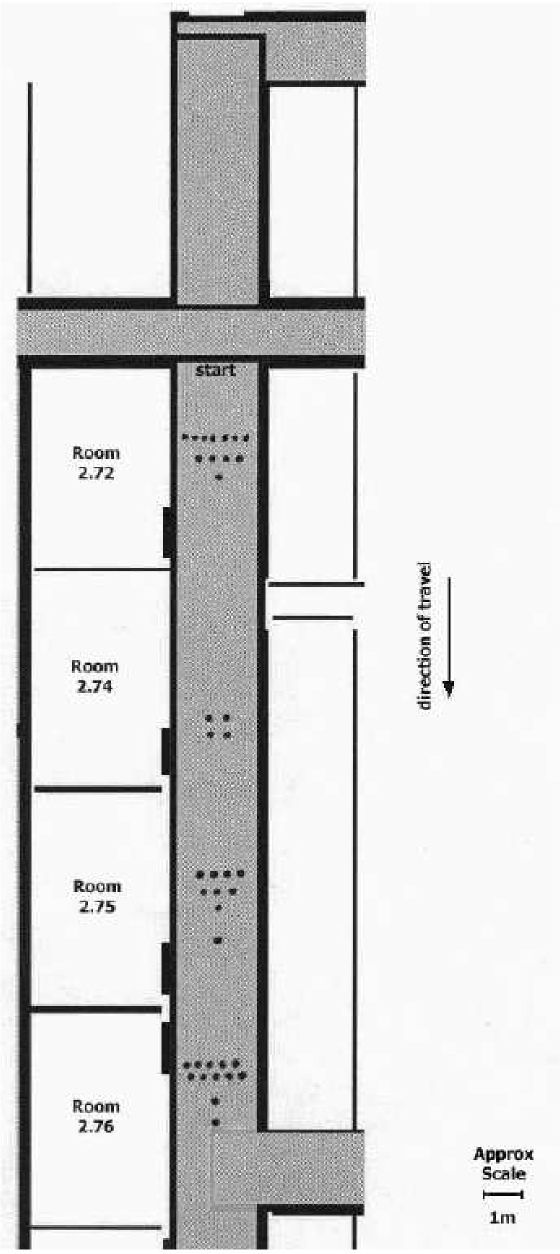

Figure 17: Test environment 5 showing peak occurrences over 12 runs.

6.3.4 Results Obtained in Environment 6

The results obtained in environment 6 are shown in figure 18.

As in environment 5, very tight and clearly defined clusters are obtained. Repetition rates are slightly lower than in environment 5. As before, it is interesting to see that large parts of the corridor are free of entries, indicating areas that are not usefully to be mapped.

7 Conclusion

7.1 Discussion

The filter proposed in this report has a number of interesting applications, apart from that of selecting landmarks for mapping, to achieve more efficient landmark-based robot navigation systems.

The purpose of the filter is to acquire a model of the "normal" temporal relationship between sensory perceptions as the robot travels through its environment. "Normality" here is defined by the input-output mapping established in the Perceptron, which is established through the presentation of many input-output pairs, and generalising that data. This perception model is then used to highlight locations where the prediction of sensory perceptions is poor, thus identifying locations that differ in their perceptual properties from "normal" locations. Such selection obviously reduces the amount of computing resource wasted on standard situation, and thus leads to more efficient and focused applications.

This is very similar to applications in novelty detection. In novelty detection, the task is to identify signals that differ significantly from what is normally perceived. One major problem in such applications is that such unusual signals are rare by definition, which makes it hard (in fact, often impossible) to train a learning system to identify such unusual signals. A common method is therefore not to pursue abnormality detection, but normality detection, as is the case here. Therefore, the landmark selection filter discussed here is applicable to a wide range of tasks in which a norm is to be established, and signals deviating from that norm are to be detected.

There is a further interesting property of the filter. Usually in landmark-based robot navigation systems, the specific mechanism (e.g. a map building mechanism) creates some kind of representation of the *environment*, facilitating navigation in that environment. This is not the case here. The landmark selection mechanism discussed in this report established a model of the *robot*, by learning the temporal relationship between subsequent sensory perceptions. This means that the filter, once learned, can be applied in any type of environment. It is therefore conceivable that it is incorporated as part of a robot's sensing hardware, and serves as a preprocessing front end to robot sensors.

In all experiments presented in this report, the robot moved along a corridor. This was done because of the kind of environments available for experimentation, not because the landmark selection mechanism requires it. The model learned by the Perceptron relates subsequent sensory perceptions, assuming that the robot travels in a straight line between perceptions. Whether these straight-line movements are executed in corridors or open spaces is of no consequence to the algorithms ability to detect locations that do not conform to the acquired model.

7.2 Future Work

Encouraged by the results obtained, we are currently investigating the performance of this algorithm on different robots, in different environments to those at Manchester.

Furthermore, we are currently investigating the effects of different smoothing functions to those mentioned in section 5.2 or different artificial neural network architectures.

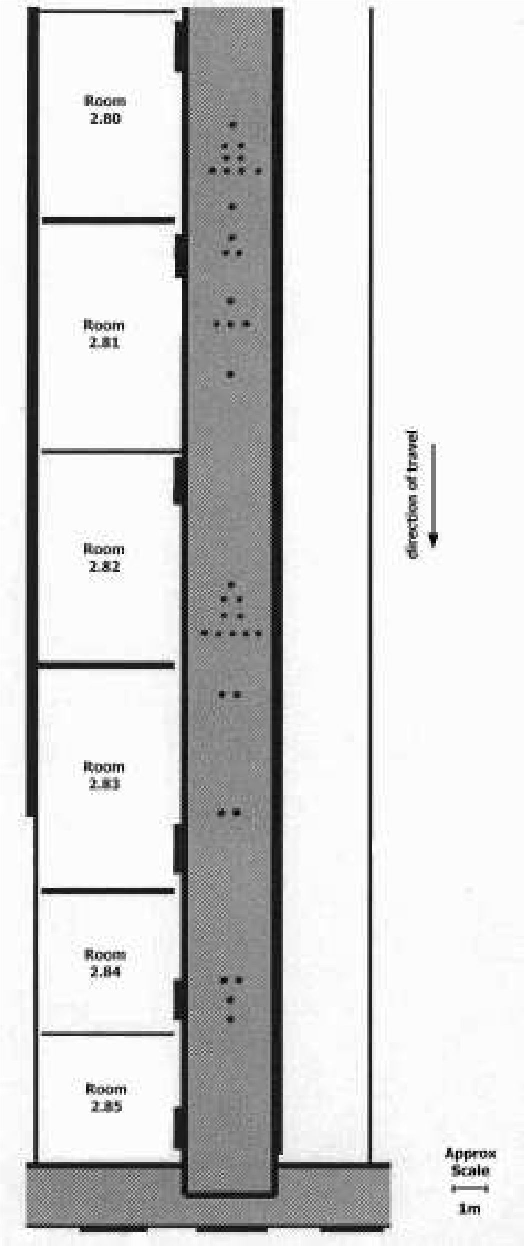

Figure 18: Test environment 6 showing peak occurrences over 12 runs.

Acknowledgements

The work reported here is supported by a grant from the British Council and the German Academic Exchange Service (grant number ARC 878), their support is gratefully acknowledged.

The preliminary results with the Bremen wheelchair were obtained during a study visit to Bremen University. We thank Bernd Krieg-Brückner and his group for the hospitality and access to the wheelchair.

References

[Bishop, 1995] Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

[Bourque and Dudek, 2000] Bourque, E. and Dudek, G. (2000). On-line construction of iconic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'2000)*, pages 2310–2315.

[Duckett, 2000] Duckett, T. (2000). *Concurrent Map Building and Self-Localisation for Mobile Robot Navigation.* PhD thesis, Department of Computer Science, University of Manchester. to appear.

[Duckett and Nehmzow, 1998] Duckett, T. and Nehmzow, U. (1998). Mobile robot self-localisation and measurement of performance in middle scale environments. *Robotics and Autonomous Systems*, 24(1–2):57–69.

[Duckett and Nehmzow, 1999] Duckett, T. and Nehmzow, U. (1999). Learning to predict sonar readings for mobile robot landmark-selection. Internal report.

[Franz and Mallot, 2000] Franz, M. and Mallot, H. (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30:133–153.

[Franz et al., 1998] Franz, M., Schoelkopf, B., Mallot, H., and Buelthoff, H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125. ISSN 0018-9219.

[Gutmann and Konolige, 1999] Gutmann, J.-S. and Konolige, K. (1999). Incremental mapping of large cyclic environments. In *Proceedings Proc. International Symposium on Computational Intelligence in Robotics and Automation, Monterey.*

[Kurz, 1996] Kurz, A. (1996). Constructing maps for mobile robot navigation based on ultrasonic range data. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 26(2):233–242.

[Nehmzow, 1999] Nehmzow, U. (1999). "Meaning" through clustering by self-organisation of spatial and temporal information. In Nehaniv, C., (Ed.), *Computation for Metaphors, Analogy and Agents.* Springer Verlag. Lecture Notes in Artificial Intelligence 1562.

[Nehmzow and Owen, 2000] Nehmzow, U. and Owen, C. (2000). Robot navigation in the real world: Experiments with Manchester's FortyTwo in unmodified, large environments. *J. Robotics and Autonomous Systems*, 33. to appear.

[Nomadic Technologies, year] Nomadic Technologies (no year). Nomad simulator. `http://www.robots.com`.

[Owen, 2000] Owen, C. (2000). *Map-Building and Map-Interpretation Mechanisms for a Mobile Robot.* PhD thesis, Department of Computer Science, University of Manchester. to appear.

[Pauly et al., 1999] Pauly, M., Finke, M., Peters, L., and Beck, K. (1999). Echtzeitfähige Positionskorrektur auf Basis natürlicher Landmarken. In Schmidt, G., Hanebeck, U., and Freyberger, F., (Eds.), *Autonome Mobile Systeme*, pages 98–107. Springer Verlag, Berlin, Heidelberg, New York.

[Simhon and Dudek, 1998] Simhon, S. and Dudek, G. (1998). Selecting targets for local reference frames. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98)*, pages 2840–2845.

[Thrun, 1998] Thrun, S. (1998). Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1).

[Vlassis et al., 2000] Vlassis, N., Motomura, Y., and Krose, B. (2000). Supervised linear feature extraction for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'2000)*, pages 2979–2984.

[Yamauchi and Beer, 1996] Yamauchi, B. and Beer, R. (1996). Spatial learning for navigation in dynamic environments. *IEEE Trans. Systems, Man and Cybernetics, Part B*, 26(3).

[Zimmer, 1997] Zimmer, U. (1997). General and specific models in complex robotics systems - a critique and a proposal. Technical Report 1063, GMD.

[Zimmer, 1995] Zimmer, U. R. (1995). Self–localization in dynamic environments. In *Proceedings IEEE/SOFT International Workshop BIES'95, Tokyo.*

[Zimmer, 1996] Zimmer, U. R. (1996). Robust world modelling and navigation in a real world. *Neurocomputing, Special Issue 1996*, 13(2–4).