Learning to Autonomously Select Landmarks for Navigation and Communication

Jason Fleischer **Stephen Marsland**[*]
Dept. of Computer Science
University of Manchester
Manchester M13 9PL, UK
{jfleischer,smarsland}@cs.man.ac.uk

Abstract

Selecting landmarks for use by a navigating mobile robot is important for map-building systems. However, it can also provide a way by which robots can communicate route information, so that one robot can tell another how to find a goal location. A route through an environment can be described by the landmarks encountered along the path, and a robot following the same path must identify the perceptions corresponding to the actual landmarks in the description in order to localise itself. This paper presents an algorithm to automatically select landmarks, choosing as landmarks places that do not fit into a model of typical perceptions acquired by the robot. Four methods of aligning the landmarks between different runs on the same route are also presented. The different alignment methods are evaluated according to both how well they produce matching landmarks and how suitable such alignment methods would be for use in a route communication system.

1 Introduction

Perceptual landmarks – navigational landmarks based on the sensory perceptions of the robot – form the basis of many successful mobile robot navigation systems, see for example (Yamauchi and Beer, 1996, Duckett and Nehmzow, 1998). These landmarks are used because they avoid the problem of drift error that is inherent in odometry measurements.

The selection of landmarks is important for other applications, too. The problem of robot communication has become increasingly important over the last few years, as multiple robot systems have become more and more common. One question that has begun to be addressed is how a group of robots can share navigational information, for example so that one robot can tell another how to find a goal location. A possible way to approach this question is to send information about the landmarks that are seen along the route and the actions that were taken at these landmarks. Such a system would require the following capabilities:

- A landmark selection method
- A mapping from selected landmarks to words to be communicated
- An inverse mapping from words to landmarks

It is the first of these requirements that is addressed in this paper. Wherever navigational landmarks are used it is important that they are detected consistently, so that the same landmarks are found on every run through an environment. Otherwise, the route that should be taken is unclear and, for communication, the physical grounding of the symbols that the robots were communicating with would be very different.

In previous work (Marsland et al., 2001) it was shown that a robot could adaptively learn a model of its environment by using its current sensory perceptions to predict the next set of perceptions. It was argued that the places where the prediction was not accurate – those places that were not adequately described in the acquired model – were suitable choices for landmarks. These places were found by using a Kalman filter to monitor the error curve of the output of the neural network.

In that work, which is described in more detail in section 3.1, the robot travelled a constant distance (20 cm) between sensor scans of the environment, and stopped in order to make the scans. This meant that interesting landmarks could be missed if they were placed between the sampling steps. Here we investigate a more complex version of that task, where the robot samples the environment continually as it explores, even while turning corners. This means that the only limit on the number of samples taken by the robot as it travels is the operating capabilities of the robot.

We compare the quality of landmark matches between successive runs in an environment by looking at how well

[*]Authors are given in alphabetical order, both being principal authors

the landmarks match in terms of preserving landmark sequence, distance, and category between the two runs. The results are promising.

2 Relevant Literature

2.1 Landmark Selection

The most common method of selecting landmarks has been to take sensory scans at regular intervals with every scan being used as a landmark that is put into the map. This avoids the problem of ensuring that landmarks should be detected consistently, but since in many environments – such as corridors – a large number of perceptions are almost identical, the map is filled up with information that does not aid navigation.

A number of researchers have considered selecting suitable landmarks for mobile robot navigation. One technique is to ask the user to define the landmarks before the robot explores. Humans typically select objects that they find easy to recognise, such as doors, or line segments extracted from camera images recorded as the robot travels (Kortenkamp and Weymouth, 1994). However, there is no guarantee that these objects are easy for the robot to recognise.

(Thrun, 1998) approached the problem through Bayesian learning, aiming to select an optimal set of landmarks for performing self-localisation in one specific environment. The landmarks were made up 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 directly the quantity of interest, namely 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., 2000), who showed that their optimisation method produced better results than principal component analysis. Our approach differs in that we do not carry out any analysis of the utility of the landmarks selected, but instead use a self-acquired model of 'typical' sequences of perceptions, which is independent of any particular task or environment.

(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 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 that best represent an entire environment. This approach kept running statistics on what is a 'typical' perception, together with backtracking to previously visited locations that were subsequently found to be 'atypical'.

2.2 Landmark Communication

In the route communication task, landmarks can be considered as categories that the robot recognises and can communicate about. The problem of communicating landmarks between robots then becomes one of learning a mapping between the different categorisations that are privately held by each robot in the communicating community. Although it is not difficult for agents to communicate once they posses the same categorisations of the environment, the methods by which agents can acquire matching categorisations and learn to communicate them is still the topic of ongoing research. The most common method of learning to communicate categorisations is to assume that categorisation is a process that is private to each individual agent, and communication consists of learning a public code, or set of symbols, that maps meanings between the internal categories of each agent. Early work by (Yanco and Stein, 1993) showed how two robots could use reinforcement learning to create an encoding that enables the robots to communicate the simple categories 'turn left' and 'turn right'. (Billard and Dautenhahn, 2000) demonstrated that imitation is a plausible mechanism for robots to learn a common symbol-meaning system describing various perceptions in the environment. (Steels, 1996) proposed a formalism to both learn perceptually grounded meanings of such encodings and a means by which agents can modify their internal categorisations to increase their success rate in communication. An alternative approach is to produce a non-adaptive system that maps categories onto symbols. For instance, (Skubic et al., 2001) showed how a set of fuzzy logic spatial relationships can be combined with a sensor prototyping method called the histogram of forces to generate linguistic descriptions of a robot's environment that are understandable to human users.

The problem of learning symbols to describe a robot's route through an environment has not, to our knowledge, yet been addressed in the literature. The closest related work is that done on segmenting a robot's sensory flow into categories by (Tani and Nolfi, 1999) and (Linaker and Niklasson, 2000). Both of these works mention the possibility of communicating routes between robots as sequences of categories, but do not actually attempt the task. Our own work (Fleischer and Nehmzow, 2001) has demonstrated that it is possible for two robots to learn individual categorisations of locations in the environment and then an encoding from their categorisations onto a set of public symbols, enabling the robots to agree on symbols representing locations in the environment.

3 Approach

3.1 Overview

We suggest that the perceptions that are most suitable as landmarks are those that differ in some way from the main run of perceptions. This means that we are attempting a form of novelty detection, highlighting perceptions that are in some way unusual. The first part of any novelty detection process (Marsland, 2001) is to learn the model of normality. This is done by allowing the robot to travel through the environment using a wall-following behaviour and collecting data from its sonar sensors. These readings are used as inputs to train a single-layer neural network, described in section 3.2. Landmark detection is performed by finding peaks in the error curve of the single-layer network. This is a fairly difficult problem as the residual noise level is high. A one-dimensional Kalman filter is used, as is described in section 3.3.

3.2 The Sensor Prediction Network

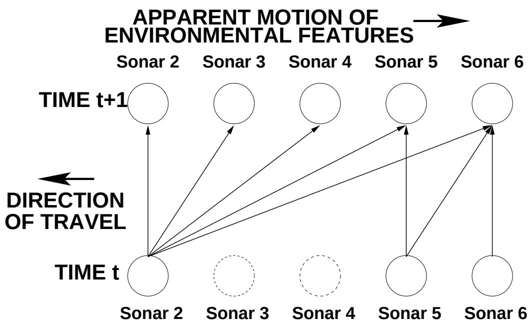

Figure 1: The single-layer network used to learn the mapping between the current sensory perception and the next. To aid clarity, connections for nodes drawn dotted are not shown.

A single-layer neural network with a sigmoidal activation function, shown in figure 1, is trained to acquire a model of the relationship between successive sensory perceptions. The network structure is not fully connected; the perceptions of sensors facing forwards are used as inputs to model rear-mounted sensors, but not vice-versa. This reflects the fact that the robot drives forward, and so the future perceptions of any one sensor depend only on the sensors in front of it. While this assumption about the sensors breaks down when the robot turns corners, this does not seem to be a problem. In addition to the network inputs shown in the figure, a bias input that is permanently set to -1 is used.

The network weights, W, are adapted during a training phase, where the robot explores an environment and records its sensory perceptions. The sensory perceptions at each time step, $p(t)$, are used as both network inputs (at the current time step) and training examples for the next time step. The weights are adapted using the standard Widrow-Hoff learning rule:

$$\Delta W(t) = \eta \left(p(t) - W(t)p(t-1)\right)^{\mathrm{T}} p(t-1) \, . \quad (1)$$

The single-layer network was trained using early stopping on a training set made of runs through a number of corridor environments, using a learning rate of $\eta = 0.2$. Further details are given in section 4.

Once the training is completed the landmark selector can be used on-line. The error curve of the neural network is monitored, where the error is the squared difference between the predicted output, $o_k(t)$, and the actual perceptions of the robot, $p_k(t)$, for each sensor k:

$$E = \sum_k \left(o_k(t) - p_k(t)\right)^2 . \quad (2)$$

3.3 Landmark Selection Using a Kalman Filter

The criteria that is used for selecting a landmark is that the output error of the sensor prediction network is sufficiently high. A Kalman filter (Kalman, 1960) provides a principled method for determining when this is true. The Kalman filter is a method for recursively estimating the state of a discrete-time controlled process that is governed by a linear stochastic difference equation. In our case we have a scalar variable, the error of the network, E, which would be a constant value, hopefully near zero, if the sensor values remained unchanging. However, there is noise in E from both sensor repeatability (measurement noise) and robot orientation and position (process noise).

The Kalman filter attempts to optimally re-estimate the variable E at each step along the way so at to remove the effects of both measurement and process noise, leaving behind only the prediction error actually produced by variation of the environment. The filter equations compute a gain, K, that is used to recursively update estimates of the true error, $\widehat{E}$, and its variance, v. The Kalman filter equations for our simple, one-dimensional case are:

$$K(t) = \frac{v(t-1)}{v(t-1) + R}, \quad (3)$$

$$\widehat{E}(t) = \widehat{E}(t-1) + K(t)\left(E(t) - \widehat{E}(t-1)\right), \quad (4)$$

$$v(t) = (1-K)\left(v(t-1) + Q\right), \quad (5)$$

where Q is the process noise variance and R is the measurement noise variance. The value of Q is estimated as

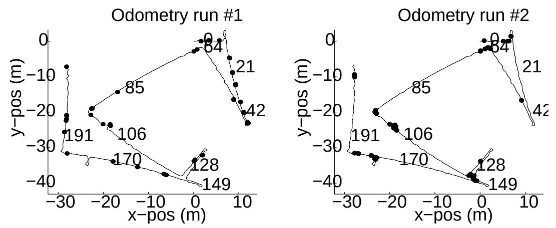

Figure 2: The difference between a constant valued measurement noise variance (run 2, right) and the time-varying function of variance calculated via the Delta method (run 1, left). The line shows the robot path as measured by the robot odometry, and the points are the places that were selected as landmarks. The numbers refer to their distance from the start of the run. The time-varying noise model shows only 74 landmarks while the constant model finds 181 using the same data. The mean distance between landmarks in the time-varying model was 2.86 m but only 1.16 m in the constant one.

the variance of $\widehat{E}$ during training, and R can either be assigned to be the same value, set to 0, or approximated by measuring the variance of the sonar sensors as the robot travels in a short, perfectly straight stretch of featureless hallway, and propagating this variance through the nonlinear transfer functions of the neural network and sum-of-squares error using the Delta Method (Rice, 1994) to generate a time-varying noise model. Figure 2 shows the effects that this choice makes on a typical set of sonar data.

The time-varying noise model produces fewer, more distinct landmarks than either $R = 0$ or $R = Q$ noise models. Most of the extra landmarks found by the constant-valued noise model come from many successive perceptions at the same location being counted as landmarks, whereas the time-varying model often prevents this problem from occurring. We used the time-varying noise model in all of the experiments reported here.

As well as trying to remove noise from the error estimate, the Kalman filter also conveniently maintains an estimate of the error variance. This variance estimate can be used to determine if $\widehat{E}(t)$ is greater then some number n of standard deviations away from the mean of the Kalman estimated error at the current time, $\bar{E}(t)$. Therefore, we can define a landmark as any perception where,

$$\widehat{E}(t) > \bar{E}(t) + n\sqrt{v(t)}\ . \tag{6}$$

The parameter n provides a method for adjusting the required level of conspicuousness and therefore relative frequency of landmarks. Typically, we find that values of n between four and five work well.

3.4 Categorising Landmarks

Two of the four methods of aligning landmarks between different runs that we used employed landmark categories. Landmark categories are also of more general interest because they can form the basis for communication about landmarks between different robots (Fleischer and Nehmzow, 2001). We used Kohonen's Self-Organising Map (Kohonen, 1982), an unsupervised neural network based on vector quantisation, to perform categorisation of the landmarks. The Self-Organising Map (SOM) is a neural network that maps a set of inputs onto activations of a set of output nodes arranged on a lattice. The network categorises the robot's sensor perceptions by feeding them as inputs into the network and taking the identity of the output node with the highest activation level (the 'best-matching node'), as the category of the input. The SOM is trained by presenting it with a data set and adapting the network weights such that, for each input in the data set, the region in the output map around the best-matching node is moved closer to the presented input.

We used the SOM Toolbox (Vesanto et al., 2000) to produce the SOMs used in this work. The toolbox will automatically select the size and shape of the map and the training parameters based on the number of inputs, the number of data points, and the principal components of the training data set. This feature was employed here. For the training data sets used in this work, two-dimensional, toroidal SOMs with between 24 and 66 output nodes were produced, with one dimension of the map being roughly twice as long as the other.

4 Experiments

4.1 Description

The experiments in this paper were performed on a Nomad Scout, a differential drive robot with 16 Polaroid sonar sensors capable of giving range information on objects between 15 cm and 6 m away from the robot. The sensor values were updated as quickly as the processing speed of the PC controller would allow, giving less temporal structure to the data than there would be if the sensors were only updated at fixed distance or time intervals. Each experiment run consisted of between 30 and 200 metres of travel in normal office building corridors that have not been modified for the robot's use in any way. These runs were made during daytime hours with normal use being made of the corridors; no attempt was made to remove the anomalies in the data created by people walking through the sensor range, as these would be the conditions in which the system would have to work as part of a navigation or communication system.

The robot used a hard-wired wall-following program to follow the left-hand side wall at a constant distance.

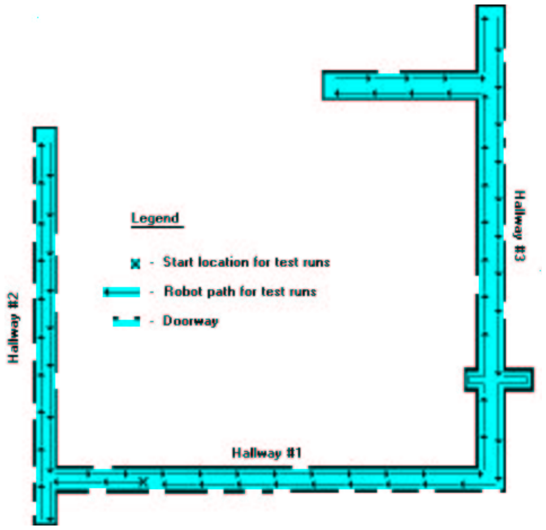

Figure 3: The hallway environment. The robot was trained in either any one of these hallways or in all three. During testing runs the robot travelled through all three hallways. Each run consists of a complete circuit of the hallway(s); the robot executed a left-hand side wall-following program until it arrived back at the location it had started from.

Noise in the system comes principally from inaccuracies in the wall-follower and error in the sonar sensors.

Three different, adjacent, hallway sections were used for training the landmark detection algorithm (see figure 3). The robot travelled through each hallway three times, thus producing a total of nine different data sets for training the algorithms. The training data were used to do three things: (1) train the sensor prediction model, (2) calculate values of Q and R for the Kalman filter and (3) train the SOM for landmark categorisation. The testing data was produced by propping open the doors between the three hallways and collecting a further three data runs using a 200 m long route through all three hallways. To compare the alignment between landmarks the three test sets were paired in all six possible permutations (since order of comparison matters) and the measures described in the next section were calculated.

4.2 Analysis

We investigated four different methods of producing alignments between landmarks in two runs. These different alignment methods reflect some possible ways that a robot might use to align its own landmark perceptions with a landmark-based route description. The alignment methods used match landmarks between runs using sequence, distance travelled, and landmark category, or a combination of them. In each alignment the first run is taken as the route description received by the robot, which is trying to match up the the second run (the landmarks it has perceived) against the original. The alignment methods can be described as:

Sequential Each landmark in the second run is aligned with the next landmark in the first run. If there is a mismatch between the number of landmarks there is no alignment for the excess landmarks.

Distance The $x - y$ position odometry is transformed into a single dimension of distance travelled from the beginning of the run. Landmarks in the second run are aligned to the landmarks in the first run that are closest to them in distance travelled.

Category and distance Each landmark in the second run is aligned to the landmark in the first run that is closest to the same distance travelled and is also a member of the same landmark category. If there is no category match then no alignment is made for that landmark.

Category and distance with limited range As previously, but matches are only allowed that are at the same distance travelled plus or minus an error term representing odometric drift. In these experiments the allowed misalignment, ν_k increases with distance travelled, d (in metres), $\nu_k = 0.25 + 0.05d$, for each landmark k.

Alignments between test runs can be simply evaluated by looking at actual landmark matches. Examples are shown in figures 4 to 7. In addition to these figures three metrics were devised to evaluate the alignments according to the criteria they were based on:

Category score The number of assignments where the aligned landmarks in both runs share the same category, divided by the number of alignments.

Distance score The mean of $\exp \frac{-\delta_k^2}{2\nu_k^2}$ for all aligned landmarks, where δ_k is the difference between the recorded odometry distance for landmark k and its aligned partner.

Sequential score The fraction of alignments where the aligned partner of landmark $k+1$ is the same or further distance than the aligned partner of landmark k. Landmarks without assigned alignments are not counted as a break in the sequence.

Note that these metrics do not penalise alignments that produce fewer matches of better quality. This is based on the assumption that navigation with many landmarks of uncertain quality is more difficult than navigation with fewer, better quality landmarks, i.e., landmarks that are found consistently.

Alignment	Score	Left Lmks μ	Left Lmks σ	All Lmks μ	All Lmks σ	Left All μ	Left All σ
Sequence	Cat	0.24	0.16	0.15	0.10	0.20	0.14
	Dist	0.30	0.19	0.30	0.19	0.30	0.19
	Seq	1.00	0.00	1.00	0.00	1.00	0.00
Distance	Cat	0.27	0.18	0.19	0.11	0.23	0.16
	Dist	0.82	0.14	0.82	0.14	0.82	0.14
	Seq	1.00	0.00	1.00	0.00	1.00	0.00
Category	Cat	1.00	0.00	1.00	0.00	1.00	0.00
+	Dist	0.64	0.12	0.63	0.12	0.62	0.12
Distance	Seq	0.87	0.05	0.81	0.04	0.87	0.05
Category,	Cat	1.00	0.00	1.00	0.00	1.00	0.00
Distance,	Dist	0.85	0.11	0.86	0.10	0.85	0.10
Rng lim	Seq	0.94	0.05	0.90	0.05	0.95	0.04

Table 1: Alignment scores for a landmark selector ($n = 5$), training the SOM with three different types of input: 'Left Lmks' left-side sonars from landmark perceptions, 'All Lmks' all sonars from landmark perceptions, or 'Left All' left-side sonars from all perceptions. The scores represent mean and standard deviation values of the 54 possible permutations in the training/testing sets. Two-sample t-tests show that the 'Left All' and 'Left Lmks' parameters produce better scores than 'All Lmks', but are indistinguishable from each other.

5 Results

We investigated how suitable the four alignment methods described previously were for use in the route description task using the three metrics. The way in which the SOM was trained was also varied. We tested using all of the 16 sonar sensors as inputs, or just the five left-hand sensors that were used as inputs to the single-layer network. We compared using a SOM trained only on the sensor inputs that were selected as landmarks with one trained on every set of sensor readings. Finally, we compared training in hallways shorter then the test environment with training in hallways of the same size.

5.1 Alignments and Scores

It is useful to note some general trends and features of the alignments and scores that are presented in this section. Table 1 shows mean values for each score and alignment combination. As would be expected, the sequence and distance alignments have perfect sequential scores, but they produce very poor categorisation scores. This is presumably because they are matching landmarks that are close together, either in the sequence or by distance, but that are caused by different features. The sequential alignment also produces the worst distance score. This occurs because the number of landmarks can vary by 100%, producing misalignments of up to 80 m, as can be seen in figure 7. The category-based algorithms, naturally, produce perfect category scores. While the category and distance alignment produces good alignments in many cases, it can periodically match up landmarks at opposite ends of the environment. The range-limited version was devised to overcome this problem, and is very successful in doing so; it produces many fewer landmark alignments, an average of 50, as compared with 122 for the other alignments. It succeeds in producing the highest quality and most desirable matches when there are small clusters of landmarks in the same location. Another benefit is that the alignments it produces are relatively insensitive to the order of comparison, unlike the other alignment methods.

5.2 Effects of SOM Inputs

Table 1 and figures 4 to 6 show the alignments and scores for three choices of SOM training inputs. We compared the scores with a two-sample t-test ($\alpha = 0.05$) to determine what effects SOM training inputs had on alignment performance. Using only the left-side sonars produced a detectable improvement in the sequential score on both category-based alignment methods. It is possible that the slightly improved performance for left-side only sonars is because there was not space for someone to walk between the robot and the wall it was following (left), and so the sonar returns were not confused by people walking past the robot. We also compared training the SOM with only landmark perceptions versus all sonar scans. In this case, there were no statistically significant differences ($\alpha = 0.05$) in any of the scores.

5.3 Effects of Training Environment

The landmark selection algorithm sets parameters related to the noise models and the SOM learns to categorise perceptions based on the training environment. If there is enough difference between training and testing environments we might expect poor landmark selection and alignment. Of course it is unlikely that it would be possible to train in the entire environment that the robot will need to navigate in; often at best a subset, or a related environment might be all that was available. Therefore, the assumed standard in table 1 is to train in only one hallway and test in runs spanning all three hallways. However, comparing those values to training on three-hallway runs using the same training parameters we can see no statistically significant differences in any score for any alignment ($\alpha = 0.05$). Qualitatively, the difference can be seen by comparing the alignments shown in figure 6, trained in one hallway, with the alignments in figure 7, trained in all three. Both figures show alignments generated on the same test data. But there are fewer landmarks detected when training occurs in the same hallways as testing, and while there are fewer landmark alignments numerically, a higher proportion of those landmarks are assigned matches in the category-based alignment methods.

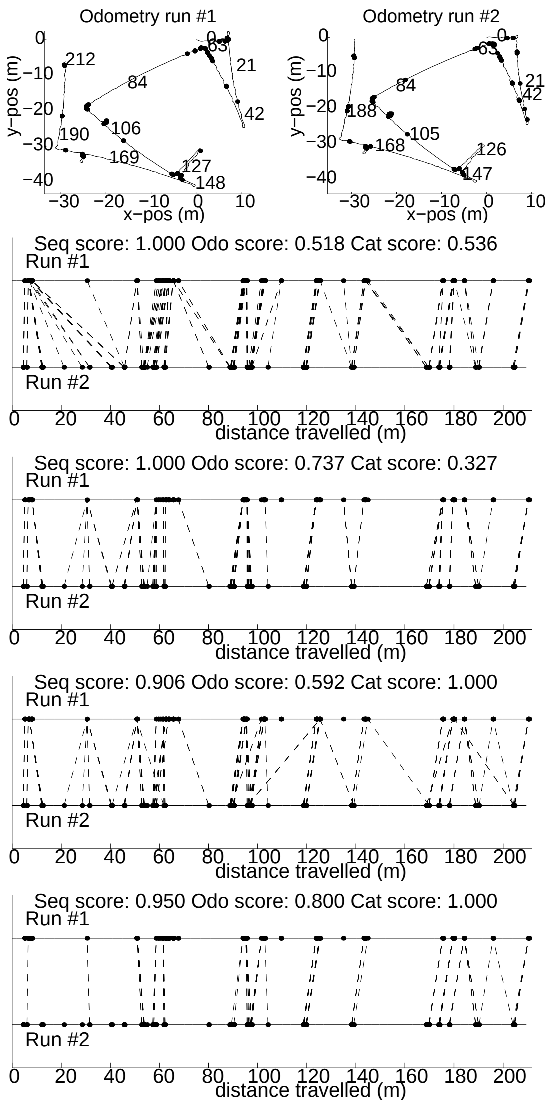

Figure 4: Example alignments for a landmark detection system trained on a run in hallway 1 ($n = 5$, using left-side sonars at landmark perceptions for SOM training). The system was then tested on two separate runs that spanned all three hallways. This set of SOM training parameters produces the best alignment scores, along with the ones in figure 6, which are statistically indistinguishable from this set (see table 1).
Robot path is represented by solid lines, landmarks by black dots, and alignments by dashed lines. *Top:* The robot's internal odometry showing $x - y$ position. The numbers on the path are the total distance travelled in the to that point. *Underneath:* The $x - y$ odometry is projected onto a distance-travelled axis, and aligned landmarks have dashed lines drawn between them. The alignments are (from 2nd from top to bottom) sequential, distance, distance & category, and distance & category with range limit.

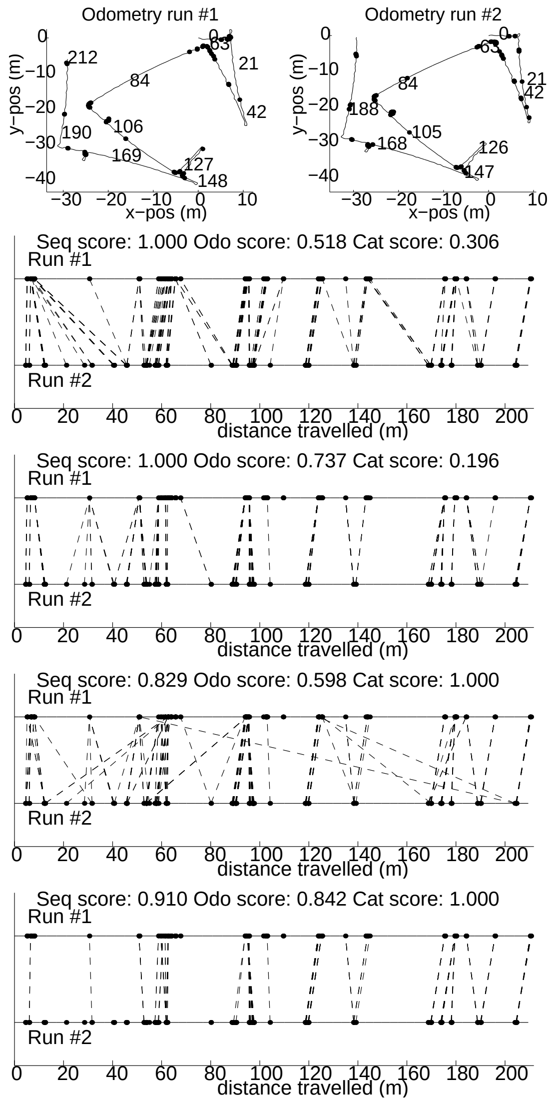

Figure 5: Example alignments for a landmark detection system trained on a run in hallway 1 ($n = 5$, using all 16 sonars at landmark perceptions for SOM training). The system was then tested on two separate runs that spanned all three hallways. Using all the sonars like this is less successful than using only the sonars on the left of the robot, as can be seen in table 1.
Robot path is represented by solid lines, landmarks by black dots, and alignments by dashed lines. *Top:* The robot's internal odometry showing $x - y$ position. The numbers on the path are the total distance travelled in the to that point. *Underneath:* The $x - y$ odometry is projected onto a distance travelled axis, and aligned landmarks have dashed lines drawn between them. The alignments are (from 2nd from top to bottom) sequential, distance, distance & category, and distance & category with range limit.

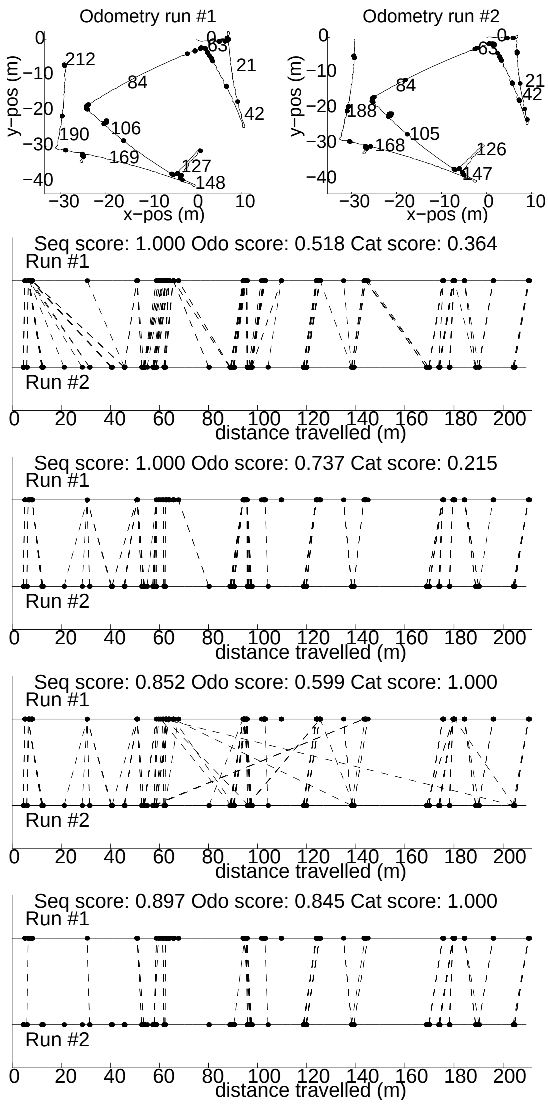

Figure 6: Example alignments for a landmark detection system trained on a run in hallway 1 ($n = 5$, using left-side sonars at every perception for SOM training). The system was then tested on two separate runs that spanned all three hallways. This set of SOM training parameters produces the best alignment scores, along with the ones in figure 4, which are statistically indistinguishable from this set (see table 1).
Robot path is represented by solid lines, landmarks by black dots, and alignments by dashed lines. *Top:* The robot's internal odometry showing $x - y$ position. The numbers on the path are the total distance travelled in the to that point. *Underneath:* The $x - y$ odometry is projected onto a distance travelled axis, and aligned landmarks have dashed lines drawn between them. The alignments are (from 2nd from top to bottom) sequential, distance, distance & category, and distance & category with range limit.

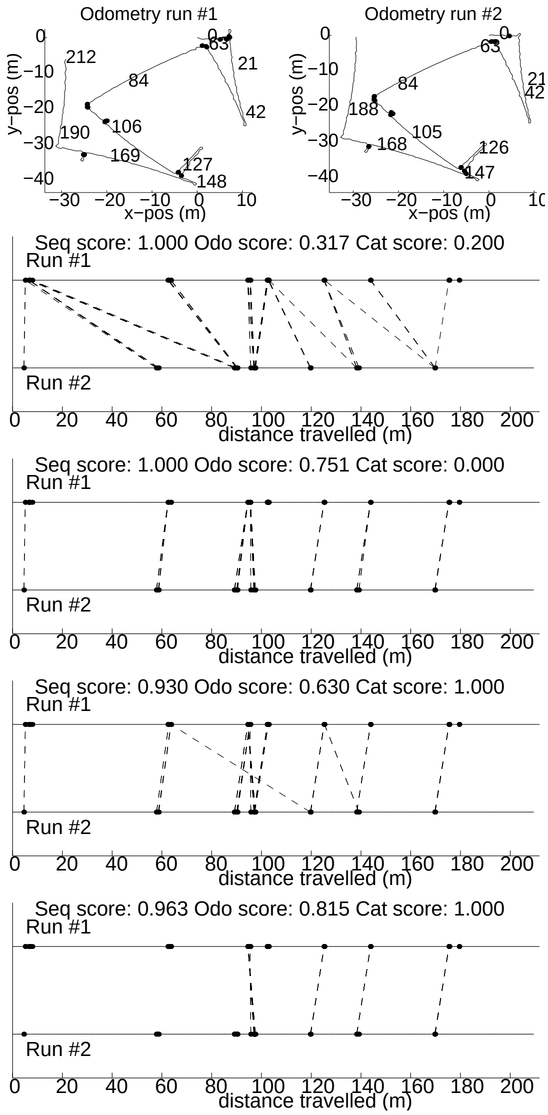

Figure 7: Example alignments for a landmark detection system trained on a run spanning all three hallways ($n = 5$, using left-side sonars at every perception for SOM training). The system was then tested on two separate runs that also spanned all three hallways. Note that there are considerably fewer landmarks in each alignment, as compared with the other figures.
Robot path is represented by solid lines, landmarks by black dots, and alignments by dashed lines. *Top:* The robot's internal odometry showing $x - y$ position. The numbers on the path are the total distance travelled in the to that point. *Underneath:* The $x - y$ odometry is projected onto a distance travelled axis, and aligned landmarks have dashed lines drawn between them. The alignments are (from 2nd from top to bottom) sequential, distance, distance & category, and distance & category with range limit.

5.4 Too Complex?

Given how much computational machinery is at work in this algorithm, it is important to ask how much of it is actually necessary. There are three major components: a predictor network, a Kalman filter operating on the predictor network error to detect landmarks, and a SOM producing categorisations of the resulting landmarks. Our previous work (Marsland et al., 2001) has shown that simple threshold detection of the error curve is insufficient, and that a Kalman filter is necessary. This paper also demonstrated that the single-layer network was sufficient for the sensor prediction. The SOM is used because it is a well-known self-organising algorithm, but many other algorithms could be substituted to produce landmark categorisations.

But could the Kalman filter operate directly on the sonar inputs, removing the need for the prediction network? The problem with this approach is that the R values would then be the variance of the sonar values due to measurement error. The sonar that is directly perpendicular to the wall will have an R value of almost zero, since sonars have very little measurement error for hard surfaces that have small angles of incidence to the beam. By the Kalman update equations the gain will become unity and the variance estimate will become zero. The filter will see every perception as a landmark unless some clever voting scheme is introduced that removes the effects of the perpendicular sonar.

6 Conclusions and Future Work

We have presented a system that is capable of extracting landmarks from realistic, continually updated, robot sonar data. The landmarks are selected based on the principle of unexpected perceptions – places where the robot's predictive sensor model breaks down are unusual and therefore conspicuous and distinctive. Previous work has shown that the system works well in a tightly structured and discretely sampled environment (Marsland et al., 2001), but this is the first time that the method has been applied to more realistic, noisy, continually sampled data. In spite of the added difficulties, some slight modifications to the algorithm produced a robust and efficient landmark detector capable of reliably reproducing most of the same landmarks at each pass through an environment. While it is not perfect – it will typically produce uneven numbers of landmarks at certain locations or occasional landmarks that appear in one run but not another – from a qualitative standpoint, the landmark selection is very satisfactory.

We evaluated the use of the landmarks generated by this system for aligning landmarks between two different trips through the same environment. We produced alignments between landmarks in pairs of runs based on sequence, distance, and both categorisation and distance. These alignments were scored for their quality in three different measures and inspected visually for suitability. In general, alignment using categorisation and distance with a limited range produced the most consistent scores and most pleasing alignments visually. It is clear that combining different types of landmark alignments (e.g., category and distance) produces better results than using just one type of alignment.

Training the SOM using only left-side sonars produced statistically better scores than using all sonars — an effect that might be attributable to reducing the influence of humans walking through the hallway on the robot perceptions. Training in a subset of the testing hallways produced scores which were not detectably worse than training in the entire hallway area. This is useful since a robot may not be able to train in every environment in which it might need to operate, nor is it desirable to have to perform so much time-consuming training.

We believe that the results to date serve as an strong indication of the suitability of this method for the route communication task. We have demonstrated in this paper that the same robot can align landmarks with high accuracy between different runs of the same environment using information about distance travelled and the categories of the landmarks. In previous work (Fleischer and Nehmzow, 2001), we showed that two robots can learn to reliably and consistently link symbols with perceptual categories of the environment. The communication system can provide a link between the internal categories of two robots, thus enabling one robot to interpret a set of landmarks and categories describing a route followed by another robot. Alternatively, the landmark alignment system could be seen as a method by which two robots might be able to learn symbols representing perceptions that they have both encountered on runs in the same environment. The next step in our investigations will be to combine these two components into a system capable of aligning landmarks between two runs in the same environment performed by different robots. We will be investigating both the possibility of using the landmark detection and alignment algorithms to enable the robots to learn consistent symbol-landmark mappings, and the effects of symbol-category consistency on the performance of matching landmarks between two runs by different robots in the same environment.

7 Acknowledgements

Part of this work was supported by a Canadian & U.S. Citizens Scholarship and an ATLAS grant from the Department of Computer Science at the University of Manchester.

References

Billard, A. and Dautenhahn, K. (2000). Experiments in social robotics: Grounding and use of communication in autonomous agents. *Adaptive Behavior*, 7(3/4).

Bourque, E. and Dudek, G. (2000). On-line construction of iconic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00)*, pages 2310–2315.

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.

Fleischer, J. and Nehmzow, U. (2001). Towards robots that give each other navigational directions: Learning symbols for perceptual categories. In *Towards Intelligent Mobile Robotics: Proceedings of the 3rd British Conference on Autonomous Mobile Robotics*, Manchester.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:34 – 45.

Kohonen, T. (1982). Self-organised formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kortenkamp, D. and Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelth National Conference on Artificial Intelligence (AAAI'94)*, pages 979–984, Seattle, Washington.

Linaker, F. and Niklasson, L. (2000). Extraction and inversion of abstract sensory flow representations. In *From Animals to Animats: The 6th International Conference on Simulation of Adaptive Behaviour (SAB'00)*, pages 199 – 208. MIT Press.

Marsland, S. (2001). *On-line Novelty Detection Through Self-Organisation, With Application to Inspection Robotics*. PhD thesis, Department of Computer Science, University of Manchester.

Marsland, S., Nehmzow, U., and Duckett, T. (2001). Learning to select distinctive landmarks for mobile robot navigation. *Robotics and Autonomous Systems*, 37:241 – 260.

Rice, J. (1994). *Mathematical Statistics and Data Analysis*. Duxbury, 2nd edition.

Skubic, M., Matsakis, P., Forrester, B., and Chronis, G. (2001). Generating linguistic spatial descriptions from sonar readings using the histogram of forces. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*.

Steels, L. (1996). Emergent adaptive lexicons. In *Animals to Animats 4*, pages 562 – 567. MIT Press.

Tani, J. and Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for heirarchical learning in sensory-motor systems. *Neural Networks*, 12:1131–1141.

Thrun, S. (1998). Bayesian landmark learning for mobile robot localisation. *Machine Learning*, 33(1):41 – 76.

Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (2000). SOM Toolbox for Matlab 5. Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

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'00)*, pages 2979–2984.

Yamauchi, B. and Beer, R. (1996). Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man and Cybernetics Section B*, 26(3):496 – 505.

Yanco, H. and Stein, L. A. (1993). An adaptive communication protocol for cooperating mobile robots. In *Animals to Animats 2*, pages 478–485.

Zimmer, U. R. (1996). Robust world-modelling and navigation in a real world. *Neurocomputing, Special Issue*, 13(2–4):247 – 260.