"Virtual Coordinates": Perception-based Localisation and Spatial Reasoning in Mobile Robots *

Ulrich Nehmzow[†], Toshihiro Matsui and Hideki Asoh
Electrotechnical Laboratory
1-1-4 Umezono
Tsukuba, Ibaraki 305-8568
Japan
nehmzow,matsui,asoh@etl.go.jp

Abstract

This paper presents a perception-based navigation mechanism that facilitates mobile robot self-localisation and determination of novel routes across uncharted territory.

During a training phase "virtual coordinates" are associated with perceived landmarks, and subsequently used for spatial reasoning. Our experiments show that a good correspondence between virtual coordinates and true Cartesian coordinates can be achieved, and that localisation and path planning are therefore possible by means of the virtual coordinates associated with perceptual landmarks.

1 Introduction

1.1 Motivation

There are two major reasons why the problem of navigation is highly relevant to mobile robotics. Firstly, the benefits of mobility can only be fully exploited if the mobile agent possesses the capability of goal-directed motion. There is ongoing debate within the robotics community regarding the complexity of behaviour that can be achieved using solely reactive actions, but even if goal-directed motion *is* possible through reflex-like responses, we believe that abstract representations of space are desirable because they facilitate a range of *different* navigational manoeuvres (such as localisation and path planning), all exploiting the same global representation. Secondly, mobile robot navigation requires enabling competences such as avoiding dangerous situations and real time processing of sensory perceptions, as well as abstract spatial reasoning. It is therefore a challenging test case for applied artificial intelligence, requiring real time competent interaction with a dynamic environment. Autonomous navigation is therefore a fundamental stepping stone towards intelligently behaving mobile robots.

Because even such seemingly simple tasks such as self-localisation are so hard to achieve in practice, current research has focussed on navigation within previously explored territory. If navigation is to be achieved over short time intervals and short distances only, odometry-based systems can be used. Such systems either rely on odometry alone, or incorporate a perception-based correction element, which allows the occasional calibration of a robot's wheel encoders ([5, 3]). Because of the fundamental problem of odometry — wheel slippage — alternative approaches have used perception-based methods for navigation ([4, 1, 6]). Such methods do not suffer from accumulated sensing error, but fail if several locations in the environment appear identical to the robot's sensors ("perceptual aliasing"). All of these approaches have in common that their spatial reasoning is restricted to explored territory. They do not address the questions of how novel routes (shortcuts) are detectable, or how localisation in unknown parts of the environment can be accomplished.

For the experiments described in this paper, we were interested in developing a navigational mechanism that allows a mobile robot to reason globally about the space it inhabits, and in particular to be able to determine novel routes without prior exploration, simply using current spatial knowledge to learn new facts.

*International Conference on Intelligent Autonomous Systems 5 — IAS 5 — Sapporo 1998.

[†]On leave from Department of Computer Science, Manchester University, Manchester M13 9PL, UK.

Global spatial reasoning would be easy if a reliable global reference frame existed, for example in form of Cartesian coordinates, longitude and latitude, *etc.*. However, as discussed above, to use odometry for this purpose is not a feasible option because of wheel slippage.

Based on this idea, however, we surmised that if such a global, absolute reference frame could be grounded in exteroception (sensory perception) rather than proprioception (odometry), it could be used to determine novel routes and for self-localisation in uncharted territory. In this paper, we therefore introduce the concept of "virtual coordinates", which facilitates global spatial reasoning without using proprioception.

Virtual coordinates are quasi-Cartesian coordinates that are associated with perceptual landmarks, through a learning process. In the first stage of this process, a mobile robot is left to explore its environment, associating its perceptions with quasi-Cartesian coordinates (i.e. the virtual coordinates). In the second stage, both self-localisation and planning of novel routes is then possible, using sensory perception to determine the current virtual coordinate (and thus self-localisation), and trigonometry to determine novel routes.

The main question addressed in this paper, then, is: Is the correspondence between "true" Cartesian coordinates and virtual coordinates good enough to allow global spatial reasoning?

The paper is organised as follows. In section 2 the neural network architecture is described that was used to associate perception with virtual coordinates. Section 3 describes the experimental procedure used in our investigations, and section 4 the results obtained. The final section discusses the implications of this work, its strengths and limitations and open questions for future research.

2 Associating Perception with Global Position

Figure 1 shows the general structure of the radial basis function network ([2]) used for associating perception with virtual coordinates.

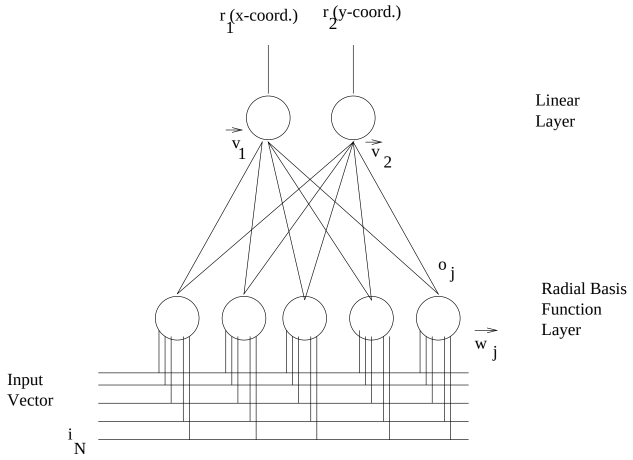

Figure 1: Radial basis function network used to associate perception with virtual coordinates.

In our experiments, the input vector $\vec{i}$ contained 90 elements (the power spectrum of the camera image of a landmark (see subsection 3.2)). The output o_j of each RBF unit j is determined by equation 1:

$$o_j = e^{-\sum \frac{||\vec{i}-\vec{w_j}||}{\sigma}}, \qquad (1)$$

with $\vec{w_j}$ being the weight vector of RBF unit j, $\vec{i}$ being the (90-element) input vector, and σ being the parameter controlling the width of the bell curved capture region of the radial basis function ($\sigma = 0.05$ in our experiments).

For the proof-of-concept experiments discussed here we decided to use the simplest teaching paradigm possible to train the RBF units: the number of RBF units chosen was identical to the number of perceptual landmarks to be learned (fifteen), and the input vector of each landmark formed the weight vector of one RBF unit. This results in one RBF unit being "responsible" for one landmark.

The second layer of the network shown in figure 1 is a Perceptron. Outputs r_1 and r_2 are determined by equation 2:

$$r_1 = \vec{v_1} \cdot \vec{o}, \quad r_2 = \vec{v_2} \cdot \vec{o}, \tag{2}$$

with $\vec{v_1}$ and $\vec{v_2}$ being the weight vectors of output units 1 and 2 respectively, and $\vec{o}$ the output vector of the RBF layer.

Training the output layer is performed as shown in equation 3:

$$\Delta\vec{v_k} = \eta(t_k - r_k) \cdot \vec{o}, \tag{3}$$

with $\Delta\vec{v_k}$ being the required change applied to weight vector $\vec{v_k}$, t_k the desired output for unit k, $\vec{o}$ the output vector of the RBF layer and η being the learning rate ($\eta = 0.1$ in our experiments).

In the experiments, the weights of the RBF layer were set in a one-step process, whilst the weights of the output layer were trained over 1000 learning steps.

3 Experimental procedure

3.1 Data logging procedure

In our robotics experiments, a Nomad 200 mobile robot (see figure 2) was driven manually through the environment shown in figure 2, along the trajectory shown in figure 3.

Figure 2: The Nomad 200 mobile robot (right), and the environment used for the navigation experiments.

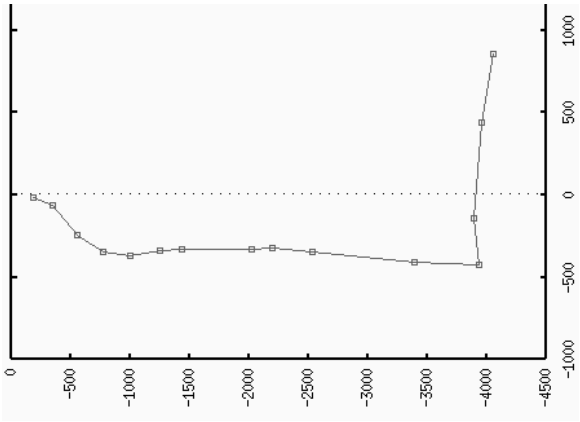

Figure 3: Trajectory taken by the robot in Cartesian space. Dimensions are in tenths of inches.

During the traversal of the this route over 700 images of the robot's omnidirectional CCD camera and their corresponding positions in Cartesian space (as obtained from the robot's odometry system) were logged (one such image is shown in figure 5). Thirty-four of those perceptions were then used off-line for the experiments reported here. The positions where these images were taken are shown in figure 4.

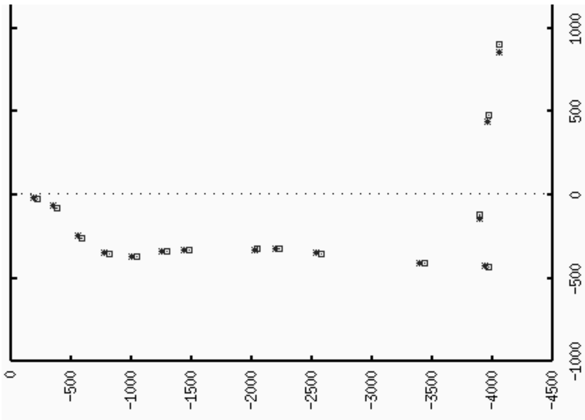

Figure 4: Training and test data obtained along the trajectory shown in figure 3. The fifteen training perceptions (asterisks) were used to train the network, the fifteen test perceptions (boxes)were used to determine the correspondence between virtual and Cartesian coordinates. Dimensions are in tenths of inches.

Fifteen images were used to train the neural network to associate virtual coordinates with the power spectra of these fifteen landmarks (a discussion of the input vector generation follows in subsection 3.2), the remaining fifteen images, whose Cartesian positions were slightly offset from the fifteen training landmarks known to the robot, were used to test the correspondence between true Cartesian position and the position in virtual coordinates, computed by the network. In all experiments, the "true" Cartesian position of a landmark was taken to be identical to the odometry reading obtained at that location. As discussed earlier, odometry does not provide "true" position, but for the proof of concept presented here it is irrelevant whether correct or imaginary positions are associated with perception. For a later real world application of the system, "true" landmark positions would have to be supplied externally.

3.2 Image processing and input vector generation

In total, over 700 images and their Cartesian coordinates were logged automatically. Of these, we used 34, obtained along the trajectory shown in figure 3. Input vectors to the network were generated from these images as follows.

In a first stage of image processing irrelevant data contained in the images was removed (see figure 5).

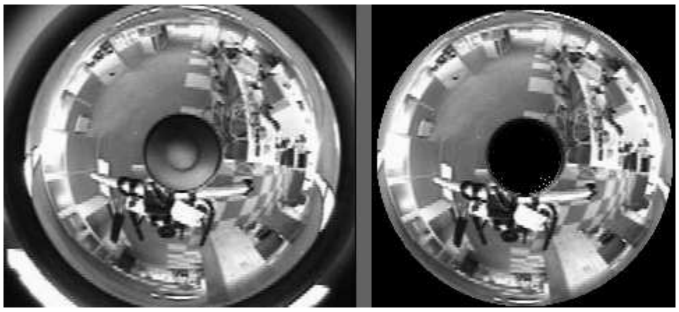

Figure 5: Image preprocessing I. The image on the left shows raw data obtained from the robot's omnidirectional CCD camera, the image on the right shows the processed image, with irrelevant data being replaced by zeros (shown black in the image).

Images such as the ones shown in figure 5 will differ depending on the orientation of the robot's camera, even if images are obtained at the same, identical location. This poses a problem for robot self-localisation, as it is virtually impossible to revisit the very same location twice, having the very same turret orientation. We therefore sought a way to remove the orientation-dependency of the images.

To achieve such an orientation-independent encoding of perceptual landmarks, the power spectrum along 90 radii of the preprocessed image was computed (see figure 6).

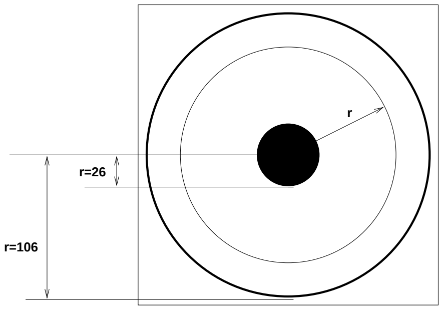

Figure 6: Image preprocessing II. To remove orientation dependency, the power contained in the image is computed along 90 radii between r=26 and r=106. Along each radius grey level values are taken in increments of 1.14^o, 314 values per radius.

The radii r covered the relevant part of the image, i.e. in this case $26 \leq r \leq 106$. Along each radius, greylevel values h_j were taken in increments of 1.14^o, irrespective of radius, i.e. we did not compensate for resolution differences along short and long circumferences. This resulted in 314 pixel values being taken along each radius.

The power H along each radius was then computed by equation 4:

$$H = \sum_j h_j^2. \tag{4}$$

This power, computed here in the space domain, is identical to the power in the spatial frequency domain of the image (Parseval's theorem).

Three power spectra of three different landmarks are shown in figure 7. The 90-element power spectrum, computed for each landmark, was then used as input to the network shown in figure 1, and associated with a virtual coordinate.

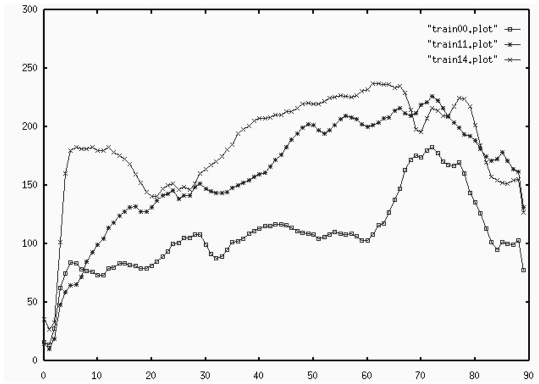

Figure 7: Power spectra for the landmarks at (-23,-181 — 'train00'), (-432,-3932 — 'train11') and (852,-4044 — 'train14') (cf. fig. 4).

4 Results

4.1 Localisation

The first point to establish was to show whether learned virtual coordinates correlate closely to true Cartesian position in space[1].

Figure 8 shows that the network successfully learns to associate *training* landmarks with virtual coordinates that are very close to the Cartesian coordinates of those landmarks. This is the hoped-for result, and merely demonstrates that the network learns well.

[1] For the remainder of the paper, when we speak of "true Cartesian position" we mean the position determined by the robot's wheel encoders, knowing that this is only an estimate of the robot's true position. See discussion in subsection 3.1.

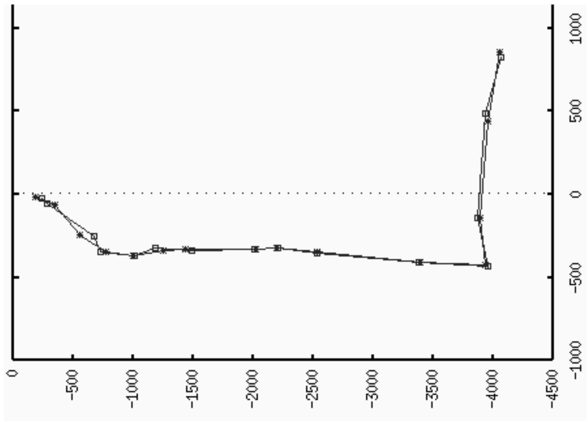

Figure 8: ACTUAL TRAINING TRAJECTORY TAKEN (ASTERISKS) AND TRAJECTORY PERCEIVED IN VIRTUAL COORDINATES (BOXES). DIMENSIONS IN TENTHS OF INCHES.

We then exposed the network to hitherto unknown images, whose Cartesian coordinates were within a few tens of centimetres of known landmarks (test data, see figure 4). The results shown in figure 9 demonstrate that the virtual coordinates associated with the test data correspond very well with the Cartesian coordinates of those test landmarks.

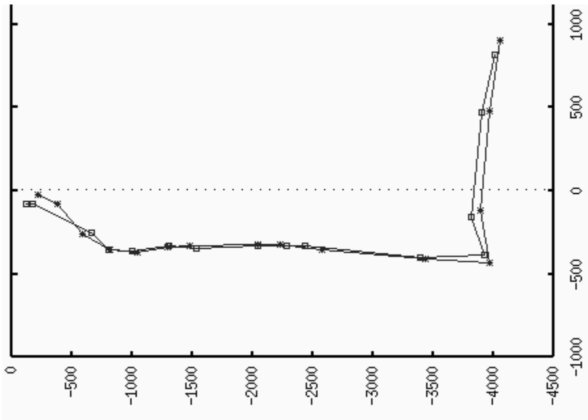

Figure 9: ACTUAL TEST TRAJECTORY TAKEN (ASTERISKS) AND THE TRAJECTORY PERCEIVED IN VIRTUAL COORDINATES (BOXES).

This result *is* interesting, because it is not a foregone conclusion that continuous motion in physical space will result in continuous change in perceptual space. In other words, it is conceivable that small movements of the robot (displacement between training and test location) result in large changes in perceptual space, and that therefore the virtual coordinates computed for a test landmark do not resemble that landmark's Cartesian coordinates well.

But, as figure 9 shows, the correspondence *is* good, which means that virtual coordinates *can* be used for self-localisation and planning of novel paths (see next subsection).

4.2 Detecting novel paths

Next, our objective was to determine how virtual coordinates could be used to determine novel paths between locations known to the robot. For this purpose, we computed the required headings for a number of novel paths in true Cartesian space, and the headings computed in virtual space for the same paths.

Table 1 shows the difference in degrees between the required heading in Cartesian space, and the computed heading using virtual coordinates.

As can be seen from the table, the error is small, meaning that a mobile robot could use its internal, global reference frame — the virtual coordinates — to compute the required heading to a known landmark, even if it has never traveled that route before. Whether that route is actually

	(-362,-816)	(-362,-2572)	(-122,-3882)	(895,-4050)
(-27,-212)	7	2	3	-1
(-362,-816)	—	1	0	-1
(-362,-2572)		—	-3	-4
(-122,-3882)			—	-2

Table 1: Detecting novel paths between various "real world" locations (Cartesian coordinates in tenths of inches). The numbers in the table indicate the error in degrees that the robot makes, computing the required headings using virtual coordinates.

traversable is not of concern here, we are mainly interested to investigate whether a mobile robot can establish a global reference frame based on perception that can be used for global path planning in the real world. The results presented here suggest that the answer to this question is "yes".

5 Discussion

5.1 Localisation far away from known landmarks

One major drawback we see of the mechanism of virtual coordinates is that the network will *always* provide a virtual coordinate for a perceptual stimulus. There is no way of telling whether the robot is on track, or lost. In an extreme example, a robot could be trained in one location, and then made to navigate in a completely different location. At the moment, we see no mechanism (other than looking at the overall excitation level of the RBF layer — see discussion in subsection 5.2) to determine that the robot is in an unknown location. We were therefore interested to find out what would happen if the robot was presented with landmarks that are far away from known landmarks (albeit still within visual range of those known landmarks). We selected four landmarks, whose distances to the nearest landmark known to the robot were 27 cm, 50 cm, 61 cm and 72 cm respectively. The position of these four "far" test positions and their associated virtual coordinates are shown in figure 10.

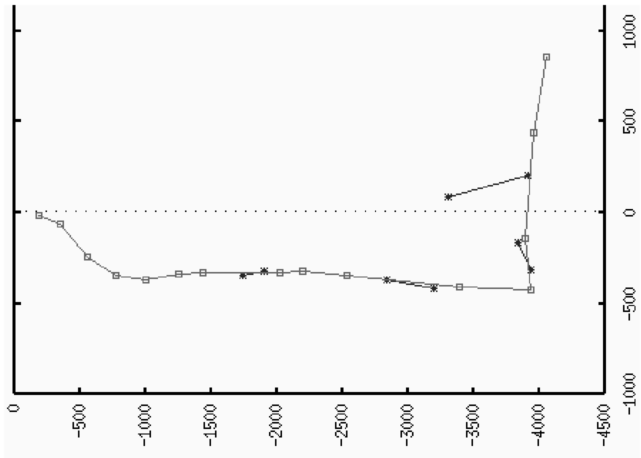

Figure 10: Localisation far from known landmarks. Boxes indicate the locations of known landmarks, and linked asterisks the Cartesian and virtual coordinates of locations far from previously learned landmarks.

Although the discrepancy between Cartesian and virtual coordinates is larger here than in the case where test landmarks were close to training landmarks, the correspondence is nevertheless remarkably good. In all four cases the virtual coordinates are in the "right ballpark"; nevertheless a robot navigating from these "far" landmarks would have to rely on encountering a known landmark eventually, and thus reducing its localisation error.

Table 2 shows the heading error computed for various routes emanating from these "far" locations. All coordinates in this table are given in Cartesian coordinates, so that this table can be related to figures 4 and 10, but obviously for the computation of headings in virtual space the virtual coordinates of the respective locations were used.

	(-32,181)	(-432,-3932)	(852,-4044)
(-422,-3192)	-1	-2	-12
(-324,3929)	5	23	-7
(194,-3906)	-1	50	-34
(-351,-1733)	0	-1	1

Table 2: Heading error in degrees computed from the four "far" landmarks to other locations known to the robot (all locations given in Cartesian coordinates).

Particularly for the landmark located at (194,-3906) the error is large in certain directions. This is clear also from looking at figure 10, and demonstrates that path planning from "far" locations, using virtual coordinates, will not always be possible. This was to be expected; the more interesting observation from table 2 is that for quite a number of routes the error is very small.

5.2 Future work

The system's apparent inability to detect localisation failure has been discussed above. One possible way to address this problem is to look at the overall excitation level of the RBF layer, which will be low if the landmark does not resemble any known landmark.Whether it is possible to determine a threshold of RBF excitation below which no virtual coordinate can be computed reliably is subject to future investigations.

There are a number of procedural parameters that merit further investigation. We have used one RBF unit per learned landmark here. For large environments this will result in large and computationally expensive networks. However, using a smaller number of RBF units than there are landmarks in the environment, and training them similar to the way competitive networks are trained (by shifting their weight vectors *towards* an input vector, rather than making their weight vector *identical* to an input vector) is a possibility.

The proof-of-concept experiments presented here have not investigated how sensitive the computation of virtual coordinates is to lighting conditions and small changes in the environment. This will be investigated in future research.

Virtual coordinates can be used in conjunction with proprioception, in that the virtual coordinates can be used to recalibrate a robot's wheel encoders constantly. By doing this, proprioception *and* exteroception are combined, possibly leading to more reliable robot localisation and path planning. Again, this will be investigated in future research.

In summary, then, we have presented experiments that demonstrate that the association between perception and location ("virtual coordinates") allows robust robot self-localisation and planning of novel routes across uncharted territory. The advantages of virtual coordinates are that they offer the elegance and utility of a global, absolute reference frame, without suffering from the problems of drift that plagues navigation systems that are based on proprioception alone.

References

[1] A. Kurz, "Constructing Maps for Mobile Robot Navigation Based on Ultrasonic Range Data", IEEE Trans. Systems, Man, and Cybernetics - Part B: Cybernetics, Vol. 26, No. 2, April 1996.

[2] D. Lowe and M. Tipping, "Feed-Forward Neural Networks and Topographic Mappings for Exploratory Data Analysis", Neural Computing and Applications 4, pp. 83-95, 1996.

[3] S. Mahadevan, G. Theocharous and N. Khaleeli, "Rapid Concept Learning for Mobile Robots", to appear in J. Machine Learning, 1998.

[4] U. Nehmzow, T. Smithers and J. Hallam, "Location Recognition in a Mobile Robot Using Self-Organising Feature Maps", in G. Schmidt (ed.), "Information Processing in Autonomous Mobile Robots", Springer Verlag, 1991.

[5] B. Yamauchi and P. Langley, "Place Recognition in Dynamic Environments", Journal of Robotics Systems, Special Issue on Mobile Robots, Vol, 14, No. 2, pp. 107-120, 1997.

[6] U. R. Zimmer, "Self–Localization in Dynamic Environments", IEEE/SOFT International Workshop BIES'95, Tokyo, Japan, May 30-31 1995.