Using a Mobile Robot for Cognitive Mapping

Chee K. Wong, Jochen Schmidt, and Wai K. Yeap
Robotics Research Group, Institute for Information Technology Research
Auckland University of Technology
PO Box 12397, Auckland, New Zealand
{chee.wong, jochen.schmidt, wai.yeap}@aut.ac.nz

Abstract

When animals (including humans) first explore a new environment, what they remember is fragmentary knowledge about the places visited. Yet, they have to use such fragmentary knowledge to find their way home. Humans naturally use more powerful heuristics while lower animals have shown to develop a variety of methods that tend to utilize two key pieces of information, namely distance and orientation information. Their methods differ depending on how they sense their environment. Could a mobile robot be used to investigate the nature of such a process, commonly referred to in the psychological literature as cognitive mapping? What might be computed in the initial explorations and how is the resulting "cognitive map" be used to return home? In this paper, we presented a novel approach using a mobile robot to do cognitive mapping. Our robot computes a "cognitive map" and uses distance and orientation information to find its way home. The process developed provides interesting insights into the nature of cognitive mapping and encourages us to use a mobile robot to do cognitive mapping in the future, as opposed to its popular use in robot mapping.

1 Introduction

In robot mapping, one is concerned with the development of efficient algorithms for the robot, with its particular sensors, to simultaneously localize and map its environment. The map, a robot created, is precise. By "precise", it is meant that each surface encountered is remembered and its position is known with a certain degree of accuracy. The robot also knows its position in the environment. In contrast, the humans' (and animals') mapping process, referred to as cognitive mapping, produces fragmentary maps initially which later turn into a representation laden with one's own interpretations and experiences of the world. The map produced in such a process is known as a cognitive map.

The idea that animals compute cognitive maps was first suggested by Tolman [1948], who conducted experiments with rats finding their way in a maze. Since then, much has been said about the complex nature of cognitive maps (see e. g., [Golledge, 1999]) and several computational theories of cognitive mapping have been proposed [Chown *et al.*, 1995; Kuipers, 2000; Yeap and Jefferies, 1999]. More recently, researchers began to use mobile robots to test ideas about cognitive mapping as opposed to robot mapping.

For example, Kuipers and his students have been experimenting with robots to find ways to compute his Spatial Semantic Hierarchy from the ground up (see e. g., [Piers and Kuipers, 1997]). Both the gateway construct in the PLAN model of cognitive mapping [Chown *et al.*, 1995] and the use of exits in the ASR-model of cognitive mapping [Yeap and Jefferies, 1999] were tested on a mobile robot (see e. g., [Kortenkamp, 1993] for the former and [Jefferies *et al.*, 2004] for the latter). Ideas about cognitive mapping based upon neurological findings were also being tested using mobile robots. Examples of such work include [Gaussier *et al.*, 2002; Hafner, 2005]. However, many of these attempts produced algorithms that were more an inspiration from observations about cognitive mapping than a test-bed for theories of cognitive mapping. These researchers were concerned that their robots were able to map its environments successfully and they thus solved much of the robot mapping problem. For instance, they were keen that their robots "close the loops" and produce "real world SLAM results".

In this paper, our goal differs. We note that different animals compute cognitive maps using different sensors and therefore our robot should be treated as a kind of animal with its own peculiar sensing capabilities. For some unknown reasons, humans do not remember a precise map after one or two visits to a new environment. We assume animals do not too and so neither should our robot. To investigate our robot's cognitive mapping process, it is thus best to have our robot compute an imprecise map first and then investigate animal-like strategies for finding its way home using such a map. It is argued that the behavior of such a robot might shed light on cognitive mapping.

To do so, we use a robot equipped with sonar sensors to compute a description of each local space visited. The robot's "cognitive map" is thus a network of such local spaces. Following [Yeap and Jefferies, 1999] theory of cognitive mapping, we refer to each local spaces computed as an Absolute Space Representation (ASR). With sonar sensors, the description of each ASR computed (or more precisely, the shape computed) is not accurate enough to allow its identification on its return journey. Our robot is not programmed with powerful heuristics such as those found in humans. However, lower animals have been observed to use distance and direction information encoded in their cognitive map to find their way. We implemented two such strategies for our robot, one utilizes distance information and the other relative orientation information.

Section 2 describes the way our robot computes its "cognitive map". Section 3 describes the two strategies that our robot uses to find its way home. Section 4 shows the results of our experiments and Sect. 5 concludes with a discussion of the insights obtained from our experiments.

2 Mapping the Environment

When exploring the environment for the first time, the robot creates a "cognitive map" of its environment. This section gives a short description of the process involved; details have been presented by [Schmidt *et al.*, 2006a; 2006b].

2.1 Data Gathering and Pre-Processing

The mapping process used in our system (a wandering robot that records sonar data) is as follows: The robot acquires sonar readings while moving until it runs into an obstacle. At this point an obstacle avoidance algorithm is used, after which the robot can move forward again. Using this input we build a simplified geometric map containing the robot movement path as well as linear surfaces approximated from the sonar data. In the first step, the recorded sonar data is low-pass filtered and converted to surfaces, being a piecewise linear approximation of the sonar distances. These surfaces are simplified further by grouping them, thus removing small gaps.

2.2 Generating ASRs

The approach for generating ASRs from the data gathered by the robot is based on a region split and merge algorithm. The pre-requisite for this algorithm is a geometric map that contains the robot movement path as well as surfaces in terms of line approximations of the original range sensor data. The objective is to divide the perceived information (which is in the form of a fuzzy metric map) into a network of ASRs, thus effectively creating a topological map on top of a metric one. Splitting is done along the robot movement path, using an objective function that computes the quality of a region, based on criteria such as the average room width (corridors are long and narrow compared to rooms) and overall direction (e. g., a corridor is separated from another one by a sharp bend in the wall). Additionally, a regularization term is used in order to avoid the formation of very small regions, which may originate from missing (gaps) or unreliable sensor data. Examples of maps including ASR splittings are shown in the experiment's section (see Fig. 2). The result of this mapping stage will be called the *original map* further on. This is the map the robot will use for returning home.

3 Finding the Way Home

3.1 Re-Mapping

On its way home, the robot basically performs the same data gathering and processing steps as described previously. In contrast to pure mapping, where all data is gathered first, and processed only once at the final position (i. e., the return point), on the return journey the robot performs a map processing and ASR splitting each time it has to stop, which is normally because of an obstacle in its way. This means that at each of these intermediate stops, a new map of the environment as well as a new high-level representation in terms of ASRs is available and can be used in combination with the original map for localization. The result of the localization step is the index of the ASR the robot believes it is currently in, which is a rough estimate of its global position. As it is argued in [Yeap and Jefferies, 1999], this estimate is sufficient for navigation, and an accurate map will not be necessary as long the robot can find the exits to adjacent ASRs.

In the following, we will describe the strategies that we use for localization based on ASR information, and a data fusion algorithm that allows for an overall position estimate computed from the single localization methods.

3.2 Localization Strategies

Two different strategies for localizing the robot based on the original map generated on its way to the current position are presented in the following. Each method computes a local confidence map that contains a confidence value between 0 and 1 for each ASR of the original map. Note that these confidence values are not probabilities, and they do not sum up to one; the interval has been chosen for convenience, and different intervals can be used as desired. The two strategies mentioned previously will be described in the following, together with the method for computing local confidence maps independently for each strategy. The fusion of all local confidence maps, which may have been generated by different robot localization methods with varying reliability, is based on the idea of *Democratic Integration* introduced in [Triesch and von der Malsburg, 2001]. It was developed for the purpose of sensor data fusion in computer vision and computes confidence maps directly on images. The original method has been extended and embedded into a probabilistic framework in [Denzler *et al.*, 2002], still within the area of machine vision. We extend the original approach in a way that we do not use images as an input, but rather generate local confidence maps using various (more or less reliable) techniques for robot localization. A main advantage of this approach is that the extension to more than two strategies is straightforward, as is the replacement of a method by another.

Distance

The first strategy is based on the idea of using the distance the robot traveled from its return point to the current position, just as humans have a rough notion of how far they walked. Note that neither do we care about an exact measurement, nor do we use the actual distance traveled as provided by odometry. Using the odometry data directly would result in very different distances for each journey, as the robot normally moves in a zig-zag fashion rather than straight. Instead we use distance information computed from the ASR splitting of the maps, i. e., ASR length, which is defined by the distance between the entrance and the exit the robot used when passing through a particular ASR. In the maps shown in Fig. 2, start and end points of an ASR are depicted by dark dots (split points) located on a set of connected lines representing the path the robot took. The zig-zag movement of the robot in between two splits is clearly visible, and can be quite different from the line connecting start and end points. The basic strategy is now to compare the distance d traveled when returning home, measured in ASR lengths taken from the intermediate map computed on the return journey, to the ASR lengths taken from the original map computed during the mapping process.

The local confidence map $\boldsymbol{c}_{\text{Dist}} \in \mathbb{R}^N$ (N being the total number of ASRs in the original map) is computed as follows: The confidence for each ASR depends on the overall distance d traveled on the return journey; the closer an ASR is to this distance from the origin, the more likely it is the one the robot is in currently. As the distance traveled is an unreliable estimate, adjacent ASRs should be considered as well, the more the closer they are to the most likely one. We decided to use a Gaussian to model the confidences for each ASR, the horizontal axis being the distance traveled in mm. The Gaussian is centered at the current overall distance traveled d. Its standard deviation σ is dependent on the distance traveled, and was chosen as $\sigma = 0.05d$. Note that although a Gaussian is used here, we do not try to model a probability density. A Gaussian was rather chosen for a number of reasons making it most suitable for our purpose: It allows for a smooth transition between ASRs, and the width can be easily adjusted by altering the standard deviation. This is necessary as the overall distance traveled gets more and more unreliable (due to slippage and drift) the farther the robot travels. The confidence value for an ASR is determined by sampling the Gaussian at the position given by the accumulated distances from the origin (i. e., where the robot started the homeward journey) to the end of this ASR. After a value for each ASR is computed, the local confidence map $\boldsymbol{c}_{\text{Dist}}$ is normalized to the interval $[0; 1]$.

Relative Orientation

The second method for computing estimates of the robot's position with respect to the original map is based on using relative orientation information generated while dividing the map into ASRs. During its journey, the robot enters an ASR at one location and exits at a different one, usually including zig-zag movements in between. We define the direction of an ASR as the direction of the line connecting the entrance and exit points. Certainly this direction information varies every time the robot travels through the environment, but the overall shape between adjacent ASRs is relatively stable. Therefore, we propose to use angles between ASR directions as a simple measure of the current position of the robot. Note that this information is pretty much useless on its own, because the same angles (i. e., direction changes) can be found in different locations of the environment. However, combining this strategy with others can help to decide between position estimates that would otherwise be indistinguishable. It has the advantage that angles between adjacent ASR directions are a local measure of direction changes, thus keeping the influence of odometry errors due to drift and slippage to a minimum.

The basic idea of the strategy is as follows: Firstly, all angles $\alpha_1, \ldots, \alpha_{N-1}$ between adjacent ASRs in the original

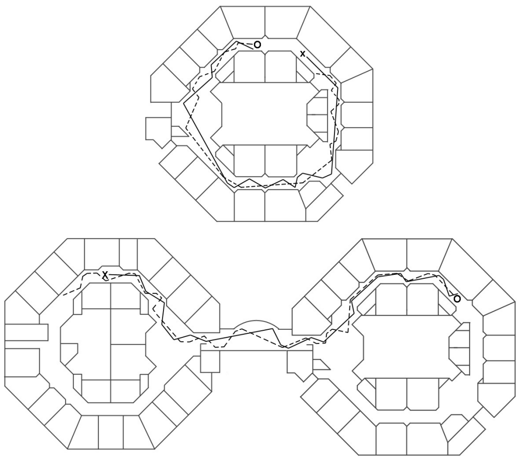
Figure 1: Maps showing the actual layout of the building used for Experiment 1 (top) and Experiment 2 (bottom). The path the robot took while generating the original map is shown as a solid line, with 'X' marking the starting location of the experiment. The dashed line visualizes the path the robot took on the homeward journey and 'O' marks its starting point.

map are computed. This can be done offline, as this map is fixed during the homeward journey. In the re-mapping process while going home, new ASRs are computed in the new map based on data gathered while the robot travels. Using the direction information contained in this map, the angle β between the current ASR and the previous one can be computed. Comparing this angle to all angles of the original map gives a clue (or many) for the current location of the robot. The comparison of angles and conversion to confidence values is done by computing the cosine of the difference angle, and mapping the resulting values to the interval $[0; 1]$, which results in a local confidence map $\boldsymbol{c}_{\text{Dir}}$:

$$\boldsymbol{c}_{\text{Dir}i} = \frac{1}{2}(\cos|\alpha_i - \beta| + 1), \quad i = 1, \ldots, N-1 \quad . \quad (1)$$

This results in high values for similar angles and low values for dissimilar ones. The confidence map computed this way can already be used for further processing. Since the overall reliability of the relative orientation strategy as described above is rather low compared to the confidence values from other methods (in this case using distance information), we currently reduce the confidence values by half.

3.3 Fusion of Strategies

We will now describe how to merge the separate local confidence map into a single global one based on the original Democratic Integration approach published by [Triesch and von der Malsburg, 2001]. The basic idea is straightforward, as the fusion is done by computing a weighted sum of all local confidence maps. The main advantage of using democratic integration becomes visible only after that stage, when the weights get adjusted dynamically over time, dependent on the reliabilities of the local map. Given M local confidence maps $\boldsymbol{c}_{\text{loc}i}(t)$ at time t (i. e., as in our case two, namely $\boldsymbol{c}_{\text{Dist}}$ and $\boldsymbol{c}_{\text{Dir}}$) generated using different strategies, the global map

$\boldsymbol{c}_{\text{glob}}(t)$ is computed as:

$$\boldsymbol{c}_{\text{glob}}(t) = \sum_{i=0}^{M-1} w_i(t)\boldsymbol{c}_{\text{loc}\,i}(t) \quad , \tag{2}$$

where $w_i(t)$ are weighting factors that add up to one.

An estimate of the current position of the robot with respect to the original map can now be computed by determining the largest confidence value in $\boldsymbol{c}_{\text{glob}}(t)$. Its position b in $\boldsymbol{c}_{\text{glob}}(t)$ is the index of the ASR that the robot believes it is in. The confidence value c_{glob_b} at that index gives an impression about how reliable the position estimate is in absolute terms, while comparing it to the second best one (and maybe even third best one) shows the reliability relative to other ASRs.

In order to update the weighting factors, the local confidence maps have to be normalized first. The normalized map $\boldsymbol{c}'_{\text{loc}\,i}(t)$ is given by:

$$\boldsymbol{c}'_{\text{loc}\,i}(t) = \frac{1}{N}\boldsymbol{c}_{\text{loc}\,i}(t) \quad . \tag{3}$$

The idea when updating the weights is that local confidence maps that provide very reliable data get higher weights than those which are unreliable. Different ways for determining the quality of each local confidence map are presented in [Triesch and von der Malsburg, 2001]. We use the normalized local confidence values at index b, which has been determined from the global confidence map as shown above, i. e., the quality $q_i(t)$ of each local map $\boldsymbol{c}_{\text{loc}\,i}(t)$ is given by $c'_{\text{loc}\,b}(t)$. Normalized qualities $q'_i(t)$ are computed by:

$$q'_i(t) = \frac{q_i(t)}{\sum_{j=0}^{M-1} q_j(t)} \quad . \tag{4}$$

The new weighting factors $w_i(t+1)$ can now be computed from the old ones:

$$w_i(t+1) = w_i(t) + \frac{1}{t+1}\left(q'_i(t) - w_i(t)\right) \quad . \tag{5}$$

This is a recursive formulation of the average over all qualities from time zero to t. Using this update equation and the normalization of the qualities in (4) ensures that the sum of the weights equals one at all times.

4 Experimental Results

To evaluate the performance of the algorithm proposed, we used a Pioneer 2 robot from Activmedia, equipped with sonar sensors and an odometer. As described earlier, the robot must first explore and generate a representation of the environment, i. e., a "cognitive map". The robot was then instructed to return home (home being the start point of the mapping process). The aim was to determine whether the robot can find its way back home using this inexact "cognitive map".

We conducted various experiments in an office environment, two of which are presented in this paper. Further experimental results can be found in [Schmidt *et al.*, 2006b]. Figure 1 presents the layouts of the environment used. The paths that the robot took during mapping (solid line) and going home (dashed line) are both shown in Fig. 1 as well. For the mapping stage, the robot started at the location marked by

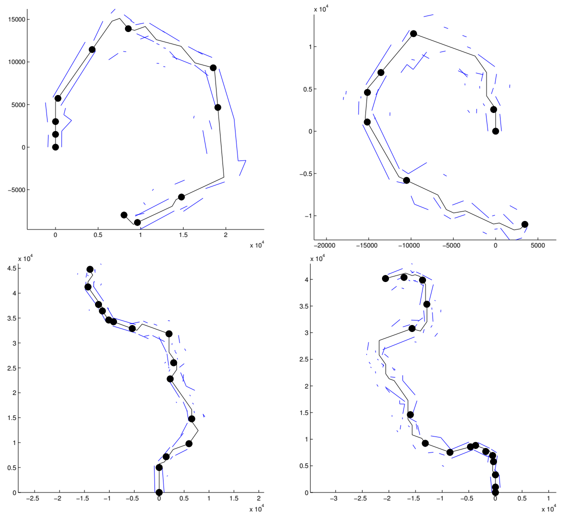

Figure 2: Top: Experiment 1, bottom: Experiment 2. Left: original maps, right: maps generated during the homeward journey. The black dots indicate the points where the map is split into separate ASRs. The robot movement always starts at the origin. The origin of the homeward journey is (approximately) the same position as the end point coordinate of the respective original map; in particular this means, that the map of the homeward journey for Experiment 2 (bottom right) is upside-down compared to the original map (bottom right).

'X', and was stopped at a random position (marked by 'O'), where it started returning home. The return journey stopped when the robot believed that it reached home, or, more precisely, the ASR that contains the start location 'X'. It is important to note that we did not intervene with the environment at any point in time. That is, things that existed in the environment during the mapping process (such as rubbish bins, flower pots, cabinets, etc) may or may not be there during the return home journey; and doors may be open or closed depending on the time of the experiment.

Figure 2 shows the representations generated from the two experiments. The top row depicts the maps generated from the mapping and going home processes respectively for Experiment 1; the bottom row are the maps generated from the mapping and going home processes respectively for Experiment 2. Comparing the maps generated during mapping and going home highlights the difficulty in using these maps directly for localization. Each time, the robot goes through the same environment, it will generate different representations due to sensory inaccuracies. Figures 3 and 4 show the confidence maps computed at four locations during the return home journey for Experiments 1 and 2 respectively. The light dotted lines shows the ASR estimate using the ASR length information (distance method) and the dark dashed lines depicts the ASR estimate using the angles between ASRs (relative orientation method). The solid line is the overall ASR estimate for the corresponding ASR (horizontal axis). Note that the confidences have values between 0 and 1 (vertical axis) but do not sum up to 1. In Fig. 3, the top left map shows a narrow peak for the overall confidence at ASR 1, with cor-

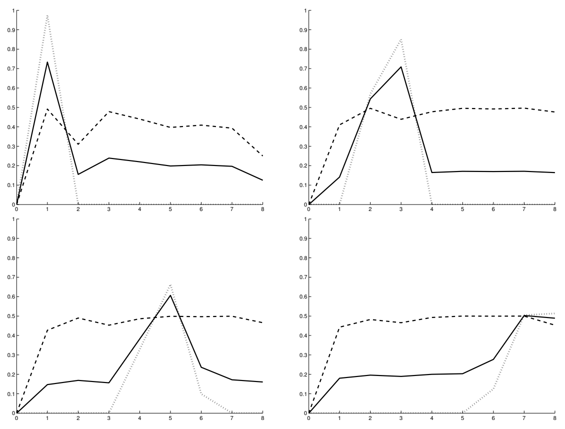
Figure 3: Confidence maps at four locations during the return home journey for Experiment 1: distance (light dotted), relative orientation (dark dashed), and overall map (solid). Horizontal axis: ASR number; vertical axis: confidence (0 to 1).

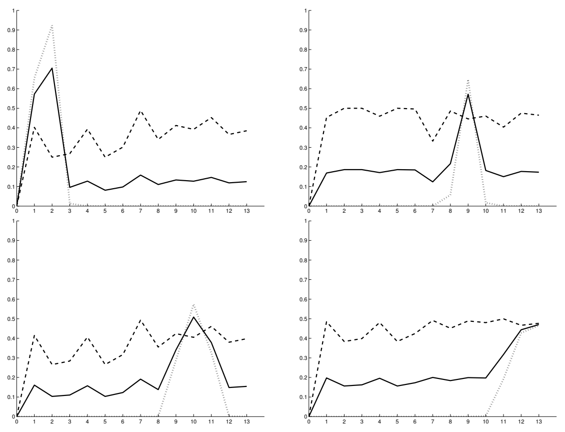
Figure 4: Confidence maps at four locations during the return home journey for Experiment 2: distance (light dotted), relative orientation (dark dashed), and overall map (solid).

responding peaks for both distance and relative orientation. The narrow peak signifies the robot being very confident of being in ASR 1. The top right map however is quite different. It has two peaks, one slightly higher than the other. It shows that the robot is around the transition region between ASR 2 and 3. The higher confident value for ASR 3 shows that the robot feels that it has already moved into ASR 3. The bottom left map shows a similar account except that this time, neighboring ASRs (4 and 6) have higher values, resulting in a wider peak. Finally, the bottom right map shows that ASRs 7 and 8 pretty much have the same confidences, meaning the robot is unsure which of the two ASRs it is in. Figure 4 shows another set of confidence maps corresponding to four locations in Experiment 2. Figure 5 shows a sequence of four consecutive confidence maps from experiment 2. These confidences are taken from the period when the robot is traveling from ASR 6 to ASR 7. Starting from top left figure, the robot is confident it is in ASR 6. As it moves to the boundary between ASRs 6

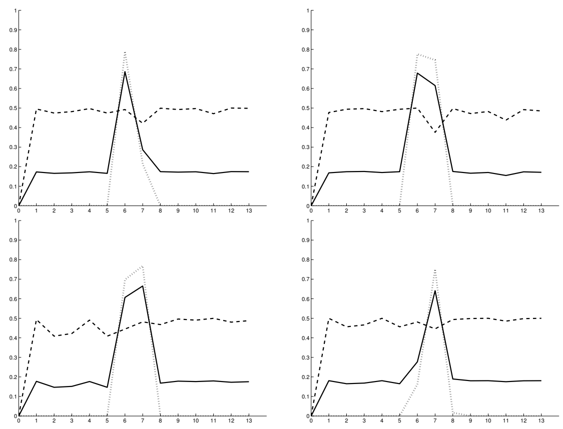
Figure 5: Four consecutive confidence maps computed during the return home journey for Experiment 2 at the transition from ASR 6 to 7: distance (light dotted), relative orientation (dark dashed), and overall map (solid).

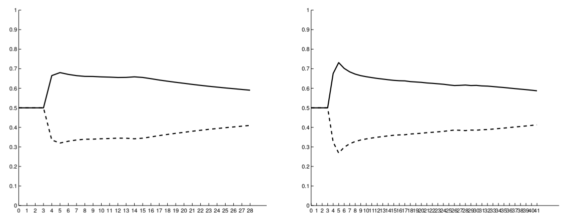
Figure 6: Adjustment of weighting factors over time: ASR length weights (solid) and ASR angle weights (dashed).

and 7, it is unsure of where it is exactly, as depicted by the top right and bottom left images. And when it has moved away from the transition area, the confidence of ASR 6 decreases dramatically and the peak is now centered on ASR 7.

Figure 6 shows the distribution of weighting used on the localization strategies for calculating the overall confidence values. The solid line is the plot of distance method weights and the dashed line represents the weights of the relative orientation method. For the first three ASRs, the weights for both distance and orientation are initialized to be 0.5 because we only start computations after the third ASR. The ASR distance weight then increases, because the confidence on ASR distance is higher than the confidence computed from the orientation method. As time progresses, the weights of the distance method decrease due to a decrease in confidence, and vice versa for confidence from the orientation method.

The results from the confidence maps show that the method proposed provides a consistent approach for using an inexact "cognitive map" to allow a mobile robot to find its way back home. It does not provide the exact pose of the robot in the environment, but rather an approximation, which we believe is sufficient for navigation and new exploration.

5 Conclusion

We have implemented a very basic algorithm for the robot to find its way home, namely exploit ASR-distance traveled to re-trace its movements to return home and comparing the relative orientation between adjacent ASRs.

Anecdotal evidence suggests that one is aware of significant turns in a journey and we thus program our robot to extract such information from its cognitive map. It turns out that in the journeys experienced, such information does not provide much help for the robot to locate itself. In the future, we would like to investigate how the robot could orient itself from the home position and how such more general orientation could be used to find one's way home. Much has been discussed with respect to the use of distance information in cognitive mapping. For example, numerous experiments with chickens and pigeons have shown that they are able to use both absolute and relative distance in their search for food (e.g., [Cheng *et al.*, 2006]). Experiments with bees and ants have shown that they can perform internal calculations of the distance and direction traveled to perform path integration (e.g., [Cornell and Heth, 2004] for a general discussion). Most of these experiments were concerned with the actual distance traveled and how the individual species deal with the errors in their measurements, as do most work on robot mapping to date. Using our robot, we have shown another way of using distance information, namely ASR-distance traveled as opposed to actual distance traveled.

ASR-distance is obtained from the shape of the ASR computed. In the past, there has been scant evidence that humans/animals do pay attention to the shape of each local environment (or, in our terminology, ASR) very early on in their initial exploration of a new environment. However, the debate has now intensified and this is especially true in the animal literature where the problem is commonly referred to as geometry in animal spatial behavior (see [Cheng and Newcombe, 2005]). In many of these experiments, a relocation task utilizing a box-shaped environment is used, and the principal axes of the environment appear to be most useful. Our work here emphasized yet another possibility, namely using a straight line distance between exits of interests in an ASR. A remark is worth making regarding the surprisingly good results obtained in our experiment. Although our robot was allowed to wander on its own during all the trials, it managed not to enter any of the rooms. Consequently, the robot appears to be constantly moving forward along the corridor and this might have accounted for much of the success of the experiment. This was not planned. It would be interesting to see how the resulting ASRs would be if the robot enters, say, the middle room and explores the space in it. Nonetheless the current work has shown how we might use a mobile robot to investigate more cognitively oriented strategies in cognitive mapping.

References

[Cheng and Newcombe, 2005] K. Cheng and N. S. Newcombe. Is there a geometric module for spatial orientation? Squaring theory and evidence. *Psychonomic Bull Rev*, 12:1–23, 2005.

[Cheng *et al.*, 2006] K. Cheng, D. M. Kelly M. L. Spetch, and V. P. Bingman. Small-scale Spatial Cognition in Pigeons. *Behavioural Processes*, 72:115–127, 2006.

[Chown *et al.*, 1995] E. Chown, S. Kaplan, and D. Kortenkamp. Prototypes, location, and associative networks (PLAN): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995.

[Cornell and Heth, 2004] E. H. Cornell and C. D. Heth. Memories of travel: Dead reckoning within the cognitive map. In G. Allen, editor, *Human spatial memory: Remembering where*, pages 191–215. Lawrence Erlbaum Associates, Mahwah, NJ, 2004.

[Denzler *et al.*, 2002] J. Denzler, M. Zobel, and J. Triesch. Probabilistic Integration of Cues From Multiple Cameras. In R. Würtz, editor, *Dynamic Perception*, pages 309–314. Aka, Berlin, 2002.

[Gaussier *et al.*, 2002] P. Gaussier, A. Revel, J. P. Banquet, and V. Babeau. From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biol. Cybern.*, 86:15–28, 2002.

[Golledge, 1999] R. G. Golledge. Human wayfinding and cognitive maps. In R. G. Golledge, editor, *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. John Hopkins University Press, Baltimore, 1999.

[Hafner, 2005] V. V. Hafner. Cognitive maps in rats and robots. *Adaptive Behavior*, 13(2):87–96, 2005.

[Jefferies *et al.*, 2004] M. Jefferies, W. Weng, J. T. Baker, and M. Mayo. Using context to solve the correspondence problem in simultaneous localisation and mapping. In *Proc. Pacific Rim Int. Conf. on Artificial Intelligence*, volume 3157 of *Lecture Notes in AI*, pages 664–672, 2004.

[Kortenkamp, 1993] D. Kortenkamp. *Cognitive Maps for Mobile Robots: A Representation for Mapping and Navigation*. PhD thesis, University of Michigan, 1993.

[Kuipers, 2000] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[Piers and Kuipers, 1997] D. M. Piers and B. J. Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–227, 1997.

[Schmidt *et al.*, 2006a] J. Schmidt, C. K. Wong, and W. K. Yeap. A Split & Merge Approach to Metric-Topological Map-Building. In *Int. Conf. on Pattern Recognition (ICPR)*, volume 3, pages 1069–1072, Hong Kong, 2006.

[Schmidt *et al.*, 2006b] J. Schmidt, C. K. Wong, and W. K. Yeap. Mapping and Localization with Sparse Range Data. In *Int. Conf. on Autonomous Robots and Agents (ICARA)*, Palmerston North, New Zealand, 2006.

[Tolman, 1948] E. C. Tolman. Cognitive Maps in Rats and Men. *Psychological Review*, 55(4):189–208, 1948.

[Triesch and von der Malsburg, 2001] J. Triesch and Ch. von der Malsburg. Democratic Integration: Self-Organized Integration of Adaptive Cues. *Neural Computation*, 13(9):2049–2074, 2001.

[Yeap and Jefferies, 1999] W. K. Yeap and M. E. Jefferies. Computing a Representation of the Local Environment. *Artificial Intelligence*, 107(2):265–301, 1999.