Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

RatSLAM on the Edge: Revealing a Coherent Representation from an Overloaded Rat Brain

Michael Milford, Gordon Wyeth and David Prasser

School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, Australia

{milford, wyeth, prasserd}@itee.uq.edu.au

Abstract – **The RatSLAM system can perform vision based SLAM using a computational model of the rodent hippocampus. When the number of pose cells used to represent space in RatSLAM is reduced, artifacts are introduced that hinder its use for goal directed navigation. This paper describes a new component for the RatSLAM system called an experience map, which provides a coherent representation for goal directed navigation. Results are presented for two sets of real world experiments, including comparison with the original goal memory system's performance in the same environment. Preliminary results are also presented demonstrating the ability of the experience map to adapt to simple short term changes in the environment.**

Index Terms – SLAM, goal, navigation, mapping

I. Introduction

The RatSLAM system [1] was developed to determine whether it is possible to create a biologically inspired SLAM system that can perform as well as, if not better than, conventional techniques for SLAM (for example [2], [3]). Previous work has demonstrated that RatSLAM is capable of performing real-time, on-line SLAM in indoor [4] and outdoor [5] environments on a scale similar to other well-known SLAM systems. RatSLAM has shown some advantages, such as the ability to use low-cost vision sensing in the place of typical laser-based measurements to perform goal directed navigation based on the learnt representations [6].

One of the interesting properties of the RatSLAM system is that its core representation, the *pose cells*, do not necessarily map directly on to the Cartesian space that they encode. A single cell might represent more than one place (a *collision*), or a single place might be represented by more than one cell (a *discontinuity*). If the pose cells are arranged sparsely over the space to be represented, these properties are less apparent. However, when the number of pose cells is decreased, the number of collisions and discontinuities increases. The representations that are formed by the pose cells are still stable and consistent, but the collisions and discontinuities create problems for the goal memory system, and goal recall becomes unreliable.

This paper examines the effect of reducing the number of pose cells, and introduces an extension to RatSLAM known as an *experience map*. The experience map produces a representation of the Cartesian space without collisions and discontinuities based on the temporal patterns of the pose cells combined with external sensor information. Using this new representation space it is possible to perform effective goal recall in the presence of a large number of collisions and discontinuities in the pose cells.

This paper proceeds as follows. Section II briefly introduces the RatSLAM system and examines the representation it builds of a large indoor environment. The goal memory system is also briefly described, followed by analysis of the temporal map it produces for the same environment. This leads into Section III, which introduces the experience mapping algorithm as a solution to the limitations of goal memory, and explains how it can adapt to simple environment changes. Section IV describes how experience maps can be used for goal recall. Section V presents SLAM and goal recall experiments run in the same indoor environment as in Section II but using the experience mapping algorithm. Results for the experiments are presented and analyzed. Section VI presents preliminary results demonstrating the ability of the experience map to adapt to the placement and removal of a simple obstacle blocking a path in the environment. Section VII discusses the experience mapping algorithm and related research before the paper concludes in Section VIII.

II. RatSLAM and Goal Memory

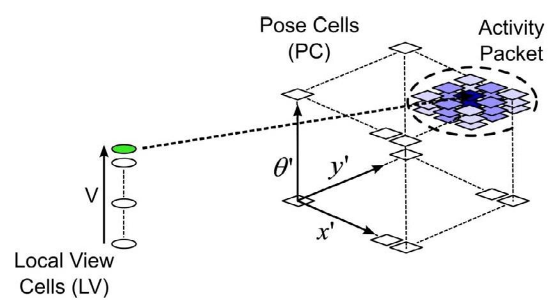

Fig. 1 The core RatSLAM pose cell and local view cell networks.

This section briefly describes the RatSLAM and goal memory systems – a more detailed description is given in [1] and [6]. Fig. 1 shows the core structure of the RatSLAM system. The robot's pose is represented by activity in a competitive attractor neural network called the pose cells. Wheel encoder information is used to perform path integration by appropriately shifting the current pose cell activity. Activity can wrap in all three directions in the pose cell matrix. The path integration model results in each pose cell initially representing a 0.25m x 0.25m x 10° space. Vision information is converted into a local view (LV) representation that is associated with the currently active pose cells. If familiar, the current visual scene also causes activity to be injected into the particular pose cells associated with the currently active local view cells.

For small environments, the RatSLAM representations have a high degree of correspondence to the Cartesian layout of the environment. However, as the environments become larger and more complex, a number of phenomena become common. Vision information starts to cause more frequent loop closures. This leads to discontinuities in the pose cell matrix where the dominant packet of activity jumps from one location to another. Because re-localization is not an instantaneous process, the system also learns multiple representations of the same physical areas in the environment. The complementary attribute of collisions in the pose cell matrix also becomes increasingly common – clusters of pose cells become associated with more than one location in the environment.

Fig. 3 shows the path of the dominant packet of activity in the pose cell matrix during an hour long experiment in the environment shown in Fig. 2. Loop closures are shown by straight dashed lines. There are many collisions and there is little spatial correspondence to the actual environment. Despite these discontinuities and collisions, the maps produced by the RatSLAM system become consistent and stable over time. They are valid representations of the space, but are difficult to use directly for navigation.

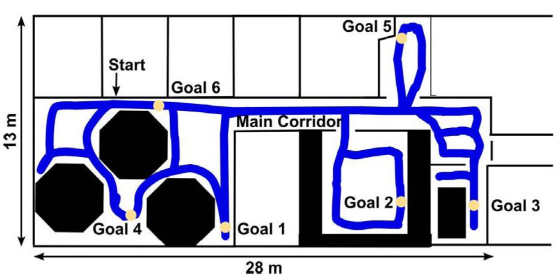

Fig. 2 Floorplan of the indoor environment showing the robot's path. The goal locations are for the experiments in Section V.

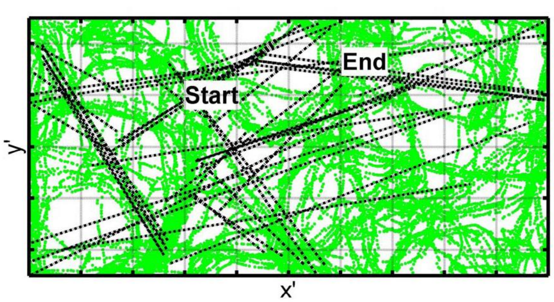

Fig. 3 Path of the dominant packet of activity through the pose cell matrix for a 40 x 20 x 36 pose cell matrix, projected onto the $(x'y')$ plane. Each grid square represents 4 x 4 pose cells in the $x'y'$ plane. 'Start' and 'End' mark the initial and final location of the dominant activity packet.

A. Goal Memory

Fig. 4 shows the temporal map created for this environment by the goal memory system described in [6], for a goal located at the 'Start' location in Fig. 3. Because the goal memory system produces a temporal map by using a copy of the pose cell matrix, it inherits the discontinuities and multiple representations of the pose cell matrix. Consequently the goal memory system is not capable of producing a meaningful temporal map under these conditions. The following section describes a new algorithm that uses the RatSLAM representations to build a map that solves the problems highlighted in this section.

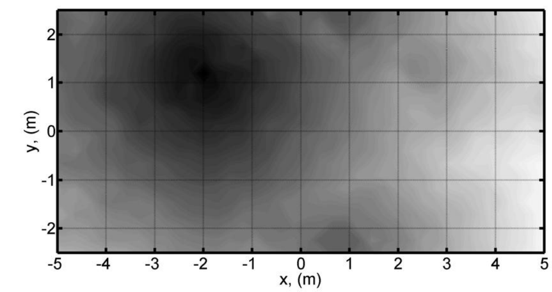

Fig. 4 Temporal map produced by the goal memory system.

III. Experience Mapping

The premise of the experience mapping algorithm is the creation and maintenance of a collection of experiences and inter-experience links. The algorithm creates experiences to represent certain states of activity in the pose cell and local view networks. The algorithm also learns behavioral, temporal, and spatial information in the form of inter-experience links. Fig. 5 shows the relationship between the experience map and the core RatSLAM representations. A more detailed discussion of the algorithm is given in [7].

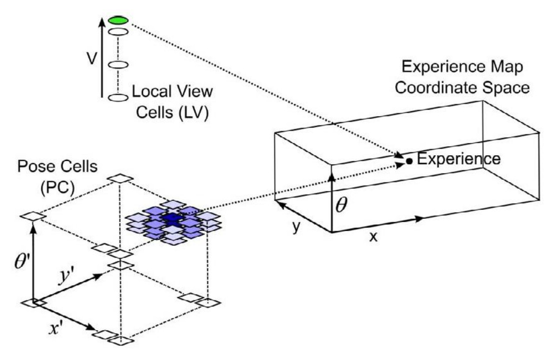

Fig. 5 An experience is associated with certain pose and local view cells, but exists within the experience map's own coordinate space.

A. Experiences

Experiences have an activity level that is dependent on how close the activity peaks in the pose cells and local view cells are to the cells associated with the experience. The component of activity determined by the pose cell network activity is given by:

$$r_r' = \frac{\sqrt{\left(x'_{pc} - x'_i\right)^2 + \left(y'_{pc} - y'_i\right)^2}}{r_a} \quad (1)$$

$$\theta_r' = \frac{\left|\theta'_{pc} - \theta'_i\right|}{\theta_a} \quad (2)$$

$$E_{x'y'\theta'} = \begin{cases} 0 & \text{if } r_r' > 1; \\ 0 & \text{if } \theta_r' > 1; \\ 2 - r_r' - \theta_r' & \text{otherwise} \end{cases} \quad (3)$$

where x'_{pc}, y'_{pc}, and θ'_{pc} are the coordinates in the pose cell matrix of the dominant activity packet, x'_i, y'_i, and θ'_i are the coordinates of the pose cells associated with experience *i*, r_a is the zone constant for the (x', y') plane, and θ_a is the zone constant for the θ' dimension. The visual scene V_i switches an experience on or off:

$$E_i = \begin{cases} 0 & \text{if } V_{curr} \neq V_i; \\ E_{x'y'\theta'} & \text{if } V_{curr} = V_i \end{cases} \quad (4)$$

where V_{curr} is the current visual scene, and V_i is the visual scene associated with experience *i*. The most active experience is known as the peak experience. Learning of new experiences is triggered by the peak experience's activity level dropping below a threshold value.

B. Experience Transitions

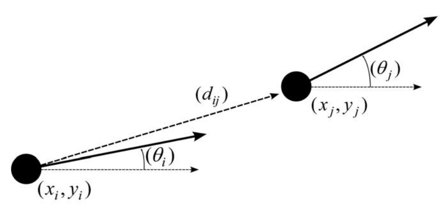

Fig. 6 Links between experiences store several types of information, including odometric information about the robot's movement during the transition.

Inter-experience links store temporal, behavioral, and odometric information about the robot's movement between experiences. Fig. 6 shows a transition from experience *i* to experience *j*. The physical movement of the robot during this transition is given by:

$$d\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i = \begin{pmatrix} \theta_j \\ x_j \\ y_j \end{pmatrix} - \begin{pmatrix} \theta_i \\ x_i \\ y_i \end{pmatrix} = \begin{pmatrix} d\theta_{ij} \\ dx_{ij} \\ dy_{ij} \end{pmatrix} \quad (5)$$

where $d\mathbf{p}_{ij}$ is a vector describing the position and orientation of experience *j* relative to experience *i*. Repeated transitions between experiences result in an averaging of the odometric information:

$$d\mathbf{p}_{ij}^{new} = A.d\mathbf{p}_{ij}^{old} + B.d\mathbf{p}_{ij}^{curr} \quad (6)$$

where $A = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & d_s.\cos d\theta & -d_s.\sin d\theta \\ 0 & d_s.\sin d\theta & d_s.\cos d\theta \end{pmatrix}$, $B = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$,

$$d\theta = \frac{1}{2}\left[\tan^{-1}\left(\frac{dy_{ij}^{curr}}{dx_{ij}^{curr}}\right) - \tan^{-1}\left(\frac{dy_{ij}^{old}}{dx_{ij}^{old}}\right)\right], \text{ and}$$

$$d_s = \left(d_{ij}^{curr} + d_{ij}^{old}\right) / \left(2d_{ij}^{old}\right).$$

C. Map Correction

Discrepancies between a transition's odometric information and the linked experiences' (x,y,θ) coordinates are minimized through a process of map correction:

$$\Delta\mathbf{p}_i = \alpha\left[\sum_{j=1}^{N_f}\left(\mathbf{p}_j - \mathbf{p}_i - d\mathbf{p}_{ij}\right) + \sum_{k=1}^{N_t}\left(\mathbf{p}_k - \mathbf{p}_i - d\mathbf{p}_{ki}\right)\right] \quad (7)$$

where α is a learning rate constant, N_f is the number of links from experience *i* to other experiences, and N_t is the number of links from other experiences to experience *i*. The experience map is subject to the same constraints of any network style learning system – appropriate learning rates must be used to balance rapid convergence with instability. Experimentation has determined that a learning rate of 0.5 resulted in the map rapidly converging to a stable state.

When the orientation of an experience is changed through the map correction process, the (dx,dy) component of the transitional information must also be updated to account for the rotation:

$$d\mathbf{p}_{ij}^{new} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta\theta_i & -\sin\Delta\theta_i \\ 0 & \sin\Delta\theta_i & \cos\Delta\theta_i \end{pmatrix} d\mathbf{p}_{ij}^{curr} \quad (8)$$

D. Experience Map Adaptation

The experience maps represent changes in the environment through modification of the inter-experience transition information. As well as learning experience transitions, the system also monitors transition 'failures'. Transition failures occur when the robot's current experience switches to an experience other than the one expected given the robot's current movement behavior. If enough of these failures occur for a particular transition, indicated by the *confidence* level dropping below a certain threshold, then that link is deleted from the experience map. The confidence level is given by:

$$c_{ij} = \frac{n_{ij}}{n_{i\beta_{ij}}} \quad (9)$$

where n_{ij} is the number of times the transition between experience *i* and *j* has occurred, and $n_{i\beta_{ij}}$ is the number of times a transition from experience *i* to any experience using the behavior β_{ij} has occurred. For longer term experiments a more advanced scheme will be required such as the recency weighted averaging described in [8].

IV. Goal Recall Using Experience Maps

The temporal link information and local spatial properties of the experience map are used to form a temporal map that can be used for goal recall.

A. Temporal Map Creation

To create the temporal map, the peak experience is seeded with a zero time stamp value, and all other experiences are seeded with a 'very large' value. Time stamp values are then assigned to linked experiences based on the peak experience's time stamp and the temporal link information. This process is iterated, with any experience A only updating experience B if they are linked and if experience B's time stamp is larger, shown by (10) and (11):

$$\tau_j = t_i + t_{ij} \tag{10}$$

$$t_j^{k+1} = \min\left(\boldsymbol{\tau}, t_j^k\right) \tag{11}$$

where t_i is the time stamp value of experience i, t_{ij} is the temporal link from experience i to experience j, τ_j is the resultant proposed time stamp value of experience j, $\boldsymbol{\tau}$ is the set of proposed time stamp values for experience j, k is the iteration number, and t_j^{k+1} is the updated time stamp value.

B. Route Finding

To find the shortest route to a goal, a gradient climbing procedure is used, starting at the goal location. The route is stored as a sequence of nodes, with each node corresponding to an experience. Each node is time stamped to represent the estimated time it will take from the robot's current position to reach each node along the route to the goal.

C. Behavior Arbitration

RatSLAM uses the goal route as input to its local movement module in order to pick appropriate movement behavior. The robot chooses the local movement path that is spatially closest to the recalled route, shown in Fig. 7.

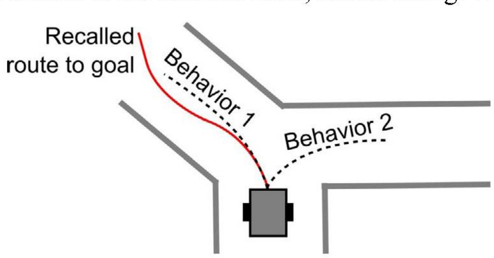

Fig. 7 Behavior arbitration. In this situation the robot would choose to go left, as that local movement behavior more closely matches the proposed route.

D. Route Loss Recovery

As the robot navigates toward the goal, the navigation system is constantly finding the shortest route to the goal based on the current peak experience. New visual scenes can trigger the learning of new experiences even when the robot is traversing a previously traveled path. This allows the robot to more completely learn the route but also results in the temporary loss of a functional temporal map, since time stamps cannot be propagated from a brand new experience that is not yet linked to any others.

In this situation the robot uses the last known 'good' route to the goal to navigate. The robot's position along the route is updated based on time elapsed. If an old route is relied on for longer than a certain time period, it is discarded and the robot returns to exploration in order to acquire a fresh route to the goal.

V. Experiments and Results

Two experiments were run in the indoor environment shown in Fig. 2 to test the ability of the system to perform goal navigation under the conditions described in Section II. These experiments used a Pioneer 2DXE mobile robot equipped with a 50° field of view camera, 180° scanning laser, and wheel encoders. Camera images were down sampled to 12 x 8 pixel grayscale images and classified using a sum of absolute differences template matching system. Local obstacle avoidance was performed using the scanning laser, although this will soon be replaced by a vision based movement system [9]. Computation was shared between the robot's on-board 400 MHz Athlon K6 processor and a 1.1 GHz Pentium III laptop wirelessly connected to the robot. The system ran in real-time with all network iteration, algorithms, and sensory updates running at 7 Hz.

Each experiment consisted of an hour of exploration, followed by navigation to six goal locations (goal navigation commenced with the robot navigating from the top left corner to goal number one). Two different pose cell matrix sizes were used. For the first experiment, a 100 x 40 x 36 (x', y', θ') matrix was used, corresponding to approximately a 25 x 10 meter area. The actual environment was larger than this, meaning it could not be represented without some wrapping of the pose cell matrix and hence collisions. The second experiment used a much smaller 40 x 20 x 36 matrix, roughly corresponding to a 10 x 5 meter area.

A. Results

Fig. 8 and Fig. 9 show the trajectory of the dominant packet of energy through the pose cell matrix for each experiment. Each dashed line shows where visual information has driven the system to close a loop. There are a large number of collisions, especially for the smaller pose cell matrix case. The pose cell matrix also contains multiple representations of the same physical place, although this is easier to see while watching the experiment in real-time.

Fig. 10 and Fig. 11 show the experience maps produced for each experiment. Approximately 5900 experiences were created during each of the 70 minute experiments. The discontinuities visible in the pose cell matrix plots are gone, and multiple representations have been grouped into overlapping areas of the map. Fig. 12 shows one of the temporal maps created during the second experiment for the 40 x 20 x 36 pose cell matrix, for navigation between goal 4 and goal 5. The planned path to the goal is shown by the arrowed line. Despite the nature of the RatSLAM pose cell representations, the goal recall mechanism was still able to plan and execute routes to each goal with a 100% success rate over 12 trials.

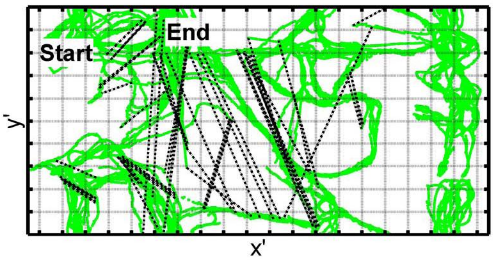

Fig. 8 Path of dominant activity packet through the pose cell matrix for a 100 x 40 x 36 pose cell matrix.

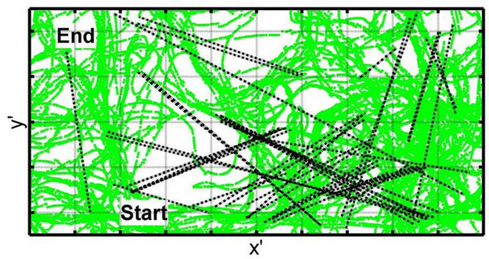

Fig. 9 Path of dominant activity packet for a 40 x 20 x 36 pose cell matrix.

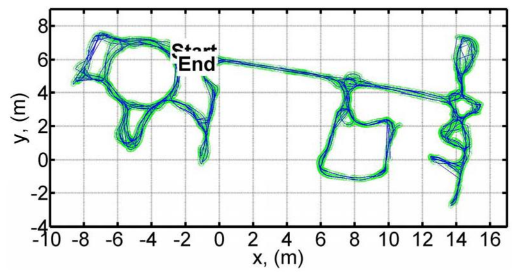

Fig. 10 Experience map produced using a 100 x 40 x 36 pose cell matrix.

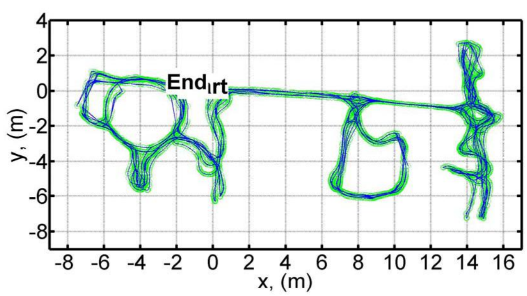

Fig. 11 Experience map produced using a 40 x 20 x 36 pose cell matrix.

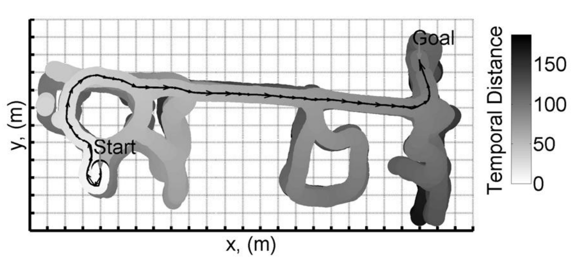

Fig. 12 Temporal map for navigating between goal 4 and goal 5. The temporal distance is measured in seconds.

VI. Adaptation to Environment Change

In this section we present results showing that the experience maps are capable of adapting to simple short term changes in the environment as described in Section IIID. Fig. 13 shows the testing environment and the location where an obstacle (a cardboard box) was added and then removed. During the experiment the robot was instructed to navigate from the start location to the goal location three times. The first instruction came just after the obstacle was placed in the environment, after the robot had spent about twenty minutes exploring the unmodified (no obstacle) environment. The second command occurred six minutes later, with the obstacle still in the environment. The obstacle was then removed, and the robot allowed six minutes to explore the environment, before the third and final trial was started.

Navigating to the goal for the first time, the robot was unaware of the obstacle, as revealed by the planned path to the goal shown in the left of Fig. 14. After multiple attempts to get to the goal via this path, the robot unlearned the inter-experience links near the obstacle's location. The robot then picked the new best route to the goal and reached it. Although the confidence level threshold can be tuned to reduce the time taken to learn an obstacle, at its current level the robot avoids learning shorter term obstacles such as people.

The second time the robot was told to navigate to the goal, it immediately planned the longer route to the goal, having learned that the top route was blocked during the first trial (Fig. 15). The suboptimal behavior at the beginning of the trial is due to the lack of a reactive turn around behavior – currently the robot cannot reactively turn around mid-corridor. During the period of exploration after reaching the goal a second time, the robot learned new links connecting the experiences along the top corridor. When told to navigate to the goal a third time, the system planned a route via the top unblocked path, and the robot successfully navigated to the goal via this route (Fig. 16).

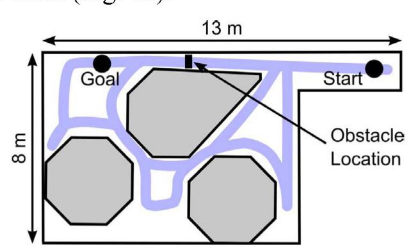

Fig. 13 Floorplan for adaptation experiment, showing obstacle location, start and goal locations, and robot's path.

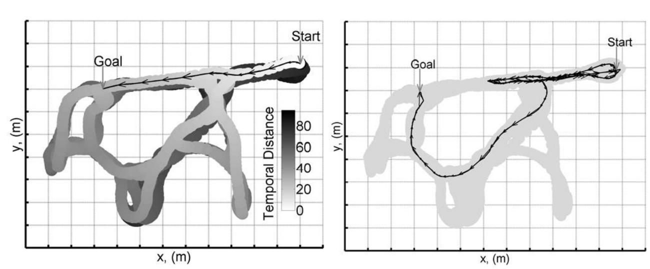

Fig. 14 Temporal map and planned route for trial 1 and actual route taken.

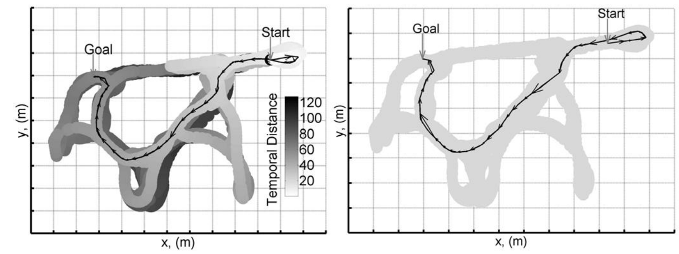

Fig. 15 Temporal map and planned route for trial 2 and actual route taken.

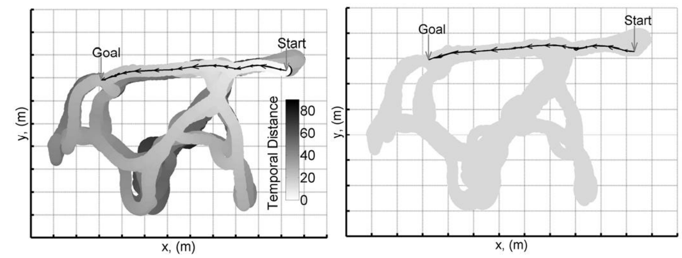

Fig. 16 Temporal map and planned route for trial 3 and actual route taken.

VII. DISCUSSION

The experience mapping algorithm shares characteristics with other research. The approach taken in [10] produces a globally consistent map by minimizing an energy function using only local metrical information. A theoretical proof of convergence is also presented; a similar proof for the experience mapping algorithm is currently being developed. The system appears to be restricted to distinct representations of places, unlike experience mapping which groups together multiple representations of places in the environment built by RatSLAM. A panoramic sensor is also used which significantly reduces the difficulty of loop closing – the largest loop closed was about 20 meters long. The experience mapping algorithm successfully closed a 122 meter long loop during the robot's first traverse of the large indoor environment, with a cumulative odometric error of 8.5 meters and 6.7 meters for the two experiments.

Subjective localization relaxes the requirement that a robot must estimate its pose in a global frame of reference [11]. Instead action respected embedding (ARE) is used to learn a low dimensional representation of vision and range data related to the generation process – such as the movement of the robot and camera through the environment. This is similar to the experience mapping algorithm learning odometric information between experiences that are themselves associated with visual templates. Results are presented demonstrating the system's ability to localize a robot during a simple repeated corridor maneuver.

SLAM can also be performed using a cognitive map [12]. The cognitive map consists of a sequence of links that connect discrete episodic memories, with each episode a sequence of event representations. Each event contains spatial and non-spatial stimuli as well as behavioral actions, much like the experiences and inter-experience links described in this paper. This paper also discusses recognizing when episode recollection fails and recovering from such failures, which is addressed in Section IVD of this paper.

Each experience uses up to 6.7 kB of memory. From experimentation so far it appears that the number of experiences required to represent an indoor office-like environment scales with the area. To sufficiently represent an environment such as shown in Fig. 2 requires about 6000 experiences, corresponding to 40 MB of memory. Computationally the time taken to perform map correction and create a temporal map approximately scales with the number of experiences. At the end of the experiments in Section V the temporal map creation process took about 10 ms.

VIII. CONCLUSION

This paper has described the application of the experience mapping algorithm in a variety of situations. The algorithm complements the core RatSLAM representations and produces maps that are more suitable for goal recall than the original goal memory system. Furthermore the technique allows the environment size to be increased (within limits) without an associated scaling of the pose cell matrix, which has computational advantages. The experience maps can also adapt to simple short term changes in an environment.

REFERENCES

[1] M. J. Milford, G. Wyeth, and D. Prasser, "RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping," *International Conference on Robotics and Automation*, 2004, New Orleans, USA.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," *AAAI National Conference on Artificial Intelligence*, 2002, Edmonton, Canada.

[3] G. Dissanayake, P. M. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localisation and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229-241, June, 2001.

[4] M. J. Milford, G. Wyeth, and D. Prasser, "Simultaneous Localization and Mapping from Natural Landmarks using RatSLAM," *Australasian Conference on Robotics and Automation*, 2004, Canberra, Australia.

[5] D. Prasser, M. Milford, and G. Wyeth, "Outdoor Simultaneous Localisation and Mapping using RatSLAM," *International Conference on Field and Service Robotics*, 2005, Port Douglas, Australia.

[6] M. J. Milford, G. Wyeth, and D. Prasser, "Efficient Goal Directed Navigation using RatSLAM," *International Conference on Robotics and Automation*, 2005, Barcelona, Spain.

[7] M. J. Milford, D. Prasser, and G. Wyeth, "Experience Mapping: Producing Spatially Continuous Environment Representations using RatSLAM," *Australasian Conference on Robotics and Automation*, 2005, Sydney, Australia.

[8] B. Yamauchi and R. Beer, "Spatial Learning for Navigation in Dynamic Environments," *Systems, Man and Cybernetics*, vol. 26, pp. 496-505, June, 1996.

[9] T. Low and G. Wyeth, "Obstacle Detection using Optical Flow," *Australasian Conference on Robotics and Automation*, 2005, Sydney, Australia.

[10] T. Duckett, S. Marsland, and J. Shapiro, "Fast, On-Line Learning of Globally Consistent Maps," *Autonomous Robots*, vol. 12, pp. 287-300, May 2002.

[11] M. Bowling, D. Wilkinson, A. Ghodsi, and A. Milstein, "Subjective localization with action respected embedding," *International Symposium of Robotics Research*, 2005, San Francisco, USA.

[12] Y. Endo, "Anticipatory and Improvisational Robot via Recollection and Exploitation of Episodic Memories," *AAAI 2005 Fall Symposium*, 2005, Arlington, USA.