Return of the Rat: Biologically-Inspired Robotic Exploration and Navigation*

Alejandra Barrera Ramírez and Alfredo Weitzenfeld Ridel
Instituto Tecnológico Autónomo de México (ITAM)
Computer Engineering Department - Robotics and CANNES Laboratories
Río Hondo #1, Tizapán San Ángel, CP 01000, Mexico City, Mexico
abarrera@itam.mx, alfredo@itam.mx

***Abstract* – In this paper we present a biologically-inspired robotic exploration and navigation model based on the neurophysiology of the rat hippocampus that allows a robot to find goals and return home autonomously by building a topological map of the environment. We present simulation and experimentation results from a T-maze tested and discuss future research.**

***Index Terms* – Affordances, mapping, path integration, rat hippocampus, reinforcement learning.**

I. Introduction

While trying to explain the ability of rats to solve spatial problems, Tolman argued in 1948 that rats should have a **cognitive map** in some part of their brain [2]. Then, in 1978, O'Keefe and Nadel argued that such map was located in the **hippocampus** [3]. Experimental work has shown that there exist at least two distinct populations of neurons in the rat hippocampus known as **place cells** and **head-direction cells**. Place cells codify information about physical locations of the animal, while head-direction cells codify orientations of the animal's head [4].

These studies on the rat brain have provided inspiration in developing alternative robotic navigation models to those based on classical approaches, such as metric and topological [1]. Examples of such navigation models based on the rat's hippocampus neurophysiology are those by Burgess and O'Keefe [5], Touretzky and Redish [6], Balakrishnan, Bhatt and Honavar [7], Trullier and Meyer [8], Arleo and Gerstner [9], Gaussier, Revel, Banquet and Babeau [10], Guazzelli, Corbacho, Bota and Arbib [11], and recently Milford and Wyeth [12].

The work presented in this paper is based on a theoretical robotic navigation model developed by Guazzelli et al. [11]. We have extended and implemented this model in an actual robot, teaching the robot to find goals in a T-maze through motivation and reinforcement learning by building a topological map of the environment. In a previous paper [13] we documented the implementation of the model using the NSL simulation system [14] under both simulated and real world environments. In this paper we extend our previous work to allow the robot to return autonomously to the departure location, thus fully automatizing the learning process. The paper is organized as follows: Section II describes the navigation model, Section III describes the process the robot follows to return home, Section IV describes the model implementation, Section V discusses the simulation and experimentation results, and Section VI presents conclusions and future work.

* This work is partially supported by collaboration projects UC MEXUS CONACYT (ITAM – UCSC), LAFMI CONACYT (ITAM – ISC), NSF CONACYT (ITAM – UCI) under grant #42440 and "Asociación Mexicana de Cultura, S. A."

II. The Robotic Navigation Model

Our model allows the rat to determine direction of movement and to build a map-based representation of the environment. Two sub-models carry out these activities: the **Taxon-Affordances model** (TAM) and the **World Graph model** (WG), respectively. Both sub-models are composed of layers of neurons that implement Hebbian [15] and reinforcement learning [16] in order to allow the expression of goal-oriented behavior. The following sections describe the sub-models and their integration.

A. Taxon-Affordances Model (TAM)

The term **affordances**, adopted from Gibson [17], refers to the sensory information that an animal uses to interact with the environment without the need to recognize objects. On the other hand, the term **taxon** refers to the notion of **affordances for movement**, representing all possible motor actions that a rat can execute through the immediate sensing of its environment; e.g., visual sighting of a corridor – go straight ahead; sensed branches in a maze – turn.

Affordances for movement are coded in a linear array of cells called an **affordances perceptual schema** (APS) that represents possible turns from -180º to +180º. Fig. 1 shows the information picked up by the APS when the rat is in the center of an eight-arm radial maze.

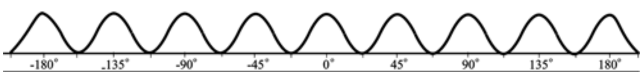

Fig. 1. Affordances perceptual schema when the rat is in the center of an eight-arm radial maze. The rat is able to sense eight different visible arms and eight different affordances (nine if we consider that -180º and +180º are represented separately). Each peak of activity represents a different affordance. The leftmost peak codes turning -180º and the rightmost peak codes turning +180º. The remaining peaks of activity code turns between -180º and +180º in 45º intervals.

Determination of the affordances for movement is based on a local coordinate system that is relative to the rat's head, as shown in Fig. 4(a).

Fig. 2 shows the different layers that compose TAM. The APS sends its output to the **affordances feature detector layer**, a layer of neurons whose activation constitutes a pattern that represents the group of affordances available for the rat at the current time. The activity pattern generated over the affordances feature detector layer is stored within a specific unit in the **affordances state layer**. TAM is able to associate expectations of future reward to specific affordances states in order to allow the expression of goal-oriented behavior. To do this, the model incorporates an **egocentric motivational schema** and a **reinforcement learning** process that will be described in section C. The egocentric motivational schema sends the expectations of future reward associated with the current affordances state to the **action selection schema**, the module that computes the choice of correct affordance; i.e., the one that leads the animal to the goal.

B. The World Graph Model (WG)

The WG model is composed mainly of a place cell layer and a world graph layer. Place cell activity is influenced by path integration information.

Path integration describes the process by which signals generated during locomotion (kinesthetic information) allow the animal to update the position of its point of departure (an environmental anchor) each time it moves in relation to its current position. In this way, path integration allows the animal to return home. As can be seen in Fig. 3, the WG model includes a **path integration module** composed of a **dynamic remapping layer** defined as a two-dimensional perceptual schema representing the particular environment and the anchor coordinates, and a **path integration feature detector layer** where the activation of its neurons constitutes a pattern of kinesthetic information that is the input to the **place cell layer** (PCL). The pattern of activity generated in this layer represents a single place or location in the environment.

The world graph is implemented in the WG model by the **world graph layer** (WGL), whose nodes are created on demand.

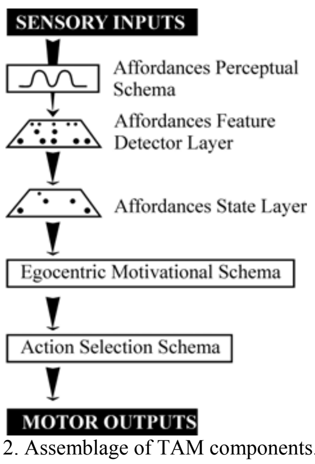

Fig. 2. Assemblage of TAM components.

Every unit in PCL is connected to every node in WGL. Each WGL node can store eight different activity patterns, one for each direction, assuming that the animal can orient itself in eight directions and can experiment different views of the same place. The creation of nodes in the WGL is modulated by the activation of distinct affordances states in TAM and by the presence of a PCL activity pattern not similar to a previously stored one. The arcs in the graph are represented by links between two WGL nodes, and are associated with the orientation of the rat's head when the animal goes from one node to the next one. In the WG model the determination of the direction of the rat's head is based on a global coordinate system, which is relative to an environmental anchor that corresponds to the departure location in the exploration process. Fig. 4(b) shows this global coordinate system.

The WG model incorporates an **allocentric motivational schema** that associates place information to expectations of future reward in order to influence the selection of the next action. The **reinforcement learning** process is carried out similarly to that in TAM. The expectations of future reward are sent to the action selection schema to contribute to the selection of the next direction of movement.

C. Motivation and Learning

The animal's motivation is related to its internal need to eat, which is represented in the model as the **hunger drive**. The hunger drive is increased by the presence of food and reduced by the ingestion of it.

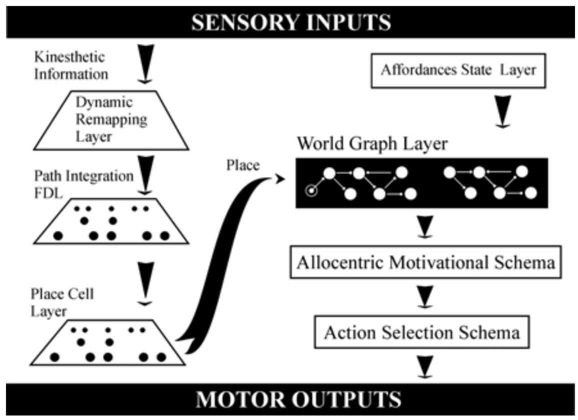

Fig. 3. The WG model. FDL stands for Feature Detector Layer.

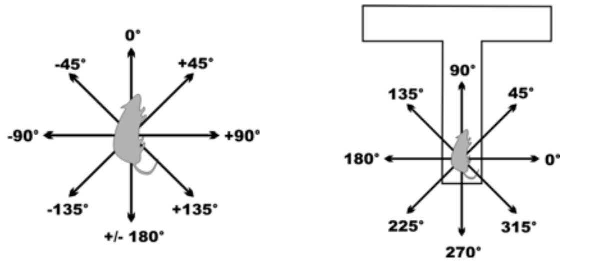

Fig. 4. (a) Local coordinate system used in TAM to determine the relative affordances for movement. (b) Global coordinate system used in WG to build the map-based representation of the external environment, a T-maze, for example. This global system is relative to the departure location (the base of the "T" in the figure).

The model computes the hunger value (*hv*) at every rat step using the formula

$hv = (phv + \alpha * d - r * phv) + i * d,$ (1)

where *phv* is the previous hunger value, α is a constant, *d* is the difference between the maximum value that the drive can take and its previous value, *r* is a reduction constant registered when the rat eats, and *i* is an incentive constant registered when the rat perceives the food.

The amount of **reward** or reinforcement (*rv*) the animal gets by the presence of food is calculated using the formula

$rv = (phv / mhv) * r,$ (2)

where *mhv* is the maximum hunger value.

In order to model the reinforcement learning in TAM, every node in the affordances state layer has an **adaptive-critic architecture** composed of eight actors and an adaptive-critic unit. This unit represents the global expectation of future reward of an affordances state, while the actors are associated to the eight different directions the animal can point to, representing in this way, the particular expectation of the rat to find reward if it moves in the affordance that corresponds to that actor.

Beginning a model iteration, reinforcement is initiated by updating the actor unit's eligibility trace of the current node associated with the last turn the animal had to make to orient itself to its current direction. The update can be an increase (positive reinforcement) if the last movement allowed the perception of food, or a decrease (negative reinforcement) if it did not. The adaptive-critic unit's eligibility trace of the current node is increased or decreased in the same way. After the creation or activation of a node, the reinforcement process is carried out for all the nodes in the layer, updating the weights of the adaptive-critic unit and of the eight actor units. The equation used to update those weights (*w*) at time *t*+1 is:

$w(t+1) = w(t) + \beta * e(t) * rh(t),$ (3)

where β is the learning rate, *e(t)* is the eligibility trace of the adaptive-critic unit or the eligibility trace of the actor unit, and *rh(t)* represents an adjusted reward value that considers the prediction of expectations of future reward at time *t* and at time *t*+1.

In the WG model the reinforcement learning process is carried out similarly to the one done by TAM using (3). If the place visited by the rat after the last turn allowed it to perceive food, the reinforcement is positive, otherwise it is negative.

D. The integrated TAM-WG model

The integrated model, called TAM-WG, combines TAM and WG models, as shown in Fig. 5. The decision to turn to a certain angle is given by a winner-take-all process performed over the integration of activation fields produced by the available affordances (*A*), the drive relevant stimuli (*F*), the expectation of reward derived from TAM (*RT*), the expectation of reward derived form the WG model (*RW*), and a curiosity level (*CL*); *RT* and *RW* contain associated noise factors *n* and *m*, respectively. In this way, the total input *I* to the action selection module becomes

$I(i,t)=A(i,t)+F(i,t)+(RT(i,t)+n)+(RW(i,t)+m)+CL(i,t),$ (4)

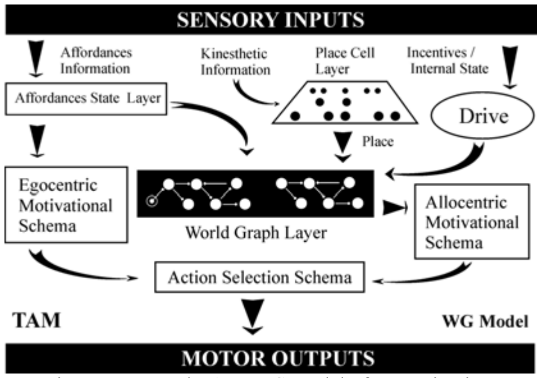

Fig. 5. Integrated TAM-WG model of rat navigation.

where *i* varies from 1 to *N*, the length of the population of cells (80) in the linear arrays used to represent the perceptual schemas, while *t* represents the time variable.

Through *CL*, the WG model considers the fact that the animal may go to places that are not yet represented in the world map. In this way, if the rat is not motivated to go towards a previously experienced place, it will tend to choose, based on its curiosity level, an affordance that leads to a place not yet represented in the map.

III. Reading the Map to Return Home

Consider, for example, the case in which the rat is exploring an environment like the one shown in Fig. 9(a). Suppose the rat goes form location "h" to location "a," building the map presented in Fig. 9(b). Every node in this map is pointed by just one arc, except for node 1, corresponding to the departure location ("h"). Node 2 represents locations "g," "f" and "e". Node 3 corresponds to location "d." Node 4 represents locations "c" and "b," and node 5 corresponds to the goal location ("a").

When the rat reaches location "a," node 5 is active and the return process begins. Initially, the rat's direction is set to the opposite direction's value of the arc pointing to the active node. A new APS is defined and established as both the current and the previous affordances state (AS). As long as the active node is pointed by an arc, the rat processes the following algorithm:

1. If the current AS is different from the previous one,
    a. the node pointed to the active one is set as the new active node;
    b. the rat's direction is set to the opposite direction's value of the arc pointing to the active node.
2. The rat moves and determines the new APS.
3. The previous AS is set to the value of the current AS, and the current AS is set to the new APS.

We will refer to this algorithm in Section V (A), where we explain in detail the return process in the T-maze used to test the model.

IV. Model Implementation

The robotic navigation model was designed and implemented using the **NSL simulation system** [14]. The model was decomposed into different modules as shown in Fig. 6. In general, the input to the TAM-WG model is composed of the current direction of the rat's head (*chA*), the current rat's view (*cV*), and the distance (*dF*) and angle (*tF*) to the food, if it is visible.

The **Drive** module computes the hunger value and the reward value using (1) and (2). The **AffordancesPS** module generates the current APS. The **AffordancesFDL** and the **AffordancesSL** modules correspond to the affordances feature detector layer and the affordances state layer of TAM. Considering the WG model, on the other hand, path integration is carried out by the **DynamicRL** and **PathIntFDL** modules. The kinesthetic information is the input to the **PlaceCellL** module that corresponds to the place cell layer of the model. The **WorldGraphL** module corresponds to the world graph layer representing the topological map of the environment. The **MotivationalS** module computes the input to the action selection schema as indicated in (4). Finally, the output of the TAM-WG model is generated by the **ActionSelectionS** module that determines the next direction of the rat's head (*nhA*), the angle by which it has to turn to point to this direction (*aT*), and the displacement the rat has to undergo to reach its next position (*d*).

The TAM-WG model can interact with a virtual or real environment. The model takes the information that it requires from a **Visual Processing** (VP) module that in turn takes as input the image perceived by either a simulated rat or by a real robot.

The VP module computes the distance and angle to the food (*dF*, *tF*), using the amount of different colored pixels found in the current image. Note that we do a color based visual processing. The amounts of pixels (*cV*) are also needed by the AffordancesPS module to compute the current APS. The VP module sends the head orientation of the robot (*chA*) to several modules of the TAM-WG model, as shown in Fig. 6. The output generated by the ActionSelectionS module of TAM-WG (nhA, aT, d) is sent to a **Motor Control** module so that the virtual or real robot is translated and/or rotated accordingly, affecting the next image that the rat will perceive as well as the current head direction considered by the VP module.

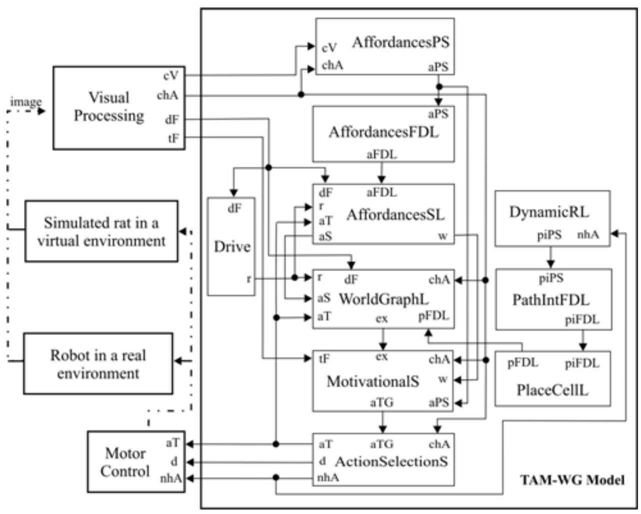

Fig. 6. The modules of the TAM-WG model interacting with a virtual or real environment.

V. Simulation and Experimentation

The experimental environment used to test our model consists of a T-maze. The simulated rat navigates from the base of the "T" to either one of the two arm extremes, and then it returns to the departure location autonomously. This process is repeated in every experiment's trial.

The rat's behavior goes through two phases during each trial: **training** and **testing**. The objective of the first phase is to train the rat to turn to the left arm motivated by the presence of food at the end of the corridor until the rat learns to turn left to that arm. In the testing phase, the food is moved to the end of the right arm in order to test that the rat can unlearn the previous food location while learning the new one. This experiment is inspired on the reversal task documented by O'Keefe in [18]. During the testing phase of the experiment, O'Keefe also moved the food to the right arm, but after some trials in the T-maze, he combined them with trials in an eight-arm radial maze so that the rat may have eight possible arms to visit in some cases and two arms in other cases. The current model described in this paper considers only the T-maze case.

A. *Simulation Results*

Fig. 7 presents the virtual environment used to simulate the model in the T-maze. At each step of the experiment, the simulated rat takes three pictures of the environment: the first one in the current head direction, the second one 90º to the right and the third one 90º to the left.

During the experiment, the simulated rat builds the world graph map, as can be seen in Fig. 8(a). The base of the T-maze is represented in the map by three nodes. From south to north, the first node corresponds to the place of departure; the second one represents the locations between the place of departure and the T junction, and the third node corresponds to the location at the T junction where the rat decides to turn left or right. Each arm of the T-maze is represented by two nodes in the map. The far most node corresponds to the end of the corridor, while the previous one corresponds to locations between the junction and the end.

During the training phase, the food is placed in the leftmost location of the maze. When the rat reaches the T junction, the sight of food makes the rat decide to turn left. The rat repeats this process for 10 trials. After that, it has learnt the position of food (see Table I).

When the testing phase begins, the food is moved to the right arm. Since the rat is not motivated to turn left anymore, its curiosity level for the other arm, not yet represented in the map, makes the rat explore it. In the following 8 trials, the rat goes through an unlearning process, where the expectations of future reward in TAM and in the WG model for the left arm will decrease continuously. During this process, the rat turns to the left during every trial. Finally, in trail number 20 of the experiment the rat decides to turn right in order to ingest the food, starting a relearning process. In the beginning of this process, the expectations of future reward for the right arm are smaller than the combined curiosity and noise levels so that the rat will tend to choose the left or right arm randomly. The relearning process lasts 12 trials. From trail number 32 the expectations of reward for the right arm are the dominant influence in the behavior of the rat, making it choose the right corridor consistently. Fig. 8(b, c, d) shows images of the rat's behavior.

Every time the rat reaches either one of the two arm extremes it proceeds with the process of returning home, following the algorithm described in Section III. Consider now that it starts returning from the left arm, as shown in Fig. 9(a). At this moment, the rat is at location "a" and map node 5 is active, see Fig. 9(b).

The rat's direction is set to 0^o (the opposite direction's value of the arc pointed to node 5, i.e. 180^o); a new APS is defined and set to [0^o] (i.e. the rat can just move ahead); the current and the previous AS are set to [0^o].

As the active node is pointed by an arc and the current AS is equal to the previous one, the rat moves to location "b" and computes the new APS [-180^o, 0^o, $+180^o$] (i.e. the rat can move ahead or return). The previous AS is set to [0^o] and the current AS is set to [-180^o, 0^o, $+180^o$].

As the current AS is different from the previous one, the active node is set to node 4, the rat's direction is set to 0^o (the opposite direction's value of the arc pointed to node 4, i.e. 180^o) the rat moves to location "c" and determines the new APS [-180^o, 0^o, $+180^o$]. The previous AS is set to [-180^o, 0^o, $+180^o$] and the current AS is set to [-180^o, 0^o, $+180^o$].

The rat moves to location "d" and computes the new APS [-180^o, 0^o, $+90^o$, $+180^o$] (i.e. the rat can move ahead, return or turn 90^o to the right); the active node is set to node 3, the rat orients to 270^o (the opposite direction's value of the arc pointed to node 3, i.e. 90^o), moves to location "e" and computes a new APS [-180^o, 0^o, $+180^o$].

The active node is set to node 2, the rat orients to 270^o (the opposite direction's value of the arc pointed to node 2, i.e. 90^o), moves to location "f" and computes a new APS [-180^o, 0^o, $+180^o$]. The rat moves to location "g" and computes the new APS [-180^o, 0^o, $+180^o$]. The rat moves to location "h" and computes the new APS [-180^o, $+180^o$] (i.e. the rat can just return). The active node is set to node 1. As this node is not pointed by any arc, the rat has reached the departure location and now it is ready to begin a new trial of the experiment.

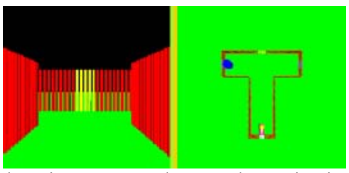

Fig. 7. The virtual environment used to test the navigation model. The left panel shows the current rat's view (robot's camera), while the right one presents an aerial view of the maze (aerial camera).

TABLE I.
THE PERFORMANCE OF THE RAT DURING THE T-MAZE EXPERIMENT

Trial #	Chosen arm	Phase	Process
1 – 10	Left	Training	Learning
11	Right	Testing	Curiosity drive
12 – 19	Left	Testing	Unlearning
20 – 31	Left or right randomly	Testing	Relearning
32 – …	Right	Testing	Relearning

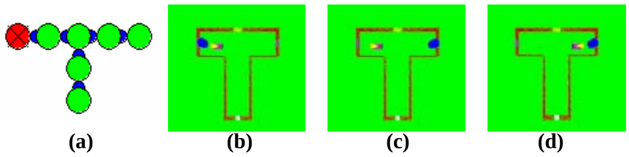

Fig. 8. (a) The map of the T-maze. The bigger circles represent the nodes and the little half circles between nodes represent arc direction. The node marked with an "X" indicates the current node in the map. (b) A typical trial in the training phase: the food is at the leftmost place of the "T" and the rat is approaching it. (c) A typical trial during the unlearning process: the food was moved to the right arm and the rat is approaching the end of the opposite corridor. (d) A typical trial during the relearning process: the food is at the right arm and the rat is approaching it.

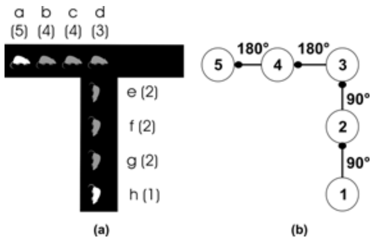

Fig. 9. (a) The path the rat follows to return home in the T-maze. (b) The map read by the rat to return home. Numbers from 1 to 5 identify nodes, while numerical arc label corresponds to rat's head direction.

B. *Robot Experimentation Results*

We tested the model using a Sony AIBO ERS-210 4-legged robot having a local camera. The experiment used to test the model in the real world was the same as the one used in the virtual world. After each trial the robot read the topological map in order to return home autonomously. The robot took three pictures after each step, same as the simulated rat.

A T-maze was built with a width and height of 150 cm, consisting of three corridors having 50 cm of distance between walls. The walls were painted with different colors so that affordances could be easily computed by the robot. During the experiment, the robot built the same world map shown in Fig. 8(a).

The robot's behavior was consistent with the simulated experiment. The robot was trained for 10 trials, and during the testing phase, the length of the unlearning and the relearning processes presents minor variations in relation with the performance shown in Table I. Fig. 10(a, b, c, d) shows pictures of the robot's behavior during the experiment. A "shortened" video can be found in our web site [19].

VI. Discussion

In this paper we have presented an extended version of a robotic navigation model based on the physiology of the rat's brain. We have shown that both the simulated rat and the real robot are able to explore a T-maze, to learn the locations of food, to build a map of the environment and to read the map to return to its departure location autonomously. It is important to say that the whole experiment used to test the model is performed by the rat or robot autonomously. In a previous version of the model we had to move the robot manually to the departure location at the beginning of every trial.

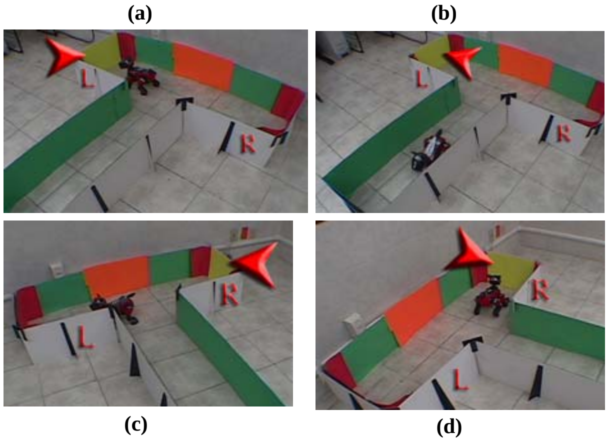

Fig. 10. (a) A typical trial in the training phase: the food is at left (L) and the robot is approaching it. (b) The robot is returning from the left arm. (c) A typical trial during the unlearning process: the food is at right (R) and the robot is approaching the end of the opposite corridor (L). (d) A typical trial during the relearning process: the food is at right and the robot is approaching it.

In terms of learning, our simulation and experimentation results matched qualitatively with those obtained by O'Keefe. After having learnt to turn to a specific arm of the T-maze, our rat, like O'Keefe's real rat, consistently chose the side to which it was trained. However, when the location of food was changed, the rat solved the T-maze by learning to switch gradually to the opposite arm at the maze junction.

In order to simplify the experimental environment we used color recognition as the basis for object identification in computing affordances although we know that rats have a more sophisticated object recognition scheme. We assigned different colors to different objects of interest, such as walls and food. We had to deal with illumination problems in getting the right calibrations assigned to the different colors in the real robot experiment. Additionally, the AIBO robot does not always walk in a straight manner or perceive a consistent number of colored pixels leading to variations in object distance calculations and recognitions. Sometimes, this affected not only the moment when the robot decided to turn at the junction, but also made it turn incorrectly. Furthermore, since the walls were made of cardboard, the robot was able to push the walls around.

In our experiment the rat had only one way to reach the goal location because the departure location was the same in every trial. In this way, every node in the topological map had at most one arc pointing to it, which simplified considerably the process to return home by just reading the map. However, if we consider that the rat can reach the goal location from two different departure locations in different trials of the experiment there will be at least one node of the map pointed by two arcs and therefore the reading of the map will need to be extended in enabling the rat to return home. Consequently, at this point we are considering to extend the model to use the path integration module to implement the return home process. For the nodes pointed by at least two other nodes, we will compare the dynamic remapping perceptual schema (DRps) associated to each of these nodes with the DRps associated to the location of departure. The most similar DRps will indicate the node that must be activated to eventually reach the departure location.

Some other extensions that we plan to implement include the use of the model in the exploration and mapping of more complex mazes and less structured environments with more sophisticated object recognition. We also would like to use spatial landmarks for guiding rat navigation. Finally, it should be emphasized that the motivation behind this work is the quest for inspiration from animal neurophysiology in efficiently solving spatial problems, as in the case of rats, where they have shown advanced learning capabilities that we expect will lead to more advanced robotic navigation models.

REFERENCES

[1] A. Barrera and A. Weitzenfeld. "Biologically-inspired robotic mapping as an alternative to metric and topological approaches." *Proceedings of the 1st IEEE Latin American Robotics Symposium*, pp. 82-87. Mexico City, Mexico. October 28 – 29, 2004.

[2] E. Tolman. "Cognitive maps in rats and men." *Psychological Review* 55, 189-208, 1948.

[3] J. O'Keefe and L. Nadel. "The hippocampus as a cognitive map." Oxford University Press, 1978.

[4] C. Hölscher. "Time, space and hippocampal functions." *Reviews in the Neurosciences*, 2003.

[5] N. Burgess, M. Recce and J. O'Keefe. "A model of hippocampal function." *Neural Networks*, Vol. 7, Nos. 6/7, pp. 1065-1081, 1994.

[6] D. Touretzky and A. Redish. "A theory of rodent navigation based on interacting representations of space." *Hippocampus* 6, 247-270, 1996.

[7] K. Balakrishnan, R. Bhatt and V. Honavar. "A computational model of rodent spatial learning and some behavioral experiments." In: M. A. Gernsbacher & Sharon J. Derry (Eds.). *Proceedings of the Twentieth Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Assoc., 1998.

[8] O. Trullier and J-A. Meyer. "Animat navigation using a cognitive graph." *Biological Cybernetics* 83, 271-285, 2000.

[9] A. Arleo and W, Gerstner. "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity." *Biological Cybernetics* 83, 287-299, 2000.

[10] 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." *Biological Cybernetics* 86, 15-28, 2002.

[11] A. Guazzelli, F. J. Corbacho, M. Bota, and M. A. Arbib. "Affordances, motivation, and the world graph theory." *Adaptive Behavior: Special issue on biologically inspired models of navigation*, vol. 6 (3/4), 435-471, 1998.

[12] M. Milford and G. Wyeth. "Hippocampal Models for Simultaneous Localization and Mapping on an Autonomous Robot." *Proceedings of the 2003 Australasian Conference on Robotics and Automation*. Brisbane, Australia, 2003.

[13] A. Barrera and A. Weitzenfeld. "A Robotic Navigation Model Inspired on the Rat Hippocampus: Motivation, Learning and Mapping." Unpublished.

[14] A. Weitzenfeld, M. Arbib and A. Alexander. "The Neural Simulation Language." MIT Press, 2002.

[15] D. O. Hebb. "The Organization of Behavior: A Neuropsychological Theory." Wiley-Interscience, New York, 1949.

[16] A. G. Barto. "Reinforcement learning." In M. A. Arbib (Eds.), Handbook of Brain Theory and Neural Networks (pp. 804 – 809). Cambridge, MA: MIT Press, 1995.

[17] J. J, Gibson. "The senses considered as perceptual systems." Allen and Unwin, 1966.

[18] J. O'keefe. "Spatial memory within and without the hippocampal system." In W. Seifert (Ed.), Neurobiology of the Hippocampus. New York, Academic Press, 375 – 403, 1983.

[19] A. Barrera and A. Weitzenfeld. "Return of the Rat. Robot Experimentation Results." http://ftp.itam.mx/pub/alfredo/ABarrera/BioRob2006-Video/. 2005.