AVAILABLE AT
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 1373–1388

Neural Networks

www.elsevier.com/locate/neunet

2003 Special Issue

Self-correction mechanism for path integration in a modular navigation system on the basis of an egocentric spatial map

Regina Mudra*, Rodney J. Douglas

Institute of Neuroinformatics, University/ETH Zürich, Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

Received 20 December 2002; revised 15 August 2003; accepted 15 August 2003

Abstract

Classical Computer Science approaches to navigation by autonomous robots continue to make good progress. However, we have only a limited understanding of how navigation is implemented in the neural networks of animals, which still perform very much better in navigational tasks than robots. In this paper we explore the implementation of neural network based navigation in a simple robot. We use a modular navigation system that contains separate representations of visual input and the path integration process. These representations are combined to influence the behavior of a robot. Both representations are encoded within recurrent neuronal networks. The outputs of the representations are vectors of polar values that encode the location of the nearest object, or of a specific place in the environment. The robot manoeuvres in relation to these attended locations, in the context of its egocentric spatial map. During ego-motion towards a goal, the network representation of the goal moves in a counter-movement due to applied motor feedback. The robot's position is continuously compared against its visual input, and mismatches between the visually perceived goal position and its spatial representation are corrected.

Keywords: Navigation; Path integration; Representation; Recurrent network; Self-correction; Robotics

1. Introduction

Exploration is a foundation of animal economies. It is the search for materials to be transformed into the energy and protective structures that sustains them and their societies. Animals often use a strategy that involves exploratory sorties from a protective home base. Consequently methods of pilotage and navigation that enable the animal to forage far from home, are crucial behavioral skills. For example, visual piloting and path integration (Gallistel, 1990), also named dead reckoning, and the related abilities of landmark recognition, location fingerprinting, odometry and compass sense sensory cues are used efficiently by insects and vertebrates for navigation and foraging (Hartmann & Wehner, 1995).

The construction of autonomous robots with navigational and foraging skills similar to animals is a great intellectual challenge, and would have great economic implications. Of course, Computer Science approaches to navigation by autonomous robots continue to make good progress in location detection, environmental mapping, and path planning (Bruce & Veloso, 2002; Choset, 2001; Fox, Burgard, & Thrun, 1998; Montemerlo, Thrun, Koller, & Wegbreit, 2002). However, those approaches are based on explicitly procedural methods, rather than being based on neural networks.

In contrast to the procedural approaches, we have only a limited understanding of how navigation is implemented in the neural networks of animals, which still perform very much better in navigational tasks than robots (Arkin, 1998; Pfeifer & Christian, 1999). The problem, then, for computational neuroscientists is to identify the principles of neural network based navigation in animals, and transpose these to robots. Here, we offer a step in that direction by exploring how a simple robot can manoeuvre in relation to an 'imagined' goal that is not always present in its sensory input; and how visual information concerning that goal, can be used to maintain the calibration of the robot's egocentric representation of the environment. The behavior of the robot is driven by interacting neural networks. We use a modular navigation system that contains separate representations of visual input and the path integration process. These representations are combined to

* Corresponding author. Tel.: +41-1-632-45-95; fax: +41-1-632-11-93.
E-mail address: regina@ini.phys.ethz.ch (R. Mudra).

0893-6080/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2003.08.004

Nomenclature

M	Egocentric spatial map
M_{xy}	Activity of neuron at position (x, y) in egocentric spatial map
m_{xy}	Input to neuron at position (x, y) in egocentric spatial map
α_p	Strengthening factor for angular coupling
α_r	Strengthening factor for radial coupling
α_{P*}	Strengthening factor for $P*$, with $* \in 1, 2, 3, 4$, for the motoric coupling
P_i	Activities of angular pointer neurons
R_j	Activities of radial pointer neurons
p_i	Input to angular pointer neurons
r_j	Input to radial pointer neurons
κ_*	Strengthening factor for nearest neighbor connections in the rings
η_*	Strengthening factor for nearest neighbor connections along the radials
I	Global inhibition
b	Input to global inhibitory neuron
$\mathscr{V}$	Visual map
V_{xy}	Activity of neuron at position (x, y) in visual map
v_{xy}	Input to neuron at position (x, y) in visual map
α_h	Strengthening factor for horizontal coupling
α_v	Strengthening factor for vertical coupling
Z_l	Activity of pointer neuron, left of visual map
Z_r	Activity of pointer neuron, right of visual map
Z_b	Activity of pointer neuron, back of visual map
Z_f	Activity of pointer neuron, front of visual map
z_l	Input to pointer neuron, left of visual map
z_r	Input to pointer neuron, right of visual map
z_b	Input to pointer neuron, back of visual map
z_f	Input to pointer neuron, front of visual map
δ_x	Angular spacing in x direction
δ_y	Angular spacing in y direction
β	Strengthening factor for inhibition
$\mathscr{C}$	Input vector of motor map
C_{stop}	Motor neuron, stop
C_{forward}	Motor neuron, forward
$C_{\text{turn}_{\text{left}}}$	Motor neuron, left turn
$C_{\text{turn}_{\text{right}}}$	Motor neuron, right turn
$x(t)$	Rotatory component of the path
$y(t)$	Translatory component of the path

influence the behavior of a robot. We take our inspiration from the growing literature on rodent navigation, which is one of the most studied animal navigation systems (Redish, 1999).

In 1996, Redish and Touretzky (Redish & Touretzky, 1998a) proposed that the rodent navigation system has five different representations: local view, head direction, path integration, place cells and goal memory (Redish & Touretzky, 1998b) (Fig. 1). These representations work in parallel, are partially redundant, and exchange information. Biological inspired models of the head direction system (Blair, 1996; Zhang, 1996) or place code have been simulated (Samsonovich & McNaughton, 1997), and also tested on robots (Burgess & O'Keefe, 1996; Burgess, Donnett, Jeffery, & O'Keefe, 1997). In these models, path integration is only used as an idiothetic input (i.e. related to the ego-motion of the animal) to learn the allocentric place code (Arleo & Gerstner, 2000; Trullier & Meyer, 2000). These models do not take into account the animal's ability to perform path integration from the moment of its first experience in an environment (Redish, 1999), nor do they explain the transformation from egocentric perception to allocentric place code (Guazzelli, Bota, & Arbib, 2001).

We followed McNaughton (McNaughton et al., 1996; Redish, 1999; Samsonovich & McNaughton, 1997) and represent the path integration process in a two-dimensional attractor network. However, our path integration process is based on a 'pointer-map' architecture (Hahnloser, Douglas, Mahowald, & Hepp, 1999). This architecture has the useful property that it combines directly, in a single recurrent network, two complementary representations: a map, an array of neurons with similar excitatory and inhibitory connections each encoding a stimulus variable, and a vector that encodes an attended location in the map. Thus, we could use the network to perform vector guided navigation (Alyan, Jander, & Best, 2000; Mittelstaedt, 2000; Redish, 1999). In particular, the attentional property can be used to maintain the activity of an imagined goal that is not necessarily present in the robot's sensory input. As the robot manoeuvres in relation to this goal, its location is adjusted by proprioceptive feedback, so providing path integration. Furthermore, the vectorial representations of the egocentric map can be combined with those of object detection in a visual map. In this way the robot's position can be continuously compared against its visual input, and mismatches between the visually perceived goal position and its spatial representation can be corrected.

In the following sections we describe the modular navigation system, and its performance. First, we describe the performance of the egocentric spatial representation alone, in simulation and in an experiment in which the robot has to move to different imagined locations in the environment. Then, in a second series of experiments we placed an obstacle in the environment and let the robot move towards it. During the run we match the vectors of the two representations pointing both to the obstacle and use them in a path integration task to analyze the influence of the visual self-correction in order to improve the navigation accuracy of the robotic system.

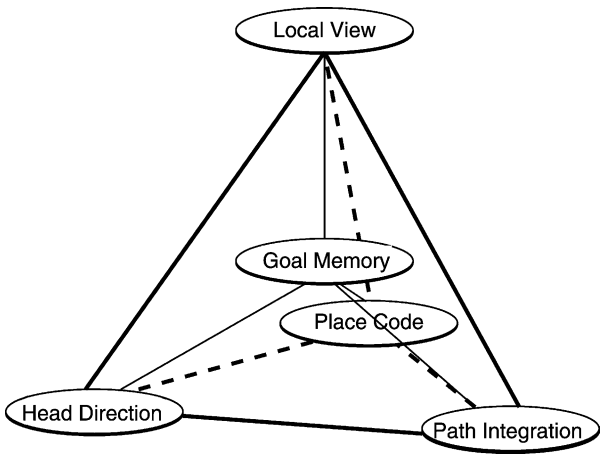

Fig. 1. Navigation between two locations is thought to require multiple representations: local view, head direction, path integration, place code, and goal memory. These five representations operate in parallel, are partially redundant, and exchange information.

2. The modular navigation system

The modular navigation system is implemented on a two-wheeled circular *Khepera* robot.

The navigation system comprises three interacting maps (Fig. 2): a spatial map; a visual map, and a motor map. Both the spatial map and the visual map have a pointer-map architecture. (Hahnloser et al., 1999). This network contains two groups of neurons. The first group, the 'map neurons', forms the representational map for the input pattern, and the second, 'pointer neurons', form the components of a vector pointing towards the highest hill of activity in the map.

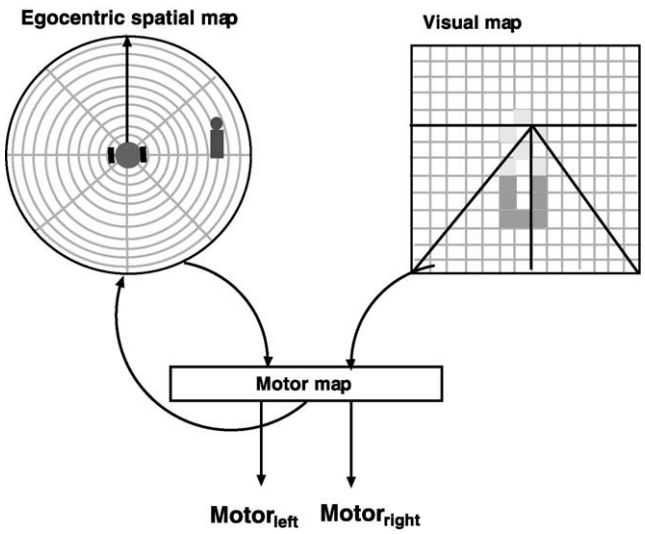

Fig. 2. The navigation system comprises an egocentric spatial map, a visual map and a motor map. The spatial map is centered on the robot and represents the location of the goal object in the environment as a hill of neuronal activity within the map. The visual map views the goal object in relation to a horizon (horizontal line), and a potential collision zone (diagonal lines meeting at vanishing point). The egocentric map and the visual map both drive the motor map using vector encodings of the location of the attended goal within their respective representations. Proprioceptive sensing of motor output modulates the nearest neighbor connections within the egocentric spatial map, resulting in feedback adjustment of the goal location.

The spatial map (Fig. 2, 'Egocentric spatial map') represents the egocentric environment. Its pointer vector attends to a dynamic location in the environmental representation that is encoded as a hill of neuronal activity. This map location usually represents a goal, or a goal position in the environment. Target attention is possible even when the target is out of sensory (visual) contact. In this sense, the robot can attend to, and manoeuvre in relation to, an imagined goal that is on the map.

The visual map (Fig. 2, 'Visual map'; and Fig. 7) is derived from the input of a CCD camera, which has a 50° field of view. The camera is oriented at $-10°$ with respect to the horizontal plane. In this way, the vertical axis of the input scene covers a region that begins close to the front of the robot and extends beyond the robot's horizon. The visual map represents the scene as motor actions based on a triangular potential collision region. The base of this triangle corresponds to the front of the robot, and the apex of the triangle lies at the vanishing point on the robot's horizon.

The output of the spatial and visual map drive a motor map, which in turn controls the speeds of the robot's two wheels. Motor feedback from the shaft encoders is used to update the goal location on the spatial map. This is achieved by changing the strengths of nearest neighbor connections within the neuronal map. These changes have the effect of moving the hill of neuronal activation that represents the location of the goal.

2.1. Egocentric spatial map

2.1.1. One-dimensional pointer-map

The implementation of the egocentric spatial map is an extension of the one dimensional recurrent pointer network, that we have described (Hahnloser et al., 1999) previously. We will briefly review the principles of operation of this network.

The one dimensional network (Fig. 3A) consists of a 1D map containing N map neurons with activities $M_1,\ldots,M_N$ and two pointer neurons with activities P_1 (left) and P_2 (right):

$$\dot{M}_x = -M_x + m_x + \alpha(P_1^+\cos\delta_x + P_2^+\sin\delta_x) - \beta\sum_x M_x^+ \quad (1)$$

$$\dot{P}_1 = -P_1 + p_1 + \alpha\sum_x M_x^+\cos\delta_x \quad (2)$$

$$\dot{P}_2 = -P_2 + p_2 + \alpha\sum_x M_x^+\sin\delta_x \quad (3)$$

All neurons of the attractor network are linear threshold neurons. The activity of the topologically arranged map neurons is determined by the external input m_x and the activities of the two pointer neurons P_1 and P_2 with

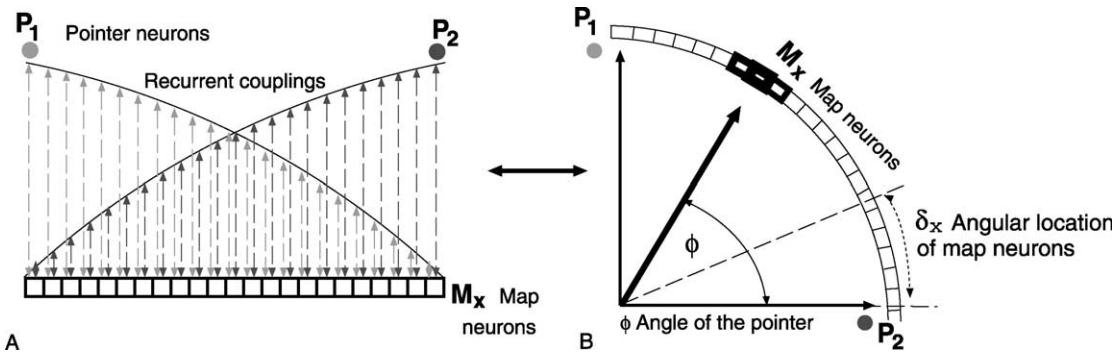

Fig. 3. (A) A one-dimensional recurrent pointer network comprises map neurons M_x and the two pointer neurons, P_1 (left) and P_2 (right). All neurons are linear threshold neurons. The strength of the reciprocal, excitatory, synaptic coupling between pointer neurons and map neurons is given by trigonometric functions. The pointer neuron P_1 has a stronger coupling to the left side of the map, while the pointer neuron P_2 has a stronger coupling to the right side of the map. The map neurons receive external sensory input m_x (not shown), and the pointer neurons receive external attentional input p_1 and p_2 (not shown). The map neurons also receive a global inhibition (not shown). (B) Map neurons M_x lie along the first quadrant of the unit circle, with angular location $\delta_x = (x-1)\pi/2N$ for the x-th neuron in the map with $x \in [1,\ldots,N]$. The pointer, which points to the maximum of activity in the map, has angle ϕ.

the inputs p_1 and p_2. Between pointer neurons and map neurons, symmetric, excitatory synaptic weights are given by trigonometric functions (Lewis & Kristan, 1998a,b): $\cos\delta_x$ and $\sin\delta_x$, where $\delta_x = (x-1)\pi/2N$ with $x \in [1,\ldots,N]$ defines the angular spacing of the map neurons around the unit circle, as shown in Fig. 3B. The coupling factor α defines the strength of the synaptic coupling, while β is the coupling factor indicating the strength of the global inhibition. There is no direct synaptic coupling between the map neurons.

Trigonometric synaptic weights are chosen so as to support a vector whose direction encodes the angular position of the highest neuronal activity in the map. Consequently the neurons of the map are arranged along the first quadrant of the unit circle (Lewis & Kristan, 1998a). The vector components are encoded by the activities of the two pointer neurons (P_1, P_2). The map neurons and the pointer neurons form a recurrently connected network. Consequently, the direction of the pointer, and the location of activity on the map are intimately linked (Hahnloser et al., 1999). The advantage of this architecture for our spatial and visual maps is that place and vectorial encodings are intimately combined.

In the network described so far, the hill of activity will remain stationary at one location, for a constant input. This means that the network can memorize a goal location, but it has no internal means of shifting the location. We modified the architecture to provide controlled goal motion, by introducing excitatory lateral synaptic coupling between nearest neighbors (κ_n in Eq. (4)) on the map (Fig. 4A). When the coupling is symmetrical, then as in the uncoupled case, the hill of activity remains stationary on the map. Asymmetric coupling results in motion of the hill towards the more

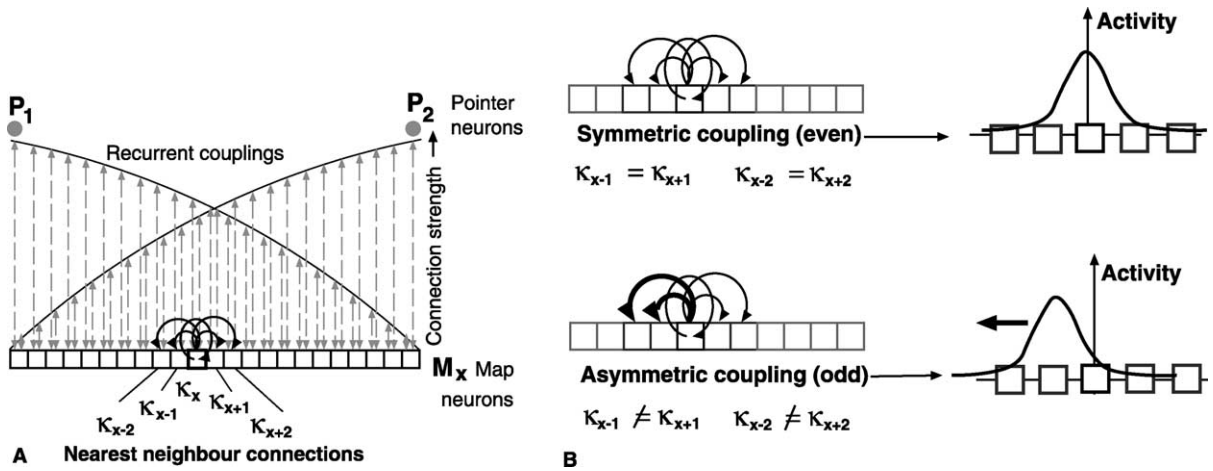

Fig. 4. (A) A one-dimensional recurrent pointer map with nearest neighbor connections has two pointer neurons P_1 and P_2, which encode the angular positions of the map neurons with which they are recurrently coupled. In addition to connections with their pointers, map neurons have nearest neighbor connections to one another. The strengths of these couplings are κ_{-2}, κ_{-1}, κ_0, κ_1 and κ_2. (B) Nearest neighbor connections have two configurations: symmetric coupling, in which the strengths of the nearest neighbor connections on both sides are the same; and asymmetric coupling in which the strengths to one side are stronger than to the other. Symmetric coupling holds the neuronal activity at a fixed position. Asymmetric coupling causes the activity to shift towards the side of stronger coupling. This control of location of activity is used to update the attended position in the map in relation to the movement of the robot.

strongly coupled neighbors. In this way the direction and speed of motion can be controlled. This principle can now be used to update the attended position in the map in relation to the movement of the robot. Zhang (1996) has used a similar approach for head-direction encoding in a one dimension network without a pointer.

The mathematical description of a one-dimensional recurrent pointer map with nearest neighbor connections can be summarized as follows:

$$\dot{M}_x = -M_x + m_x + \alpha(P_1^+ \cos\delta_x + P_2^+ \sin\delta_x) + \kappa_0 M_x^+ + \kappa_1 M_{x+1}^+ + \kappa_{-1} M_{x-1}^+ + \kappa_2 M_{x+2}^+ + \kappa_{-2} M_{x-2}^+ - I \quad (4)$$

$$I = i + \beta \sum_x M_x^+ \quad (5)$$

$$\dot{P}_1 = -P_1 + p_1 + \alpha \sum_x M_x^+ \cos\delta_x \quad (6)$$

$$\dot{P}_2 = -P_2 + p_2 + \alpha \sum_x M_x^+ \sin\delta_x \quad (7)$$

2.1.2. Two-dimensional pointer-map

We use a periodic extension of the pointer-map network with nearest neighbor connections to support the egocentric spatial map (Fig. 5). The two-dimensional representation imposed on this network is in polar coordinates. One (circular) dimension of the network encodes the angle ($\pm 180°$) of the object with respect to the orientation of the robot. The other (radial) map dimension encodes the distance from the robot to the object. The maximum distance was chosen to be half of the length of the robot's arena.

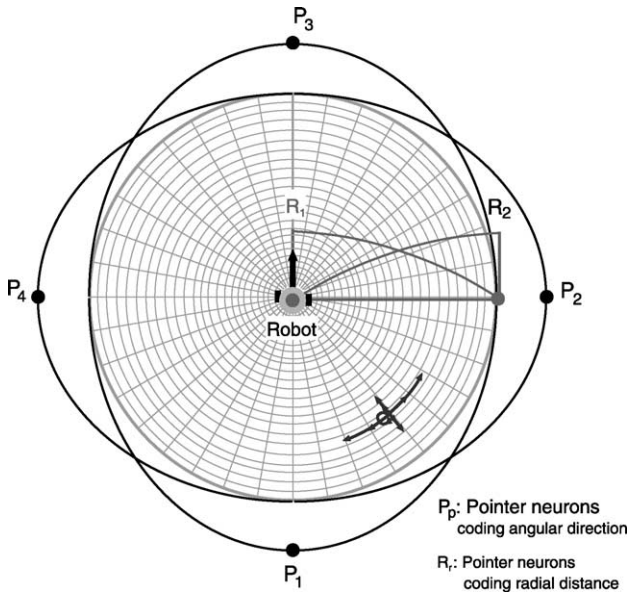

Fig. 5. The recurrent pointer network of the egocentric spatial map has a circular structure. Two of the angular pointer neurons P_i span a quadrant of the unit circle and each further angular pointer neuron is used to span a further quadrant of the unit circle. The resulting structure is a ring of map neurons with four pointer neurons encoding the four directions: back, right, front and left. This particular arrangement of the pointer map uses cosine functions for the strengths of the recurrent coupling between pointer neurons P_i and map neurons. A second set of pointer neurons R_j is used to span the rings, and so encodes distance. Each node in the mesh corresponds to one map neuron, each of which has additional connections to its nearest and second-nearest neighbors in the same ring, and to its radially nearest and second-nearest neighbors, as indicated for one neuron by small arrows.

Four pointer neurons P_i, with $i \in 1,2,3,4$, encode the center of activity in the four quadrants for all rings. The strength of their recurrent connections with the map neurons are set according to the cosine function with the angular spacing $\delta_x = (x-1)\pi/2N$ with $x \in [1,\ldots,N]$. The overall strength of the recurrent coupling to the pointer neurons P_i is given by α_p.

The multi-ring arrangement that encodes distance is coupled to another pointer pair R_j with $j \in 1,2$. The excitatory synaptic weights between this radial pointer pair and the map neurons in the different rings, are as for the one-dimensional case $\cos\delta_x$ and $\cos\delta_y$, where

$$\delta_x = \frac{\pi}{2}\left(\frac{x}{N} + i\right)$$

and

$$\delta_y = \frac{\pi}{2}\left(\frac{y}{N} - j\right)$$

with $x, y \in [1,\ldots,N]$. The factor α_r is the recurrent coupling strength to this pointer pair R_j. Only one pair of pointer neurons is necessary to encode the distance. Each additional pointer pair would only strengthen the radial recurrent coupling, but such an effect can also be achieved if the single radial pointer pair receives positive input. The global inhibition I is given by the sum of activities of all map neurons and the inhibitory input b, weighted with the inhibitory coupling factor β.

The complete mathematical formalism to describe the circularly structured attractor network used to build the egocentric spatial map is as follows:

$$\dot{M}_{xy} = -M_{xy} + m_{xy} + \sum_{i=1}^{4} \alpha_p \left(P_i^+ \left|\cos\frac{\pi}{2}\left(\frac{x}{N} + i\right)\right|\right) + \sum_{j=1}^{2} \alpha_r \left(R_j^+ \cos\frac{\pi}{2}\left(\frac{y}{N} - j\right)\right) + \sum_{k=-2}^{2} \kappa_k M_{x+k,y}^+ + \sum_{l=-2}^{2} \eta_l M_{x,y+l}^+ - I \quad (8)$$

$$I = b + \beta \sum_{s=1}^{N} \sum_{t=1}^{N} M_{st}^+ \quad (9)$$

$$\dot{P}_i = -P_i + p_i + \alpha_p \sum_{x=1}^{N} \sum_{y=1}^{N} M_{xy}^+ \left|\cos\frac{\pi}{2}\left(\frac{x}{N} + i\right)\right| \quad (10)$$

$$\dot{R}_j = -R_j + r_j + \alpha_r \sum_{x=1}^{N} \sum_{y=1}^{N} M_{xy}^+ \left|\cos\frac{\pi}{2}\left(\frac{y}{N} - j\right)\right| \quad (11)$$

where $\dot{M}_{xy}$ is the temporal derivative of the activity of the neuron at position (x, y) in the egocentric spatial map.

The network can be unfolded into a square, $N \times N$ matrix if each ring contains $x_{\max} = N$ neurons and the distance is also coded with $y_{\max} = N$. This square form of the network was chosen to simplify the implementation of the algorithm.

Each map neuron also receives nearest-neighbor connections from the first and second neighbor map neurons along the radials and in the rings. The values of κ_{-2}, κ_{-1}, κ_0, κ_1 and κ_2 specify the connection strengths to the neighbors on both sides within the rings. Radially, the coupling factors η_{-2}, η_{-1}, η_0, η_1 and η_2 define the synaptic coupling strengths to the nearest neighbors.

2.1.3. Dynamic modification of lateral excitatory coupling

During the movement of the robot, the translatory and rotatory components of the movement of the robot are obtained by odometry. The rotatory component is used to modify the symmetry of the nearest neighbor connections in the rings while the translatory component is used to modify the symmetry of the nearest neighbor coupling along the radials. Consequently the hill shifts in a counter motion to that of the robot as it moves towards the attended location (Guazzelli et al., 2001). The shifting hill of activity can be interpreted as a representation of path integration. The interaction between the egocentric spatial map and the motor map is shown in Fig. 6.

The strength of the nearest neighbor couplings is updated in relation to the extracted rotation $x(t)$ of the perceived motor feedback:

$$\begin{aligned}
\kappa_{-2}(t) &= \max\left(0, \frac{1}{2} k_{sr} m_l |x(t)|\right) + k_{ar}\kappa_{-2}(t-1) + k_{cr} \\
\kappa_{-1}(t) &= \max(0, k_{sr} m_l |x(t)|) + k_{ar}\kappa_{-1}(t-1) + k_{cr} \\
\kappa_0(t) &= 0 \\
\kappa_1(t) &= \max(0, k_{sl} m_r |x(t)|) + k_{al}\kappa_1(t-1) + k_{cl} \\
\kappa_2(t) &= \max\left(0, \frac{1}{2} k_{sl} m_r |x(t)|\right) + k_{al}\kappa_2(t-1) + k_{cl} \\
\kappa_N(t) &= 0 \text{ for } |N| > 2
\end{aligned} \tag{12}$$

The strength of the self excitation is $\kappa_0 = 0$, while the strengths of the couplings for the four nearest neighbors κ_* are given by three terms. The first term k_s, with $* = r$ for couplings to the right side and $* = l$ for couplings to the left side, depends entirely on the rotatory motion component, which is multiplied by the motor settings m_r or m_l. These settings are for turns to the right $m_r = -1, m_l = 1$ and for turns to the left $m_r = 1, m_l = -1$. The signs of the motor settings are used to define the signs of the nearest neighbor couplings κ_* that decide between left and right shifts of the hill of activity. During a left turn, the couplings to nearest neighbors on the right κ_{-1} and κ_{-2} are strong, while couplings on the left κ_1 and κ_2 are set to a constant low value. In a right turn the coupling strengths are reversed. The second

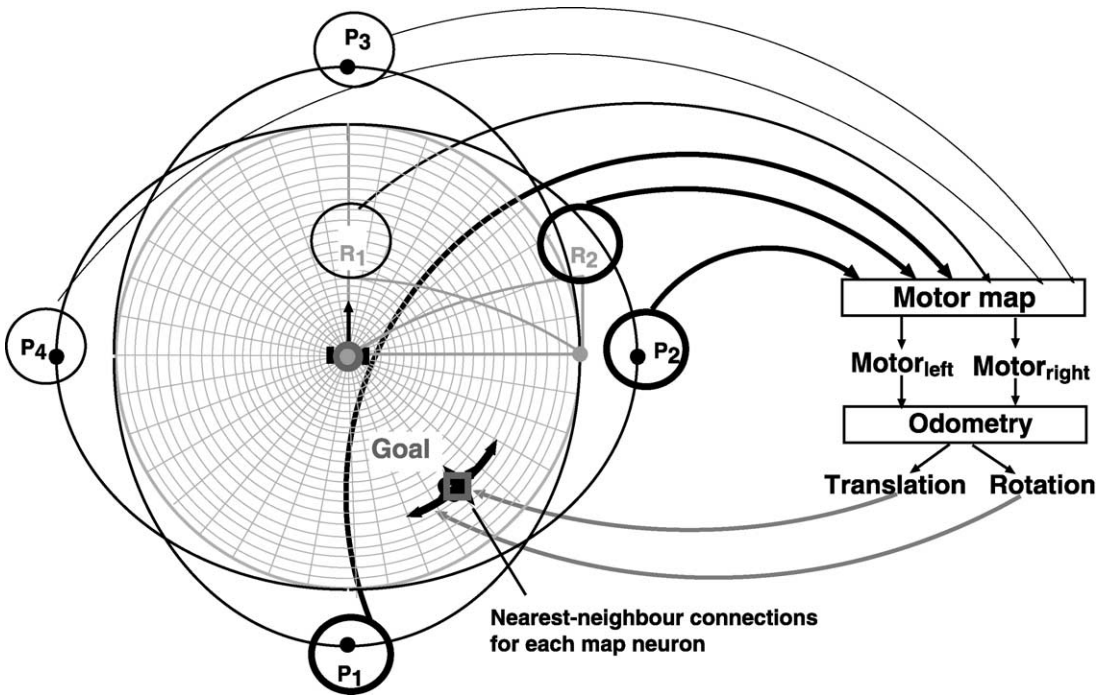

Fig. 6. The egocentric spatial map and the motor map are coupled in a feedback loop. The activities of the pointer neurons P_1, P_2, P_3 and P_4, and the pointer neurons R_1 and R_2, encode the angle and distance of the goal respectively. The activity of the pointer neurons is high if the goal is inside the part of the map they encode. The pointer neurons drive the motor map. The translatory and rotational component of the robot's movement are extracted by odometry and fed back to the egocentric spatial representation, in order to move the goal location. The rotatory component is used to modify the nearest neighbor connections along the rings, while the translatory component influences the change in the strengthening of the radial nearest neighbor connections.

term k_{a*} is chosen in relation to its strength in the previous timestep. The last term k_{c*} is a constant term. The factors $k_{sl} = 3.3$ and $k_{sr} = 3.2$ are made unequal to compensate for a slight leftward drift of the robot due to defects in its mechanical construction which were detected before the present experiments were begun. The other factors are $k_{al} = 0.3$, $k_{ar} = 0.3$, $k_{cl} = 0.25$ and $k_{cr} = 0.25$. During the turn phase, the nearest neighbor couplings along the radials are kept constant with $\eta_* = 0.25$ and the self excitation is set to zero with $\eta_0 = 0$.

When the robot starts to move forward, the nearest neighbor couplings change. The concentric nearest neighbor couplings are kept constant at $\kappa_* = 0.25$ and the self excitation is set to $\kappa_0 = 0$. Meanwhile, the nearest neighbor couplings η_* along the radials are set according to the translatory component $y(t)$ of the robot's movement, as follows:

$$\begin{aligned}
\eta_{-2}(t) &= \max\left(0, \frac{1}{2}k_{sf}m_l y(t) + k_{af}\eta_{-2}(t-1) + k_{cf}\right)\\
\eta_{-1}(t) &= \max(0, k_{sf}m_l y(t) + k_{af}\eta_{-1}(t-1) + k_{cf})\\
\eta_0(t) &= 0\\
\eta_1(t) &= \max(0, k_{sn}m_r(-y(t)) + k_{an}\\
&\quad \eta_1(t-1) + k_{cn})\\
\eta_2(t) &= \max\left(0, \frac{1}{2}k_{sn}m_r(-y(t)) + k_{an}\eta_2(t-1) + k_{cn}\right)\\
\eta_N(t) &= 0 \text{ for } |N| > 2
\end{aligned} \tag{13}$$

Again the strengths of the nearest neighbor couplings η_* contain three terms. All these strengthening factors are k_*. The k_{sn} are terms defining the strength of coupling to the inner nearer neighbors and k_{sf} define that to the outer neighbors. The first k_{s*} is set in relation to the translatory component the robot had during its movement. The forward motion of the robot is defined by $m_r = m_l = 2$ which is equivalent to a speed of approximately 2 cm/s. The second term k_{a*} is chosen as the strength of the coupling one timestep before, and the last term k_c is constant. The sign of the coupling is given by the sign of the translatory component: it is positive if the robot moves forward and negative if the robot moves backwards. The factors are $k_{sn} = k_{sf} = 1$, $k_{an} = k_{af} = 0.4$ and $k_{cn} = k_{cf} = 0.28$.

2.2. Visual map

A two-dimensional recurrent pointer map is used to represent the attended object within the visual input also as a vector encoding the 'where' information of the object. One white object is placed at the goal location within an empty environment of the size of 0.95 m × 1.35 m, with a gray floor and white walls. The on-board camera provides input images to the visual map. In the visual map the object is characterized as groups of edges by a three step pre-processing stage consisting of filtering, size reduction and threshold-normalization (Mudra, Hahnloser, & Douglas, 1999).

The two-dimensional visual pointer-map network $\mathscr{V}$ consists of 25 × 25 neurons. The input to this map is provided by 131 × 131 pixel images. These images are filtered along both axes, using a one-dimensional differential filter, having the kernel 1/10[−1 −4 −5 0 5 4 1]. Then the image size is sub-sampled to 25 × 25 pixels by summing over 5 × 5 sub-matrices and thresholded at one standard deviation above the mean of the pixel values.

The visual map $\mathscr{V}$ is connected to two sets of visual pointer neurons (named Z_* in this map). There are two horizontal pointer neurons Z_l (left) and Z_r (right), and two vertical pointer-neurons Z_b (back) and Z_f (front). The vertical pointers are named 'back' and 'front' because, due to the inclination of the camera, the vertical axis of the image can be interpreted as distance away from the robot. The map receives input v_{xy}, and the pointers receive feedforward attentional inputs z_b, z_l, z_r and z_f.

The dynamic network equations for the two-dimensional recurrent pointer network equations for the visual map are:

$$\begin{aligned}
\dot{V}_{xy} = &- V_{xy} + v_{xy} + \alpha_h(Z_l^+\cos\delta_x + Z_r^+\sin\delta_x)\\
&+ \alpha_v(Z_b^+\cos\delta_y + Z_f^+\sin\delta_y) - \beta\sum_{x,y} V_{xy}^+
\end{aligned} \tag{14}$$

$$\dot{Z}_* = -Z_* + z_* + \sum_{x,y} V_{xy}^+ f(*) \tag{15}$$

Here $\dot{V}_{xy}$ is the temporal derivative of the activity of the neuron at position (x, y). The superscript $+$ denotes the rectification of the threshold neurons. α_h and α_v are the horizontal and vertical coupling strength factors that regulate the strength of recurrent connections between pointer neurons Z_* and the visual map neurons V. The $*$ denotes the four pointer neurons $(* = l, r, b, f)$ and $f(*)$ denotes $(\alpha_h\cos\delta_x)$, $(\alpha_h\sin\delta_x)$, $(\alpha_v\cos\delta_y)$ and $(\alpha_v\sin\delta_y)$, respectively.

The angles $\delta_x = (x-1)\pi/2N$ ($\delta_y = ((y-1)\pi)/2N$) generate a regular spacing of synaptic weights between map neurons and the horizontal (vertical) pointer neurons.

The significant property of the processing in the visual map is that a negative input $z_b = -5$ to the attentional pointer Z_b can be used to bias the processing of the network towards the most salient region in the image. In the context of manoeuvring, a salient region is the base of the nearest object, as shown in Fig. 7. The base edge is selected because of attentional bias. The coupling between the map neurons and the vertical pointer is chosen to be three times stronger than the horizontal coupling ($\alpha_v = 3\alpha_h$).

In this way the visual map combines salient ('what') and 'where' processing and outputs a vector that can be used for

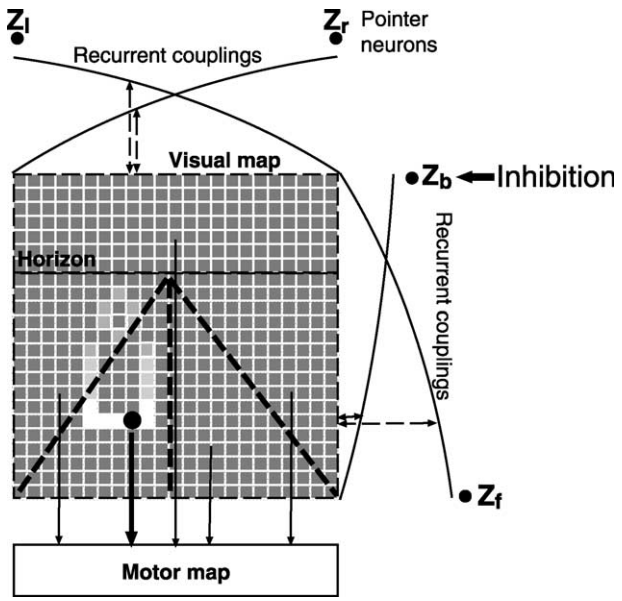

Fig. 7. The pre-processed visual input is fed into a 25 × 25 map of neurons, in which the goal location is encoded by the recurrently coupled pointer neurons. The vertical coupling is set three times stronger than the horizontal coupling ($\alpha_v = 3\alpha_h$). This arrangement results in a stronger weighting of horizontal edges as compared to vertical edges. The input to the pointer neuron Z_b is chosen to be negative, $z_b = -5$, so that the attentional pointer is attracted more strongly to input in the lower part of the image. In this way the pointer-map tends to focus on the base of the goal object (white), and neglect the upper parts of objects, and objects which are further away (light grey).

avoidance or attraction, and visual correction of path integration.

2.3. Motor map

Khepera (K-Team, 1995) has two wheels mounted on a common axis. The wheels are driven separately by two motors. In our model, these motors are driven by a motor map that is a kind of Winner-Take-All (WTA) look-up table (Fig. 8). Within the map, each motor neuron encodes a particular motor setting for the left and right motor. The four motor control neurons are the ones for *go forward* ($m_l = 2$, $m_r = 2$), *stop* ($m_l = 0$, $m_r = 0$), *turn right* ($m_l = 1$, $m_r = -1$) and *turn left* ($m_l = -1$, $m_r = 1$), where a speed setting of one stands for approximately 1 cm/s. The motor map input is given by a vector $\mathscr{C} = (C_{\text{stop}}, C_{\text{forward}}, C_{\text{turn}_{\text{left}}}, C_{\text{turn}_{\text{right}}})$, where each component specifies an input to one specific motor neuron. The input to the motor neurons is derived from the activities of the pointer neuron activities from the visual map and the egocentric spatial representation, further details will be described for the different applications. The outputs of the motor map are $\text{Motor}_{\text{left}}$ and $\text{Motor}_{\text{right}}$, which are equal to m_l and m_r of the winning motor neuron. The WTA property ensures that the motor neuron receiving the largest input wins, and so its encoded motor settings become active. The output of the motor map is then one pair of motor settings given by the winning motor neuron.

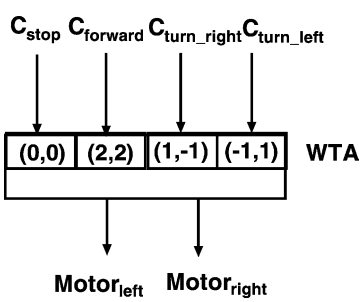

Fig. 8. The motor map is a Winner-Take-All network, in which each neuron encodes a pair of motor settings for the left and right motor. The motor map receives the input vector $\mathscr{C}$, containing the four elements C_{stop}, C_{forward}, $C_{\text{turn}_{\text{right}}}$ and $C_{\text{turn}_{\text{left}}}$, which drive motor neurons with the corresponding motor settings. The motor neuron with the highest input C_* wins, and provides the actual motor setting.

3. Methods

We tested our navigation model on the miniature *Khepera* robot (K-Team, 1995). This robot has a diameter of 5.4 cm and a height of 10 cm including a video camera and its mounting board. The camera was oriented at $-10°$ with respect to the horizontal plane. The *Khepera* is connected via a light and flexible serial connection cable to host PCs and an external power supply. In this 'tethered' mode, the robot is simply a platform for sensory input and motor actuation. Computationally intensive processing is performed on the static host PCs. *Khepera* has two wheels powered by DC motors. They are side-mounted on an axis of symmetry. The arrangement of the wheels allows the robot to rotate about its central axis, and to execute smooth curves in both the forward and backward direction. Shaft encoders on each wheel measure their speeds, and total distances covered.

The robot operated in an empty arena of 0.95 m × 1.35 m with a gray floor and white walls. Its behavior was monitored by an observation CCD camera mounted in a central position above the arena.

The navigational processing of the robot was performed on two 600 MHz PCs. A third 1 GHz PC was used for monitoring. For convenience and speed of development, most of the functionality of the robot was programmed in MATLAB (Mathworks, Inc., Natwick, MA). The trajectory of the robot was measured using the Kanade–Lucas–Tomasi Feature Tracker (KLT) (Birchfield, 1998), a tracking software package written in C.

To facilitate tracking, three LEDs were mounted on top of the robot in a triangular configuration. The triangle points in the forward direction, which is also the viewing direction of the on-board CCD camera. The object at the goal position was also marked with an LED.

During a test run, the sequence of positions of the three LEDs was stored. Snapshots at the start and end of

the run were used to extract the absolute length of the path of the robot. The observation system was calibrated to compensate for the non-linear distortion of the lens, which had a viewing angle of 120° (Mudra, 2002).

The observation process extracted the start position and heading of the robot. This origin and direction information was used to define the observed egocentric spatial frame of the robot. At the start of a trial, the robot was given the goal position in network coordinates of this egocentric spatial map. It then moved to its estimate of this goal position, and halted. A translational error r_{error} and a rotational error α_{error} were computed in relation to the start position, i.e. the origin of the egocentric map. Negative values are undershoots in angle or distance, while positive values are overshoots.

4. Results

4.1. Performing the goal task in simulation

We begin by demonstrating the operation of the network under controlled conditions, by simulating the actions and proprioceptive feedback of the robot. We examine the case of the robot moving towards an imagined goal location in the environment.

The path integration over the egocentric spatial map during this task is shown in Fig. 9.

The initial situation in the environment is shown in B1. The robot is located at an arbitrary location, but 'aware' that there is an object (or goal location) behind it on the right. A1 shows the initial situation in the egocentric spatial map, where a hill of activity encodes the location of the attended object. The activity that corresponds to this attended object has been initialized at the appropriate map location by the experimenter, by applying a brief external input to the relevant neurons. The dynamics of the pointer-map ensure that activity is maintained (memorized) in the spatial map, even after the initializing pulse has ended.

The activity of the six pointer neurons (four angular and two radial) now encode a vector pointing to the goal. These six components of the vector activate the motor neurons in the motor map and so cause the robot to act. The robot moves towards the goal by the shortest path. It begins by rotating towards the right (Fig. 9 B2). During the turn, the counter-clockwise nearest neighbor connections are strengthened in proportion to the rotatory component of the robot movement, which is determined by odometry (as explained in Section 2.1). Consequently the hill of activity representing the goal moves along the ring of neurons counter to the direction of the robot's rotation (Fig. 10). This counter-movement of the hill continues (Fig. 9 A3 and B3) until the robot is orientated towards the goal.

Now the robot is drawn towards the goal. Its motion changes from rotation to forward motion (Fig. 9 B4). This movement is transposed into the egocentric spatial map by strengthening the inner nearest neighbor connections along the radials, in proportion to the translational component of the robot movement. At the same time the angular nearest neighbor connection strengths in the rings are set to minimal and equal values. As the robot moves towards the goal, the hill of activity moves inward along the radial, approaching

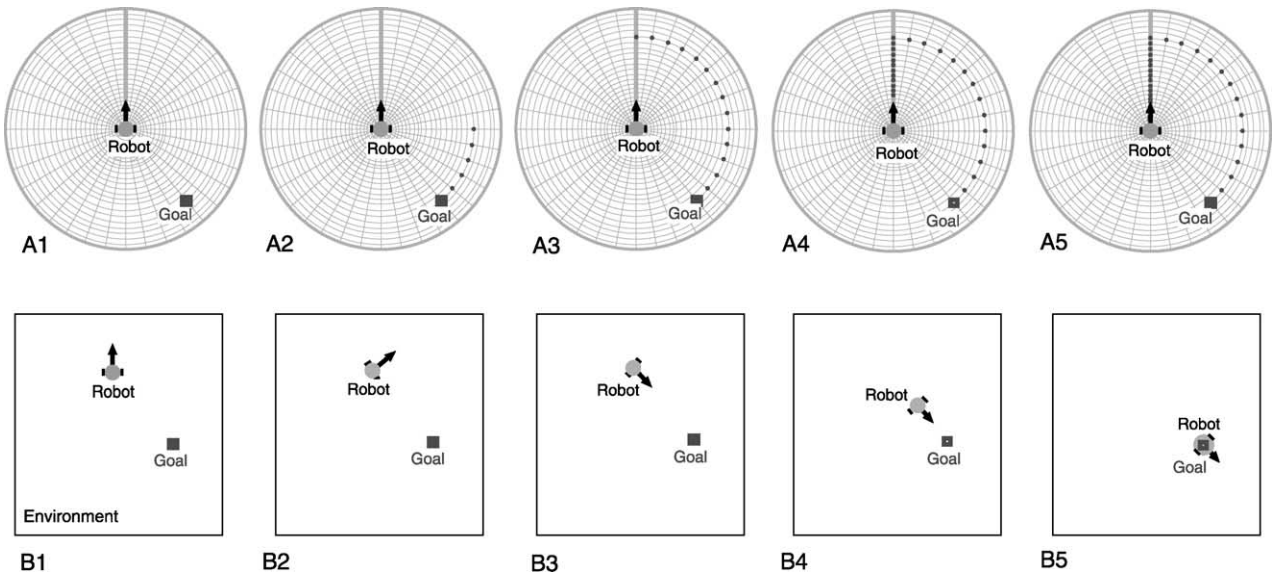

Fig. 9. The movement of the robot and the counter movement of the goal representation during the goal task is shown in a sequence of sub-figures. Sub-figures A1–A5 show schematics of the egocentric spatial representation, while sub-figures B1–B5 show corresponding top views of the environment and the movement of the robot towards the goal. Sub-figures A1 and B1 show the situation at the start. The goal is placed to the right rear of the robot. In A2 and B2 the robot begins to turn and the hill of activity representing the goal moves in counter rotation. By A3/B3 the robots rotation has brought it to face its goal, and it begins to move towards the goal (A4/B4). A5 and B5 show the situation when the robot has reached its goal, and halts. The hill of activity representing the goal has reached the innermost ring of the egocentric map.

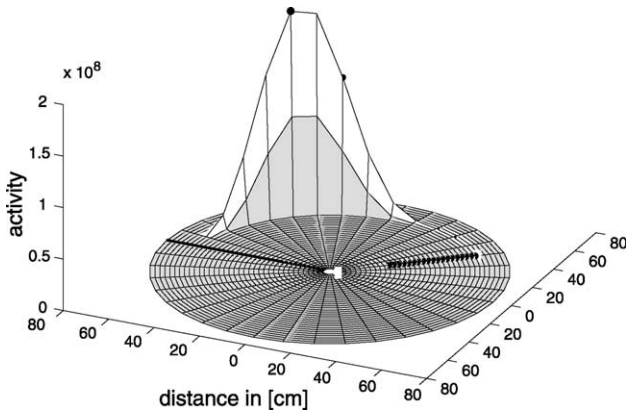

Fig. 10. The simulated activity pattern of the egocentric map with a network resolution of 36 × 36 map neurons is shown. The angular resolution is 10°, while the maximal encoded distance of the map is chosen to be $d_{max} = 74$ cm. The black line indicates the frontal axis of the robot. This activity pattern occurred in the situation shown in Fig. 9, sub-figure A2, where the robot is performing the turn towards the goal. The black filled circles indicate the trajectory of the pointer to the hill. Its present location is at the top of the hill.

the ego-center (Fig. 9 A4). A5 and B5 show the situation at the end of the run. The robot has stopped at the defined goal position, while the hill has reached the ego-center.

4.2. Performing the goal task

Now, we repeat the above experiment in the real test environment, using the *Khepera* robot.

Once again, only the angular and radial pointer neuron activities of the egocentric spatial map were used to activate the motor map. The four motor control neurons for *go forward* ($m_l = 2, m_r = 2$), *stop* ($m_l = 0, m_r = 0$), *turn right* ($m_l = 1, m_r = -1$) and *turn left* ($m_l = -1, m_r = 1$) must be activated to turn the robot towards the bearing of the goal, and then to move the robot forward until it reaches the goal and stops. The activation patterns of the motor control map that permit the robot to take the shortest route to the goal were chosen heuristically. They are:

$$C_{\text{stop}} = \alpha_{P_3} P_3 + \sum_{x=1}^{N} \sum_{y=1}^{N} M_{xy} + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 \tag{16}$$

$$C_{\text{forward}} = \alpha_{P_3} P_3 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 \tag{17}$$

$$C_{\text{turn}_{\text{left}}} = \alpha_{P_1} P_1 + \alpha_{P_4} P_4 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 \tag{18}$$

$$C_{\text{turn}_{\text{right}}} = \alpha_{P_1} P_1 + \alpha_{P_2} P_2 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 \tag{19}$$

$$\mathscr{C} = (C_{\text{stop}}, C_{\text{forward}}, C_{\text{turn}_{\text{left}}}, C_{\text{turn}_{\text{right}}}) \tag{20}$$

with the values of $\alpha_{P_3} = 0.18$, $\alpha_{P_2} = 0.355$, $\alpha_{P_4} = 0.365$ and $\alpha_{P_1} = 4.23$. The slight difference between α_{P_2} and α_{P_4} is due to mechanical differences between the two wheels.

We assessed the robot's performance in the path integration task by repeating a path to the same goal location many times (Fig. 11), and also by traversing paths to many random locations (Fig. 12). The neural network has

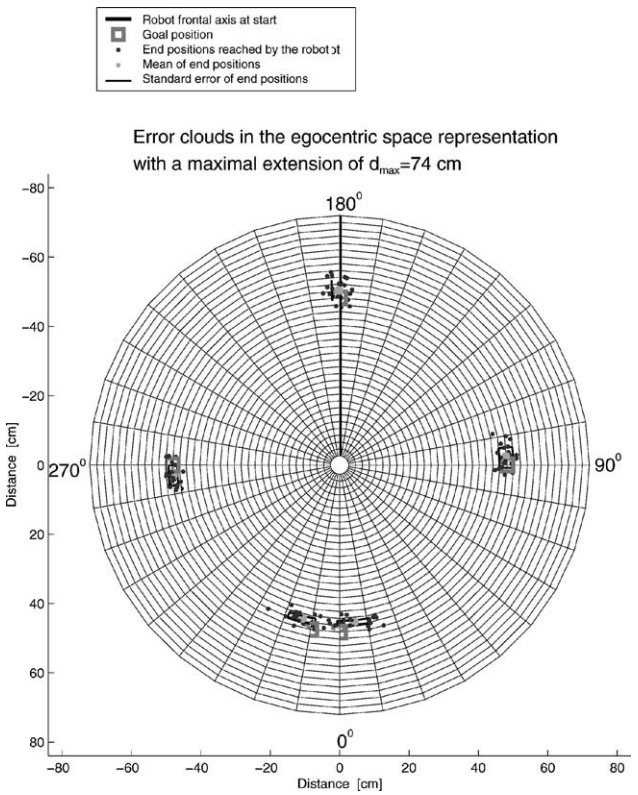

Fig. 11. The results of the goal task for five chosen goal positions are shown on a polar plot of the egocentric map. Each goal position (square) was tested twenty-five times. The goal positions were placed in front of the robot, and also to its left, right, and rear. The frontal axis of the robot is marked with a thick line. The end positions of the robot are marked with grey filled circles. The mean error around each of the five goal position is marked by a dark rectangle.

36 × 36 neurons. Each neuron encodes a specific distance and angular position. The nominal angular resolution of the network is 10°. The maximal radial extension of the network was set to 74 cm, which implies that the nominal distance represented between two concentric neurons was 2 cm. Of course, the hill of activity is distributed across a number of neurons, and so some interpolation is possible in both angular and radial dimensions.

The results for repeated path series are shown in Fig. 11. The random path series used 34 randomly chosen goals. The map distance to the goal positions varied between 3 and 34 neurons, encoding distances between 6 and 68 cm. The two inner and two outer circles of the egocentric spatial map were excluded to prevent boundary effects. The angles to different goal positions were specified to the nearest neuron (i.e. in 10° increments) in the range 0° − 350°. The results are shown in Fig. 12.

Overall, path integration by the neural network system is reasonably good, but shows systematic angular bias and undershoot (Fig. 13). It is these errors that we hoped to correct by provided visual feedback to the path integration mechanism. The correction mechanism exploits the fact that the 'where' information about the attended goal, and the location of the base of a visually detected object, are

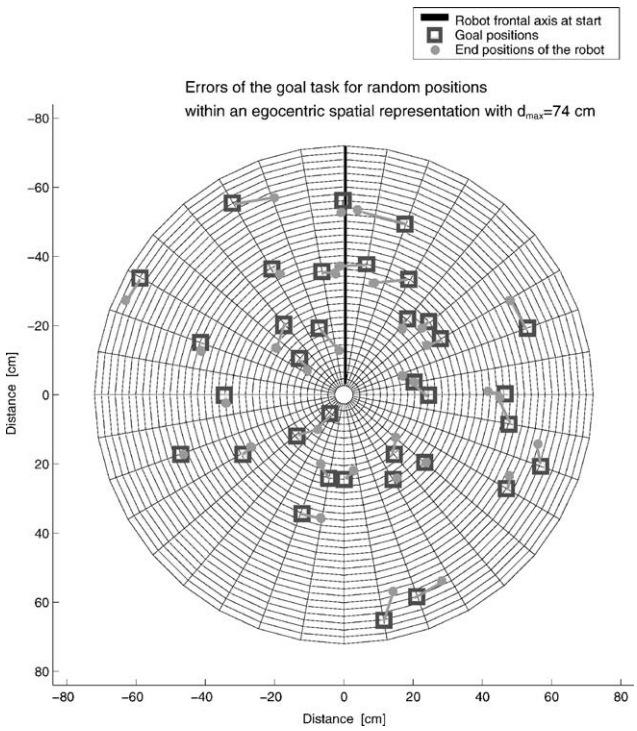

Fig. 12. The results of the goal task for randomly chosen goal positions (squares) shown on a polar plot of the egocentric spatial representation. The frontal axis of the robot is marked by a thick line. End positions are marked with a grey filled circles. The pairs of goal and end positions are linked by short lines.

both encoded as vectors. These two vectors can be matched during the task once the robot has rotated far enough that the goal falls into the camera's field of view. The angular error is the mismatch between the midline of the visual map (the direction in which the robot is moving), and the angular location of the goal encoded by the pointer neurons of the visual map.

Now, we demonstrate the interaction between the egocentric spatial representation and the visual map, while the robot repeats the same task of moving towards a goal (Fig. 14). Unlike the previous case, in which the robot moved to an unmarked location, the goal location is now marked by a white object in the empty environment that the robot can detect in its visual map.

The task for the robot is to move towards the predefined goal position on the basis of the egocentric spatial map, and then to correct its path near the goal position using the additional visual information (Fig. 14).

When the visual input to the motor map causes a path correction, the ego-motion feedback corrects the hill of activity representing the object in the spatial map. Thus, the contributions of the visual and spatial map are implicitly coupled via robot action and sensory feedback.

4.3. Goal task with visual map

In the third test series, the modular navigation system used both the visual map and the egocentric spatial map. The conditions and settings for the egocentric spatial map were the same as in the previous series. However, for this series, the activities of the motor neurons are given by the following equations, which are based on the Eqs. (6)–(10) but extended to incorporate the activities of the visual map's pointer neurons Z_f, Z_l and Z_r

$$C_{\text{stop}} = \alpha_{P_3} P_3 + \sum_{x=1}^{N} \sum_{y=1}^{N} M_{xy} + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 + \alpha_{Z_f} Z_f \quad (21)$$

$$C_{\text{forward}} = \alpha_{P_3} P_3 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 \quad (22)$$

$$C_{\text{turn}_{\text{left}}} = \alpha_{P_1} P_1 + \alpha_{P_4} P_4 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 + \alpha_{Z_l} Z_l \quad (23)$$

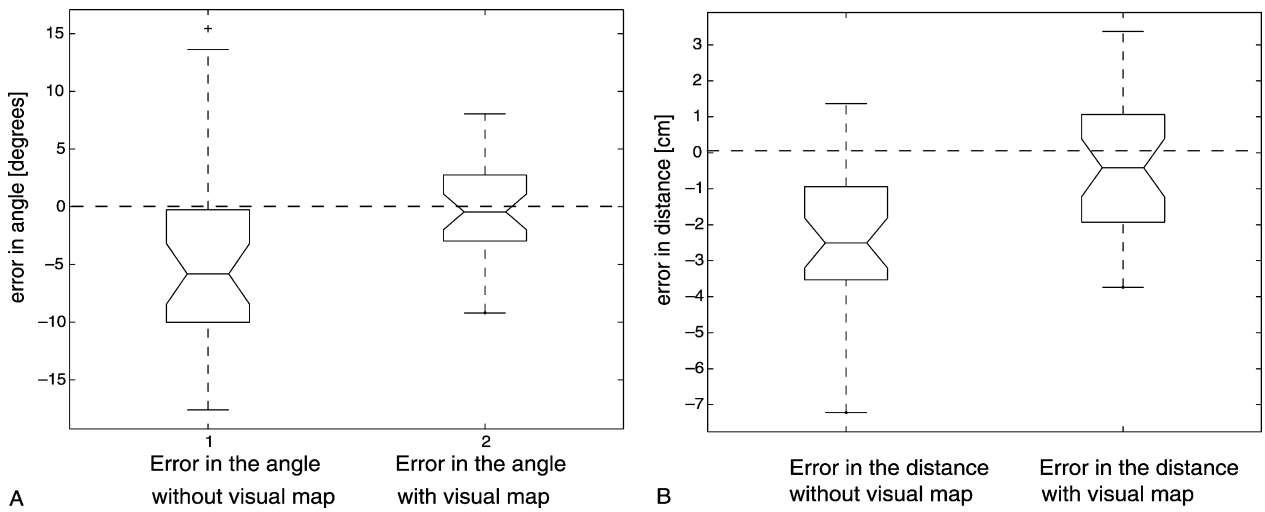

Fig. 13. (A) angle and (B) distance errors in the goal task performed with and without the assistance of the visual map. The mean value of the errors M_{error} (midline of the boxes), the 25th (first edge up from the midline of the boxes) and 75th percentiles (topline of the boxes) of the samples, the variances and the skewness are shown. Introduction of the visual map reduced angular and distance bias, and also the variance of angular errors.

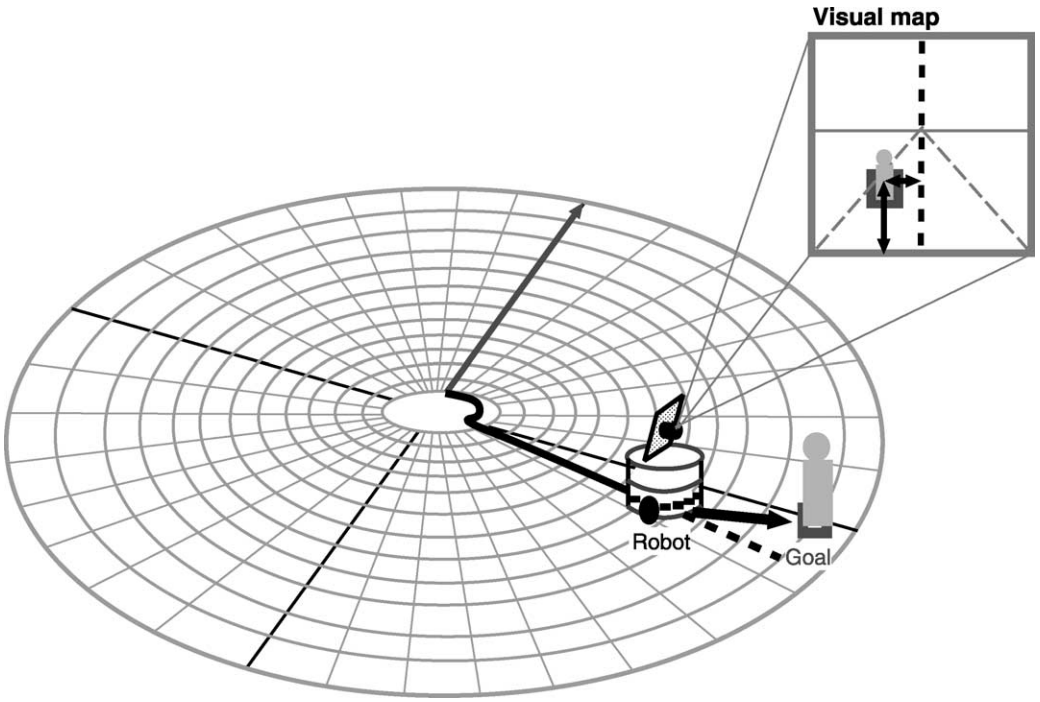

Fig. 14. The visually-mediated path correction mechanism occurs in two phases. At first, the robot takes a path towards a goal position specified within its egocentric map. Later, when the goal falls within the visual field of the on-board camera, visually induced motor correction occurs. This correction is fed back to the egocentric map by the proprioceptive-induced shift of the hill of activity within that map.

$$C_{\text{turn}_{\text{right}}} = \alpha_{P_1} P_1 + \alpha_{P_2} P_2 + \alpha_{R_1} R_1 + \alpha_{R_2} R_2 + \alpha_{Z_r} Z_r \tag{24}$$

$$\mathscr{C} = (C_{\text{stop}}, C_{\text{forward}}, C_{\text{turn}_{\text{left}}}, C_{\text{turn}_{\text{right}}}) \tag{25}$$

The activity of the visual map's frontal pointer neuron drives the motor neuron *stop*. The activity of the visual map's right pointer neuron drives the motor neuron *turn right*, while the activity of the left one drives the motor neuron *turn left*. The activities of the pointer neurons of both representations show different behaviors over time. In the first phase, the behavior of the robot is dominated by the activities of the pointer neurons of the egocentric spatial map, whereas in the second phase the pointer neurons of the visual map have the stronger influence. The activities of the pointer neurons increase as the robot moves towards the goal object.

The results are shown in Fig. 15, compared with the results of the second test series using no visual input. Visual feedback reduces the variance and bias of angular errors, and also reduces undershoot of distance (Fig. 13).

The null hypothesis that the variance in the angular errors is unaffected by visual feedback could be rejected at a significance level of $\alpha_{\text{ANOVA}} = 0.0002$, (ANOVA, $F = 16.18, p < 0.0002$). Similarly, the hypothesis that the variance in distance errors is unaffected by visual feedback could also be rejected, but at a lower level of significance: $\alpha_{\text{ANOVA}} = 0.0351$, (ANOVA, $F = 4.63, p < 0.0351$).

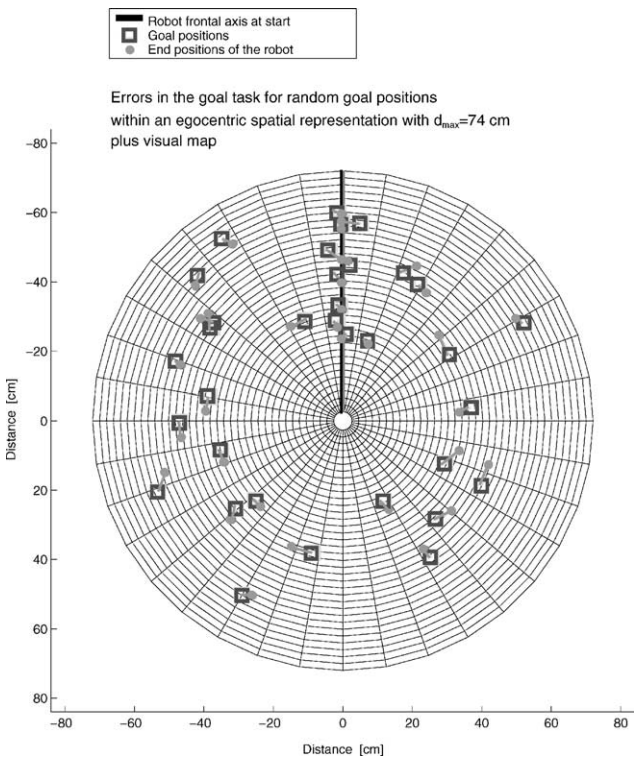

Fig. 15. The results of the goal task, now with visual assistance, for randomly chosen goal positions (squares) shown on a polar plot of the egocentric spatial map. The frontal axis of the robot is marked by a thick line. End positions are marked with a grey filled circles. The pairs of goal and end positions are linked by short lines (compare with Fig. 12).

5. Discussion

In this paper we have described a biologically inspired navigation system for a robot that is based on representations encoded in neural networks. We have explored the properties of this system as a step towards understanding how animals might manoeuvre in relation to a goal that it knows is present in its local environment, but that is not always directly detected by its sensory inputs.

The system has three separate representations. The first, a recurrent network, encodes the region surrounding the robot in form of a polar egocentric map. The second, also a recurrent network, encodes the local view detected by a forward-looking CCD camera as a rectangular map. The third representation is a one dimensional winner-take-all motor map that controls the wheel motors.

We have chosen to use a neural network foundation for our model because we are interested, finally, in how navigational abilities could be implemented in both biological networks, and neuromorphic hardware (Liu, Kramer, Indiveri, Delbrück, & Douglas, 2002). The relationship between the detail of the model, and biology on the one hand or hardware on the other, is finally a matter of taste. To make a useful linkage to biology, the model should of course respect experimental data. However, there is also a trade-off between biological fidelity and mechanisms that can be conveniently realized in silicon (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000).

It is also difficult to tread the 'biologically-inspired' path. The data are incomplete, and often not congruent with the specific task one would like to explore. Consequently, abstract models such as ours, must often make quite radical simplifications. For example, in our system, odometric feedback modifies the synaptic coupling between neurons of the egocentric map. A biological mechanism for systematically adjusting the degree of synaptic coupling between neighbors, as we do in our model, is not exactly known. However, we were more concerned to explore the general properties of the pointer-map, than to model the underlying biological detail of their synaptic interactions. And so, for convenience, we programmed those synaptic changes explicitly. This explicit code could be replaced by a more biologically plausible mechanism. For example, the left and right nearest neighbors could synapse on two symmetrical, but electrotonically separated branches of the dendritic tree of a map neuron. Then, the effect of neighbor inputs on the left or right branches of all neurons could be scaled by common 'left' and 'right' inhibitory synapses applied to the respective branches.

The question of 'biological-inspiration' also raises the general point that network based navigational systems such as the one reported here are quite different from the explicitly procedural navigation systems studied in standard computer science and engineering (Montemerlo et al., 2002). The actual biological networks that support navigation, do so by virtue of an algorithm implicit in their network dynamics. These networks are not explicitly programmed. This indirection is further increased by the need to simulate the network on a general purpose computer. Consequently, there is a large conceptual and computational overhead in implementing 'neurally inspired' navigation. So, the performance of these network models is unlikely to match standard engineering approaches until the actual biological systems are more fully understood.

Most models of place representation use an allocentric frame of reference (Arleo & Gerstner, 2000; Burgess, O'Keefe, & Reece, 1993; Trullier & Meyer, 2000). The current position of the animal is encoded by a hill of activity, as observed in the CA3 cells of the rodent hippocampus (Muller, Kubie, Bostock, Taube, & Quirk, 1987; Muller et al., 1991). In these models the path towards a specific location is learned by changing the pattern of connectivity between the place cells. The method for constructing this map is to use random motoric behavior to elicit perceptions that are correlated with place. This strategy has the disadvantage that the (model) animal is unable to move between specific locations before the map construction phase is complete. By contrast, we require that the navigator act competently from the moment that it enters a novel environment. This requirement suggests the use of an egocentric map that exploits the egocentric viewpoint of perception, provides an immediate relationship between the ego and its perceived goal, and supports path integration between present location and that goal. Indeed, Mittelstaedt (2000) claims that the path integration system is a purely egocentric system.

Furthermore, there is biological evidence that the path integration mechanism is physically separate from the allocentric place code representation. It is possible that both representations reside within the hippocampus (Redish, 1999; Redish & Touretzky, 1997; Samsonovich & McNaughton, 1997; Sharp, 1997). But, Alyan (Alyan & McNaughton, 1999; Alyan et al., 2000) has shown that a rodent can perform path integration with a disabled hippocampus, suggesting that the mechanism lies beyond those limits.

We have explored a path integration mechanism that guides an agent towards a goal. The mechanism operates with an egocentric map, and is supported by a visual map, both of which have pointer-map (Hahnloser et al., 1999) architectures. This architecture combines a place encoding in feedback with a vectorial encoding. The location of the goal is represented by a hill of neuronal activity on the egocentric map, and the direction to the goal with respect to the agent's location is reported by the pointer (or vector). The hill of activity moves counter to the movements of the robot or animal, and so provides effective path integration.

Such hills of activity have been proposed for movement fields in the monkey superior colliculus by Sparks (Munoz & Wurtz, 1995; Munoz, Pelisson, & Gitton, 1991; Sparks,

Holland, & Guthrie, 1976). The concept of a shifting hill of activity has also been used in models for the head direction system (Blair, 1996; Skaggs, Knierim, Kudrimoti,& McNaughton, 1995; Zhang, 1996) to represent the current head direction of the rodent, or in models of place coding (Samsonovich & McNaughton, 1997) to encode the current position of the rodent in the environment.

Zhang (1996) has demonstrated a head direction network that integrates direction changes by modifying the strength of connections between neighbors in a one dimensional map. He used symmetric and asymmetric weight settings to hold, or shift a hill of neuronal activity. Zhang (1996) showed that these principles could be extended also to two dimensions, and suggested that this updating mechanism could be used for path integration in the context of a map of place cells. We have used Zhang's mechanism for controlling the location of activity, but we organized the map in a polar grid, similar to the toroidal network discussed by Redish (1999). This two-dimensional ring attractor network provides the egocentric map, and is the substrate for path integration.

Our integration mechanism is different from the one used in the ant path integration model proposed by Hartmann (Hartmann & Wehner, 1995). Rather than representing location on a single egocentric map, their model uses separate chains of neurons each of which integrates a relevant path variable, such as displacement and angle of locomotion.

Our approach goes further than simply implementing 'taxon' navigation (Redish, 1999), in which an animal approaches its goal on the basis of a direct stimulus-response schema (Arkin, 1998). In our system, the goal location is mapped into a predefined egocentric spatial map, which is also a reference frame for potential movements of the robot. The vector output, given by the pointer of the egocentric spatial map encodes 'where' to go, in egocentric bearing and distance, and guides the robot towards its goal. This taxic behavior can occur even when the attended object is not present in the visual image. That is, the robot can manoeuvre in relation to an imagined goal.

The egocentric spatial map allows the robot to track its own movement relative to the goal by using counter-movement of the goal as represented in the egocentric map. This property implies a self-localization with respect to an attended external cue or landmark. Such self-localization is more sophisticated than a simple stimulus-response schema.

Self-localization does not require a complete allocentric representation of the environment, as found in a place code. It is enough to know the vectorial relationship between the ego and the attended location. While the robot is moving towards its goal, this vector can be combined with visual information to correct the path as described above. The correction of the path due to the visual input causes correction movements towards the attended object, which are then mirrored in the changes of the shift in the hill of activity due to the ego-motion feedback. In this way, the path-integration mechanism is stabilized in relation to the attended object. It is unknown whether animals perform path integration only in relation to external cues perceived in the local view, or if they rely also on self-motion feedback (Whishaw & Brooks, 1999).

The dynamic update of the goal depends on proprioceptive feedback obtained from the wheel encoders, which are subject to inaccuracies (e.g. slip errors); and so path integration may accumulate errors (Mittelstaedt, 2000). Our system minimized these errors by a method of self-correction based on vector matching. The method makes use of the local view representation in the visual map. The pointer of the egocentric map, which points to the represented goal, is matched against the pointer of the visual map, which points to actual goal in view. Their difference is used as a motor correction.

We have shown, both in simulation and in real robots, that the combination of egocentric map and motor map alone was able to support egocentric navigation towards an imagined goal. However, the real robot had quite large navigation errors due to imperfections in odometry feedback to the egocentric map. These errors could be reduced in the real robot by using visual feedback. Interaction between the egocentric and visual maps is mediated via their respective pointers (vectors). Thus, when the goal finally falls within the scope of the on-board CCD camera, the visualized goal location can be used to correct navigational errors.

It is not surprising that the errors in path integration could be reduced by using the spatial information contained in the visual map. What is interesting is the simplicity of the mechanism used for achieving this correction: the combined effect of two pointers on the motor map.

The visual correction of angular error was more successful than the correction of distance error. The poorer distance correction may be due to the fact that the on-board camera was mounted in such a way that there is a blind region of 3 cm immediately in front of the robot. The activities of the visual map pointer neurons (which cover only the visual image) do not properly reflect the distance to the attended object when it falls in this blind region. Moreover, the visual map does not cover the entire distance between ego and object, and the visual preprocessing is not sufficiently sophisticated to decide whether the entire base of the attended object is visible or only its vertical edge. The integration errors could possibly be reduced by addressing these short-comings. For example, the modular navigation system could be extended to incorporate another external representation of the near field, based on active infrared sensors. These infrared sensors could measure the distance to objects in the visual blind region. The self-correction abilities of the robot could also be improved by additional external sensor representations, such as audition in the presence of an audible goal object.

Vector encoding of 'where' information offers a useful method of combining sensory information from quite

different representations. For example, our visual map represents the egocentric view under a specific angle of the robot, which is only a segment within the egocentric spatial map covering 360° of the near environment. Nevertheless, they can be linked on the basis of their vectorial outputs, which point to the same attended object. Because the vectors represent the same 'where' information they can be combined without regard for the nature or spatial extension of the internal representation that they encode, vector combination is a promising approach to the fusion of representations of environmental cues (e.g. vision, audition, and olfaction), and internal motor feedback information (e.g. velocity, or efference copy of motor commands).

Although we have used visual information to correct progress towards an already attended goal, the process could be inverted. Processing of visual input entails detection and localization of a salient feature or object. The 'where' of the visual map is reported by a pair of pointers coding for azimuth and distance away from the robot. This information could, in principle, be used to initialize the encoding of that object in the egocentric map, and the pointers of the spatial map could be used to guide the robot towards the attended object. Used in this mode, our navigation system would be combining 'what' and 'where' information (Ungerleider & Mishkin, 1982) about an object in the execution of its goal behavior, in the manner supposed for visuomotor control in general (Goodale & Milner, 1992; Goodale, 1998).

Acknowledgements

This work was supported by the Swiss National Science Foundation SPP Grant and the Körber Foundation. We also thank A. M. Whatley for technical support during the development of the test environment and for proof-reading the manuscript.

References

Alyan, S. H., Jander, R., & Best, P. J. (2000). Hippocampectomized rats can use a constellation of landmarks to recognize a place. *Brain Research*, *876*, 225–237.

Alyan, S. H., & McNaughton, B. L. (1999). Hippocampectomized rats are capable of homing by path integration. *Behavioral Neuroscience*, *113*(1), 19–31.

Arkin, C. A. (1998). *Behavior-based robotics*. Cambridge, MA: MIT Press.

Arleo, A., & Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biological Cybernetics*, *83*, 287–299.

Birchfield, S (1998). KLT: An implementation of the Kanade–Lucas–Tomasi feature tracker, Technical report, Stanford, http://vision.stanford.edu/~birch/klt/

Blair, H. T. (1996). A thalamocortical circuit for computing directional heading in the rat. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 7, (NIPS)* (pp. 152–158). Cambridge, MA: MIT Press.

Bruce, J., Veloso, M.M (2002). Real-time randomized path planning for robot navigation. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS' 02)*

Burgess, N., Donnett, J. G., Jeffery, K. J., & O'Keefe, J. (1997). Robotic and neuronal simulation of the hippocampus and rat navigation. *Philosophical Transactions of the Royal Society London B*, *352*, 1535–1543.

Burgess, N., & O'Keefe, J. (1996). Neuronal computations underlying the firing of place cells and their role in navigation. *Hippocampus*, *Vol. 6*, 749–762.

Burgess, N., O'Keefe, J., & Reece, M. (1993). Using hippocampal place cells for navigation, exploiting phase coding. In S. J. Hanson, C. L. Giles, & J. D. Cowan (Eds.), (*5*) (pp. 929–936). *Advances in Neural Information Processing Systems 5, (NIPS)*, Los Altos, CA: Morgan Kaufmann.

Choset, H. (2001). Topological simultaneous localization and mapping SLAM: toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, *17*(2), 125–137.

Fox, D., Burgard, W., & Thrun, S. (1998). Active Markov localization for mobile robots. *IEEE Robotics and Autonomous Systems*, *25*, 195–207.

Gallistel, C. R. (1990). *The organisation of learning*. Cambridge, MA: MIT Press.

Goodale, M. A. (1998). Visuomotor control: where does vision end and action begin? *Current Biology*, *8*, 489–491.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, *15*, 20–25.

Guazzelli, A., Bota, M., & Arbib, M. A. (2001). Competitve Hebbian learning and the hippocampal place cell system: modeling the interaction of visual and path integration cues. *Hippocampus*, *11*(3), 216–239.

Hahnloser, R., Douglas, R. J., Mahowald, M., & Hepp, K. (1999). Feedback interactions between neuronal pointers and maps for attentional processing. *Nature Neuroscience*, *2*, 746–752.

Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, *405*, 947–951.

Hartmann, G., & Wehner, R. (1995). The ant's path integration system: a neural architecture. *Biological Cybernetics*, *73*, 483–497.

K-Team, S. A. (1995). *Khepera: User manual, v4.06. Technical report, EPFL, Lausanne*.

Lewis, J. E., & Kristan, W. B. J. (1998a). A neuronal network for computing population vectors in the leech. *Nature*, *391*, 76–79.

Lewis, J. E., & Kristan, W. B. J. (1998b). Quantitative analysis of a directed behavior in the medicinal leech: Implications for organizing motor output. *The Journal of Neuroscience*, *18*(4), 1571–1582.

Liu, S.-C., Kramer, J., Indiveri, G., Delbrück, T., & Douglas, R. J. (2002). *Analog VLSI: circuits and principles*. Cambridge, MA: MIT Press.

McNaughton, B. L., Barnes, C. A., Gerrard, J. L., Gothard, K., Jung, M. W., Knierim, J. J., Kudrimoti, H., Qin, Y., Skaggs, W. E., Suster, M., & Weaver, K. L. (1996). Deciphering the hippocampal polyglot: the hippocampus as a path integration system. *The Journal of Experimental Biology*, *199*, 173–185.

Mittelstaedt, H. (2000). Triple-loop model of path control by head direction and place cells. *Biological Cybernetics*, *83*, 261–270.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). *Fast SLAM: a factored solution to the simultaneous localization and mapping problem. In Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada: AAAI.

Mudra, R (2002). *A robot using a modular navigation system: attentional manoeuvring based on an egocentric spatial representation*, PhD Thesis, ETH, Zürich, Switzerland.

Mudra, R., Hahnloser, R., & Douglas, R. J. (1999). Neuromorphic active vision used in simple navigation behavior for a robot. In B. Werner (Ed.), *Proceedings of the Seventh International Conference on*

Microelectronics for Neural, Fuzzy and Bio-Inspired Systems (Micro-Neuro'99). Los Alamitos, CA: IEEE Computer Society.

Muller, R. U., Kubie, J. L., Bostock, E. M., Taube, J. S., & Quirk, G. J. (1991). Spatial firing correlates of neurons in the hippocampal formation of freely moving rats. In J. Paillard (Ed.), *Brain and Space* (pp. 296–333). Oxford: Oxford University Press.

Muller, R. U., Kubie, J. L., & Ranck Jr, J. B. (1987). Spatial firing patterns of hippocampal complex-spike cells in a fixed environment. *The Journal of Neuroscience*, *7*(7), 1935–1950.

Munoz, D. P., Pelisson, D., & Gitton, D. (1991). Movement of neural activity on the superior colliculus motor map during gaze shifts. *Science*, *251*, 1358–1360.

Munoz, D. P., & Wurtz, R. H. (1995). Saccade-related activity in monkey superior colliculus. 1. Characteristics of burst and buildup cells. *Journal of Neurophysiology*, *73*, 2313–2333.

Pfeifer, R., & Christian, S. (1999). *Understanding Intelligence*. Cambridge, MA: MIT Press.

Redish, A. D. (1999). *Beyond the cognitive map: from place cells to episodic memory*. Cambridge, MA: MIT Press.

Redish, A. D., & Touretzky, D. S. (1997). Cognitive maps beyond the hippocampus. *Hippocampus*, *7*(1), 15–35.

Redish, A. D., & Touretzky, D. S. (1998 martin.rosentritt@klinik.uni-regensburg.de (M. Rosentritt).). The role of the hippocampus in solving the Morris water maze. *Neural Computation*, *10*, 73–111.

Redish, A. D., & Touretzky, D. S. (1998b). Separating hippocampal maps. In N. Burgess, K. Jeffrey, & J. O'Keefe (Eds.), *Spatial functions of the hippocampal formation and the parietal cortex* (pp. 203–219). Oxford: Oxford University Press.

Samsonovich, A., & McNaughton, (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, *17*(15), 5900–5920.

Sharp, P. E. (1997). Subicular cells generate similar firing patterns in two geometrically and visually distinctive environments: comparison with hippocampal place cells. *Behavioural Brain Research*, *85*(1), 71–92.

Skaggs, W. E., Knierim, J. J., Kudrimoti, H. S., & McNaughton, B. L. (1995). A model of the neural basis of the rat's sense of direction. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in Neural Information Processing Systems 7, (NIPS)* (pp. 173–182). Cambridge, MA: MIT Press.

Sparks, D. L., Holland, R., & Guthrie, B. L. (1976). Size and distribution of movement fields in the monkey superior colliculus. *Brain Research*, *113*, 21–34.

Trullier, O., & Meyer, J.-A. (2000). Animate navigation using a cognitive graph. *Biological Cybernetics*, *83*, 271–285.

Ungerleider, L. G., & Mishkin, M. (1982). Two cortical visual systems. In D. J. Ingle, M. A. Goodale, & R. J. W. Mansfield (Eds.), *Analysis of visual behavior* (pp. 549–586). Cambridge, MA: MIT Press, (chapter 18).

Whishaw, I. Q., & Brooks, B. L. (1999). Calibrating space: exploration is important for allothetic and idiothetic navigation. *Hippocampus*, *9*, 659–667.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *The Journal of Neuroscience*, *1*(6), 2112–2126.