Cognitive Navigation Based on Nonuniform Gabor Space Sampling, Unsupervised Growing Networks, and Reinforcement Learning

Angelo Arleo, Fabrizio Smeraldi, and Wulfram Gerstner

***Abstract*—We study spatial learning and navigation for autonomous agents. A state space representation is constructed by unsupervised Hebbian learning during exploration. As a result of learning, a representation of the continuous two-dimensional (2-D) manifold in the high-dimensional input space is found. The representation consists of a population of localized overlapping place fields covering the 2-D space densely and uniformly. This space coding is comparable to the representation provided by hippocampal place cells in rats. Place fields are learned by extracting spatio-temporal properties of the environment from sensory inputs. The visual scene is modeled using the responses of modified Gabor filters placed at the nodes of a sparse Log-polar graph. Visual sensory aliasing is eliminated by taking into account self-motion signals via path integration. This solves the hidden state problem and provides a suitable representation for applying reinforcement learning in continuous space for action selection. A temporal-difference prediction scheme is used to learn sensorimotor mappings to perform goal-oriented navigation. Population vector coding is employed to interpret ensemble neural activity. The model is validated on a mobile Khepera miniature robot.**

***Index Terms*—Gabor decomposition, Hebbian learning, hippocampal place cells, log-polar sampling, population vector coding, reinforcement learning, robot navigation, spatial memory, unsupervised learning.**

I. Introduction

To solve complex spatial tasks, both animals and robots must interact with their environments to construct space representations which support goal-oriented behaviors. Neurophysiological findings suggest that spatial learning in rodents is supported by place- and direction-sensitive cells. Hippocampal place (HP) cells in rats provide a spatial representation in allocentric (world centered) coordinates [1]. HP cells discharge action potentials only when the animal is in a specific region of the environment, which defines the place field of the cell. Complementing this, head direction (HD) cells encode the rat's allocentric heading in the azimuthal plane [2]. A HD cell fires maximally only when the rat's head is oriented to a specific direction, regardless of the orientation of the head relative to the body, of the rat's location, or of the animal's ongoing behavior.

This paper addresses two main questions.

i) How can robots establish appropriate place representations based on locally available sensory information? We propose a neural architecture in which multimodal sensory signals are used to achieve space coding. We stress the importance of combining allothetic (e.g., visual) and idiothetic (e.g., self-motion) signals to learn stable representations (Fig. 1). Place fields based on vision are determined by a combination of environmental cues whose mutual relationships code for the current agent's location. However, when dependent on visual data alone, the representation encoded by place cells does not fulfill the Markov hypothesis (in the framework of Markov Decision Processes, the first-order Markov hypothesis assumes that the future state of the system only depends on its present state, and not on its past history) [3]. Indeed, distinct areas of the environment may provide equivalent visual patterns (perceptual aliasing) and lead to singularities in the vision-based representation. We employ idiothetic signals (i.e., path integration) along with vision in order to remove such singularities and solve the hidden-state problem. Conversely, visual cues are used to prevent the path integrator (i.e., the odometer) from cumulative error over time. This closed sensory loop results in stable space coding. During the agent-environment interaction, correlations between visually driven cells and path-integration are discovered by means of unsupervised Hebbian learning. Thus, allothetic and idiothetic space codings converge to create a robust multimodal space representation consisting of a large number of localized overlapping place fields. The rationale behind such a redundant space code is two-fold; first, to cover the space uniformly to generate a continuous coarse coding representation (similar to a dense family of overlapping basis functions); and second, to use the place cell population activity, rather than the single cell activity, for self-localization. We apply population vector coding [4] to map the ensemble place cell activity into spatial locations.

ii) How can place cells serve as a basis for goal-oriented navigation? To accomplish its functional role in spatial behavior, the model must incorporate the information about relationships between the environment, its obstacles and specific target locations. Place units drive a downstream population of locomotor action units (Fig. 1). Action learning relies on the reward-dependent

Manuscript received April 29, 2002; revised September 15, 2003. This work was supported by the Swiss National Science Foundation Project 21-49 174.96.

A. Arleo is with the Neuroscience Group, SONY Computer Science Laboratory, 75005 Paris, France (e-mail: angelo.arleo@csi.sony.fr).

F. Smeraldi is with the Department of Computer Science, Queen Mary, University of London, E1 4NS London, U.K.

W. Gerstner is with the Laboratory of Computational Neuroscience, EPFL, CH-1015 Lausanne, Switzerland.

Digital Object Identifier 10.1109/TNN.2004.826221

1045-9227/04$20.00 © 2004 IEEE

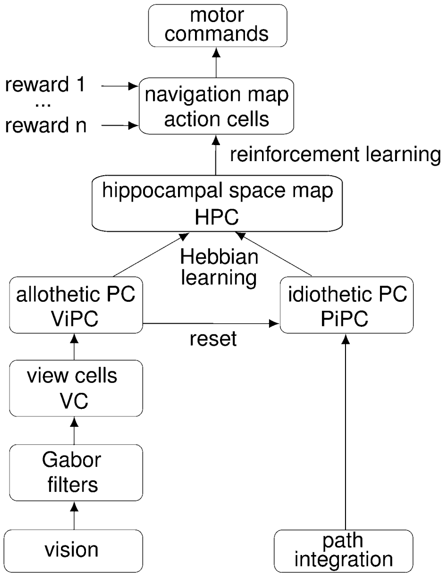

Fig. 1. Overview of the system. The visual pathway of the model (left) includes a set of Gabor filters for image processing, a set of view cells (VC) to encode views, and a network of vision-based place cells ($ViPC$). The idiothetic pathway (right) includes the path integrator and a set of units ($PiPC$) providing space coding based on self-motion stimuli. $ViPC$ and $PiPC$ converge onto the hippocampal place cell layer (HPC) where the final space representation is formed. The HPC network drives action units to support goal-directed behavior and navigation is achieved by mapping place cell activity into actions based on reinforcement learning.

modification of the connections from place to action units. This results in an ensemble pattern of activity of the action units that provides a goal-oriented navigation map. It is shown that the spatial learning system provides an incrementally learned representation suitable for applying reinforcement learning in continuous state spaces [5]. A direct implementation of reinforcement learning on real visual streams would be impossible given the high dimensionality of the visual input space. A place field representation extracts the low-dimensional view manifold on which efficient reinforcement learning is possible. The overlapping place fields provide a "natural" set of basis functions in the sense we do not have to choose parameters like width and location of the basis functions (rather, they are established automatically by unsupervised learning). In particular, basis functions in different regions of the environment may, in general, have different widths. This family of basis functions is used to learn a parameterized form of the action-value function to be optimized by reinforcement learning. Learning an action-value function over a continuous state space endows the system with spatial generalization capabilities.

A. Related Work

This paper does not address the issue of establishing a representation of the allocentrically referred direction of the robot. We have proposed a computational model for HD cells (which has been validated on the same robotic platform used in this paper) in [6], [7]. In the present study, the robot uses the output of that HD system to maintain an estimate $\phi(t)$ of its absolute direction at time t. In contrast to our previous work [8], the visual input in the current study is no longer provided by a linear sensory array but by a two-dimensional (2-D) CCD camera with 768×576 pixels. The Gabor-based decomposition technique as well as the retinotopic image sampling employed here to process visual information (Section II-C.1) have been partially introduced in the earlier work [9]. Nevertheless, the present paper provides a more comprehensive and detailed description of those techniques and reports unpublished experimental results concerning different levels of the whole spatial learning system.

The issue of building internal models of the world suitable for autonomous navigation has been largely investigated in robotics. Map-learning research has produced two principal paradigms, namely metric [10] and topological [11]. In the former, the geometrical features of the world are encoded accurately. An example of metric approach consists of representing space by means of a 2-D evenly spaced grid called the occupancy grid [10]. Each grid cell estimates the occupancy probability for the corresponding location of the environment. Topological maps are more compact representations in which spatial relations between relevant locations in the environment are modeled by means of a topological graph [11]. Only neighborhood relations are encoded while metric information is lost.

Metric representations are prone to errors concerning sensory metric information (e.g., estimates of distance to obstacles). Also, reproducing the geometric structure of the environment explicitly may be expensive in terms of memory and time. For instance, an occupancy grid that models a complex environment accurately must have a high resolution which requires heavy computation. Since topological maps are qualitative representations, they are less vulnerable to errors in metric signals and tend to optimize memory and time resources. However, since pure topological graphs rely upon a sensory pattern recognition process, perceptual aliasing phenomena may impair self localization. As metric and topological paradigms exhibit complementary strengths and weaknesses, several models have been proposed to integrate both representations into a hybrid map-learning system [12]–[14].

Another approach to space coding takes inspiration from biological solutions, which offers the attractive prospect of developing autonomous systems that directly emulate animals' navigation abilities. Burgess *et al.* [15] put forth a hippocampal model in which metric information (e.g., distance to identified visual cues) is explicitly used as input to the system. By contrast, the present model interprets visual properties by a population of neurons sensitive to specific visual patterns (however, no explicit object recognition takes place). Mallot *et al.* [16] build a topological representation of the environment consisting of a sparse view graph. Our representation is redundant and uses a large number of overlapping place fields. Trullier and Meyer [17] model the environment by a hetero-associative network encoding temporal sequences of local views. In our approach, temporal aspects are implicitly encoded by the idiothetic representation based on inertial signals. Our approach also contrasts with the model by Gaussier *et al.* [18] in which hippocampal cells do not code for places but for transitions between states. An important difference between our approach and the previous four models is that we combine allothetic (visual) information and idiothetic (path integration) signals at the level of the hippocampal representation. By contrast, the above approaches build space representations mainly on the basis of visual cues. Balakrishnan *et al.* [19] also take into account self-motion inputs for space coding. However, they use a Kalman filter technique to combine vision and path integration, whereas we employ unsupervised Hebbian learning to correlate multimodal inputs.

II. Methods

A. The Experimental Setup

The model is validated by means of a mobile Khepera miniature robot.[1] The Khepera has a cylindrical shape with a diameter of 55 mm, and, in the configuration used for the experiments, is about 90 mm tall. Two dc motors drive two wheels independently providing the robot with nonholonomic motion capabilities. The robot's sensory capabilities consist of: i) Eight infrared sensors detecting obstacles within a distance of about 40 mm; six of the infrared sensors span the frontal 180° of the robot, whereas the remaining two sensors cover approximately 100° of the posterior side; ii) a 2-D vision system whose view field covers about 90° in the horizontal plane and 60° in the vertical plane. Its image resolution is 768 × 576 pixels; and iii) an odometer to compute both linear and angular displacements based on wheel turns (i.e., dead-reckoning). Signals provided by the infrared sensors and the visual system form the allothetic inputs to the robot, whereas the self-movement signals provided by the odometer constitute the idiothetic inputs.

In this work, we develop a high-level controller determining the robot's behavior based on its own experience. In addition, a hard-coded reactive module allows the robot to avoid obstacles. Whenever the proximity sensors detect an obstacle, the low-level module takes control and prevents collisions. Both high- and low-level controllers discretize the robot's locomotion in unit-length steps $\Delta s(t)$ oriented to the current robot's allocentric direction $\phi(t)$. As described in Section II-C.1, at each visited location $\vec{s}(t)$ the robot takes four snapshots v_i, with $v_{i+1} = v_i + 90°$, to form a quasi-panoramic view. Then, it updates its motion direction $\phi(t)$ by an angle $\Delta\phi(t) \in [-180,180]°$, and moves forward by $\Delta s(t) = 50\ \mathrm{mm}$. We define a macro time step as the time necessary to the robot to acquire a quasi-panoramic view, modify its orientation, and locomote one step further. With the robotic setup (e.g., translational and angular velocity profiles) used in this work a macro time step lasts 8 s.

The experimental environment consists of a 800 × 800 mm open-field arena. Low barriers (20 mm) prevent the robot from running outside the arena. Obstacles are placed within the environment depending on the experimental protocol. The arena is placed within a standard laboratory background and the robot's behavior is monitored by means of a video camera above the arena for performance assessment.

B. Single Neuron Model

The elementary components of the system are computational units i with continuous-valued responses $r_i \in [0, 1]$. The output r_i is the mean firing rate of i (i.e., the average number of spikes emitted by i within a time window Δt) and is given by

$$r_i(t) = f\big(V_i(t) \cdot (1 \pm \epsilon)\big) \tag{1}$$

where $V_i(t)$ is the unit's membrane potential at time t, f is the transfer function determining the response of i, and ϵ is random noise uniformly drawn from $[0, 0.1]$. We employ both nonlinear and linear transfer functions f [(8) and (11), respectively]. The system dynamics is determined by the following

$$\tau_i \cdot \frac{dV_i(t)}{dt} = -V_i(t) + I_i(t) \tag{2}$$

where $\tau_i = 10\ \mathrm{ms}$ is the membrane time constant and $I_i(t)$ denotes the synaptic input to unit i from other neurons j at time t. We take

$$I_i(t) = g\big(\{r_j(t)\}, w_{ij}, \rho\big) \tag{3}$$

where $\{r_j(t)\}$ is the set of firing rates of units j, w_{ij} is the weight of the projection from j to i, and ρ is the probability for the synaptic transmission from j to i to occur at time t. The function g determines the contribution of each afferent unit j to the input received by i at time t. Excitatory inputs $I_i(t)$ tend to depolarize the cell i and the firing rate $r_i(t)$ is proportional to the amount of depolarization (1). Equation (2) is integrated by employing a time step $\Delta t = 1\ \mathrm{ms}$

$$V_i(t + \Delta t) = V_i(t) + \frac{\Delta t}{\tau_i} \cdot \big(-V_i(t) + I_i(t)\big). \tag{4}$$

C. Spatial Learning: Building a State-Space Representation

This section focuses on the first issue addressed by this work, that is it describes how a place field representation can be established based on locally available sensory inputs.

1) Allothetic Space Coding: For vision-based space coding, relevant information must be extracted from noisy visual inputs. Moving up the anatomical visual pathway (from the retina to the lateral geniculate nucleus and then toward higher visual cortical areas), neurons become responsive to stimuli of increasing complexity, from orientation-sensitive cells (simple cells) to neurons responding to more complicated patterns, such as faces [20]. The model extracts low-level visual features by computing a vector of Gabor filter responses at the nodes of a sparse Log-polar (retinotopic) graph. Then, more complex spatio-temporal relationships between visual cues are encoded by the activity of units that respond to combinations of specific visual patterns. The problem consists of detecting an appropriate low-dimensional representation of the continuous high-dimensional input space, and learning the mapping from the visual sensory space to points within this representation. Since the robot moves within a 2-D space with a camera pointing in the motion direction, the high-dimensional visual space is not uniformly filled. Rather, all input data points lie on a low-dimensional surface embedded in a Euclidean space whose dimensionality is given by the total number of camera pixels. This low-dimensional description of the visual space is referred to as the view manifold [21].

[1] The Khepera is a commercial platform produced and distributed by K-Team S.A. (http://www.k-team.com/).

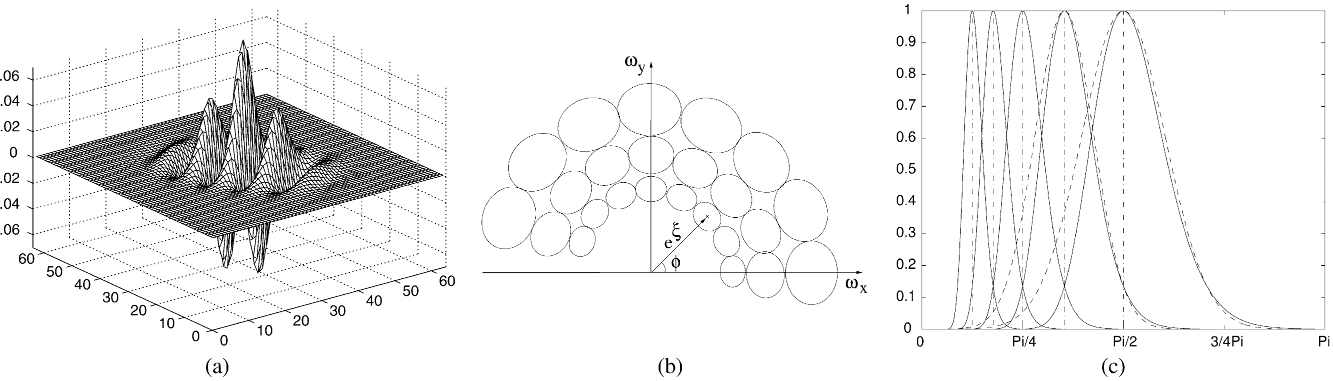

Fig. 2. (a) Real part of the complex sinusoidal wave (within a 2-D Gaussian envelope) representing a Gabor filter in the image domain. (b) Level curves of a set of modified Gabor filters in the frequency plane. A cross-sectional plot of the normalized magnitude of the filters along the ω_x axis (c) evidences the steep cutoff on the low-frequency side, which reduces the overlap between filters. Two standard Gabor filters (dashed lines) are shown for comparison.

The allothetic space representation is the result of a three-step processing (left side of Fig. 1): i) a set of Gabor filters detects visual features which then map real images onto the filter-activity space; ii) a population of view cells (VC), one synapse downstream from the filter layer, encodes the current visual input by neural activity; and iii) unsupervised Hebbian learning constructs vision-based space coding and a population of place units $(ViPC$, vision driven place cells) is built one synapse downstream from the VC layer.

a) Modified gabor filters and retinotopic image sampling for visual feature extraction: Gabor filters are frequency and orientation selective filters that provide a suitable mathematical model for the so-called simple cells in the visual cortex [22]. A set of standard Gabor filters can be constructed by scaling and rotating the Gabor function $\mathcal{G}$, represented by a 2-D Gaussian modulated by a complex sinusoidal wave [Fig. 2(a)]. The Fourier domain representation $\tilde{\mathcal{G}}$ of a Gabor filter consists of a Gaussian centered at the polar frequency coordinates $(\|\vec{\omega}_0\|, \phi_0)$, with $\phi_0 = \arctan(\omega_y/\omega_x)$. Thu,s $\|\vec{\omega}_0\|$ and ϕ_0 represent the frequency and orientation values at which the filter is tuned, that is those that will elicit the strongest response. A set of filters tuned to the same absolute frequency $\|\vec{\omega}_0\|$ but with varying preferential orientations is normally referred to as a frequency channel.

Gabor filters are optimal in that their Gaussian envelope in both the frequency and the image domains maximizes the joint spatial (locality) and frequency selectivity. However, using only a small number of logarithmically spaced frequency channels may result in the low-frequency region of the Fourier plane being oversampled, while the high frequency regions are poorly covered. This is due to the fact that the Gaussian spectra weight the high and low frequencies in their support symmetrically, whereas the spacing between filters increases with frequency. To compensate for this, we use a set of *modified* Gabor filters defined as Gaussians in the Log-polar frequency plane [23]. Let $\xi = \ln \|\vec{\omega}\|$ be the logarithmic frequency axis; we set

$$\tilde{\mathcal{G}}'(\xi, \phi) = A \cdot e^{-(\xi-\xi_0)^2/2\sigma_\xi^2} \cdot e^{-(\phi-\phi_0)^2/2\sigma_\phi^2} \tag{5}$$

where A is a normalization factor. The Log-polar frequency coordinates (ξ, ϕ) have the property that scaling and rotation in the image domain correspond to translations along ξ and ϕ, respectively. Thus, a set of modified Gabor filters is constructed as a rectangular lattice of identical Gaussians in the Log-polar frequency plane, which simplifies the design of the filter bank. Modified filters reproduce the familiar "daisy" structure of Gabor filters in the standard frequency coordinates (ω_x, ω_y) [Fig. 2(b)]. However, the overlap toward lower frequencies is significantly reduced [Fig. 2(c)].

We employ a set of 24 modified Gabor filters $\mathcal{F} = \{f_i(\omega_l, \phi_j) | 1 \leq l \leq 3, 1 \leq j \leq 8\}$ obtained by taking eight distinct orientations ϕ_j and 3 angular frequencies ω_l. Orientations ϕ_j are evenly distributed over the range $[0, \pi]$, i.e., $\phi_j = j\pi/8$. The angular frequencies ω_l have been determined by estimating three filter wavelengths λ_l suitable for our application, and then using the relation $\omega_l = 2\pi/\lambda_l$. The values $\lambda_1 = 8$, $\lambda_2 = 16$, and $\lambda_3 = 32$ pixels have been chosen. Fig. 2(b) shows the entire set $\mathcal{F}$ of modified Gabor filters in the standard Fourier domain.

Working images $\mathbf{I}(\vec{x})$ are obtained from raw visual data through histogram equalization, resolution reduction (from 768×576 to 422×316 pixels), and gray-value remapping into $[-1, 1]$. A sparse retinotopic graph obtained by Log-polar mapping is used to sample the Gabor decomposition of the images [24]. A high resolution of points characterizes a localized region of the view field (fovea), whereas peripheral areas are characterized by a low-resolution vision. The retinotopic graph is then centered on the image, and the magnitude of the response of each Gabor filter f_i is computed at the location of the N_p nodes of the graph (Fig. 3). At each retinal point $\vec{u}$ we place the 24 modified Gabor filters $f_i^{\vec{u}} \in \mathcal{F}$. This yields a population of overlapping Gaussian receptive fields that tend to cover the entire image continuously. The density of the coverage is higher at the center of the image and decreases toward the peripheral regions. For each point $\vec{u}$, the magnitude of the response of all $f_i^{\vec{u}}$ filters is computed by

$$r_i(\vec{u}) = \left(\left(\sum_{\vec{z}} \mathrm{Re}\left(f_i^{\vec{u}}(\vec{z})\right) \cdot \mathbf{I}(\vec{u}+\vec{z}) \right)^2 + \left(\sum_{\vec{z}} Im\left(f_i^{\vec{u}}(\vec{z})\right) \cdot \mathbf{I}(\vec{u}+\vec{z}) \right)^2 \right)^{1/2} \tag{6}$$

where $\vec{z}$ varies over the area occupied by the filter $f_i^{\vec{u}}$ in the spatial image domain, $\mathbf{I}(\vec{u}+\vec{z})$ is the image sampled at point $\vec{u}+\vec{z}$, and Re and Im indicate, respectively, the real and imaginary components. We assume that the response vectors $\vec{r}(\vec{u})$ provide a sufficient characterization of the visual clues. This biologically inspired image representation has been previously used for real-time head tracking [25] and face authentication [26], showing high discrimination power.

b) Encoding visual inputs by neural activity: The retinotopic system may be abstracted as a three-dimensional filter-activity space: The angular frequency ω_l and the orientation ϕ_j of each filter f_i provide the first two dimensions, while the spatial distribution of points $\vec{u}$ on the image provides the third one. We take the responses of filters $f_i^{\vec{u}}$ $(\forall\, i, \vec{u})$ as inputs to a population of view cells (VC) one synapse downstream from the filter layer. That is, we model each image $\mathbf{I}(\vec{x})$ by mapping its representation $\mathbf{I}_f$ in the filter-activity space into VC activity.

Let k be an index over all $K = 24 \times N_p$ filters forming the retinotopic grid. Given an image $\mathbf{I}$, a view cell $j \in VC$ is recruited to receive inputs from all f_k filters. Synaptic connections w_{jk} from filters f_k to unit j are initialized according to $w_{jk} = r_k\ \forall\, k$, where r_k is the response of filter f_k given by (6). Weights w_{jk} provide a long-term memory for the filter activity associated to image $\mathbf{I}$. If, at a later point, the robot sees an image $\mathbf{I}'$, the activity r_j of unit $j \in VC$ is computed as a function of the similarity of the current image $\mathbf{I}'$ to the image $\mathbf{I}$ stored in weights w_{jk}. The synaptic input to unit j is given by

$$I_j(t) = \frac{1}{K} \cdot \sum_k |r_k(t) - w_{jk}| \tag{7}$$

where r_k are the Gabor filter responses to image $\mathbf{I}'$. The firing rate $r_j(t)$ of cell j is determined by means of a Gaussian transfer function f

$$r_j(t) = f(V_j(t) \cdot (1 \pm \epsilon)) = \exp\left(\frac{-(V_j(t) \cdot (1 \pm \epsilon))^2}{2\sigma^2}\right) \tag{8}$$

where the membrane potential $V_j(t)$ is computed according to (2), and $\epsilon \in [0, 0.1]$ is random noise. Equation (8) defines a radial basis function in the filter space measuring the similarity between $\mathbf{I}'$ and the image $\mathbf{I}$ encoded by w_{jk}. The width σ determines the discrimination capacity of the system for visual scene recognition.

c) Unsupervised growing network scheme for building allothetic place fields: View cell activity depends on the agent's gaze direction and does not code for spatial locations $\vec{s}$. We apply unsupervised Hebbian learning to achieve allocentric spatial coding one synapse downstream from the VC layer. Each $ViPC$ cell receives inputs from a set of view cells whose activities code for visual features of the environment. Thus, the activity r_i of a unit $i \in ViPC$ depends on the combination of multiple visual cues, which makes unit i location sensitive (i.e., a place cell).

At each new location (a measure of familiarity for spatial locations will be defined in Section II-C.4), all simultaneously active VC units are connected to a newly recruited $ViPC$ cell. Let i and j denote $ViPC$ and VC units, respectively. A connection w_{ij}^{new} is created such that

$$w_{ij}^{new} = \mathcal{H}(r_j - \varepsilon) \cdot \mathrm{rnd}_{0,1} \tag{9}$$

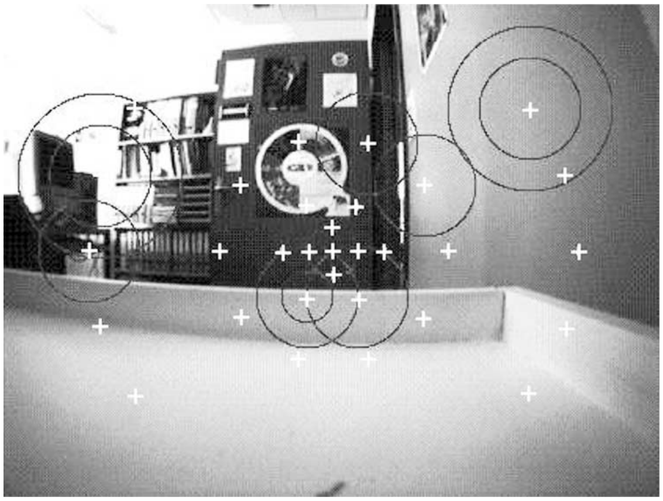

Fig. 3. The retinotopic grid, used to sample the Gabor decomposition, overlaid on the visual scene. The retina consists of $N_p = 31$ points (white crosses) arranged on $N_c = 5$ concentric circles. The innermost circle has radius zero and coincides with the center of the image. The radii of the remaining $N_c - 1$ circles increase exponentially. On each circle, the retinal points are evenly distributed. A vector of Gabor responses is computed at each retinal point. The supports of a few filters are displayed in black.

where $\mathcal{H}$ is the step (Heaviside) function (i.e., $\mathcal{H}(x) = 1$ if $x \geq 0$, $\mathcal{H}(x) = 0$ otherwise), $\varepsilon = 0.75$ is the activity threshold above which a view cell is considered to be active, and $\mathrm{rnd}_{0,1}$ means that each new connection is initialized by a random weight $w_{ij}^{new} \in (0, 1)$. The firing rate r_i, with $0 \leq r_i \leq 1$, of a place unit $i \in ViPC$ is a weighted average of the activity of its afferent signals r_j. The synaptic drive $I_i(t)$ is

$$\begin{aligned} I_i(t) &= (\vec{w_i})^T \cdot \vec{r}_j(t) \cdot \mathcal{H}(\rho - \nu(t)) \\ &= \sum_j w_{ij} \cdot r_j(t) \cdot \mathcal{H}(\rho - \nu(t)) \end{aligned} \tag{10}$$

where j varies over all VC that have been connected to cell i according to (9), $\rho = 0.95$ is the synaptic transmission probability, and $\nu(t)$ is uniformly drawn from $[0, 1]$. Then

$$r_i(t) = f(V_i(t) \cdot (1 \pm \epsilon)) = \frac{V_i(t) \cdot (1 \pm \epsilon)}{\sum_j w_{ij}} \tag{11}$$

where $V_i(t)$ is computed according to (2), and $\epsilon \in [0, 0.1]$ is random noise.

Once connections w_{ij} are established, their synaptic strength is changed by Hebbian learning

$$\Delta w_{ij} = r_j \cdot (r_i - w_{ij}). \tag{12}$$

The rationales behind (12) are to increase the weight w_{ij} whenever pre- and postsynaptic units are simultaneously active, and decrease w_{ij} whenever the presynaptic VC unit j is active while the postsynaptic $ViPC$ cell i is not. Note that (12) keeps the weight $w_{ij} \leq 1$.

The learning scheme defined by (9)–(12), is referred to as "unsupervised growing network learning" [27]. When the robot first enters a novel environment, the $ViPC$ population is empty (since there is no prior spatial knowledge). The $ViPC$ ensemble grows incrementally as a result of the robot interacting with the environment (Section II-C.4). At each visited location $\vec{s}(t)$, the robot takes four views v_1, v_2, v_3, v_4, with $v_{i+1} = v_i + 90^\circ$, and creates four VC cells (one for each view v_i) which are bound together to form a quasi-panoramic description of the place. This results in nondirectional $ViPC$ place cell activity.

2) Idiothetic Space Coding: Path Integration: Internal movement-related signals are used to drive a population of place units, $PiPC$ (path integration driven place cells), providing idiothetic space coding (right side of Fig. 1). The ensemble activity $\mathcal{R}^{PiPC}(t) = \{r_i(t) | i \in PiPC\}$ encodes the robot's current position $\vec{s}(t)$ within an allocentric frame of reference centered at the starting location $\vec{s}(t_0)$. $PiPC$ units have preconfigured metric interrelations: *(i)* One cell $i_o \in PiPC$ codes for the origin of an abstract frame of reference $\mathcal{S}'$ underlying the $PiPC$ representation. When the robot enters a novel environment, i_o is associated with the entry position $\vec{s}(t_0)$ belonging to the physical space $\mathcal{S}$. *(ii)* Each cell $i \in PiPC, i \neq i_o$, has one preferred firing location $\vec{s}_i$ relative to the origin. Preferred positions $\vec{s}_i$ are evenly distributed over the 2-D abstract space $\mathcal{S}'$. Note that, since the abstract space $\mathcal{S}'$ is mapped onto the physical space $\mathcal{S}$ depending on the entry position $\vec{s}(t_0)$, a novel environment may be encoded by two distinct $PiPC$ firing patterns if two explorations start at $\vec{s}^{\,1}(t_0) \neq \vec{s}^{\,2}(t_0)$. In other words, $PiPC$ cells have preconfigured metric relations within the abstract allocentric space $\mathcal{S}'$, but not relative to a physical absolute framework $\mathcal{S}$. As discussed in Section III, Hebbian learning is employed to combine idiothetic and allothetic representations by correlating the activity patterns of $PiPC$ cells with the local views encoded by the $ViPC$ population. This allows the system to establish a $\mathcal{S}' \rightarrow \mathcal{S}$ mapping such that $PiPC$ cells can maintain similar firing patterns across subsequent entries in a familiar environment.

As the robot moves, $PiPC$ cell activity changes according to self-motion information (i.e., translational and rotational signals) which are integrated over time by the path integrator. The firing rate $r_i(t)$ of a cell $i \in PiPC$ at time t is algorithmically taken as a Gaussian

$$r_i(t) = \exp\left(-\frac{(\vec{s}_d(t) - \vec{s}_i)^2}{2\sigma^2}\right) \tag{13}$$

where $\vec{s}_d(t)$ is the robot's current position (relative to the starting point $\vec{s}(t_0)$) estimated by dead-reckoning, $\vec{s}_i$ is the preferred location (center of the receptive field) of cell i, and $\sigma \approx 100$ mm defines the width of the field.

3) Combining Allothetic and Idiothetic Representations: Allothetic and idiothetic representations converge onto the HPC layer of the model. $ViPC$ and $PiPC$ project to HPC by means of synapses that are established based upon a correlational learning rule. According to our unsupervised growing network scheme (Section II-C-I), the system recruits a new subset of HPC place units for each new location visited by the robot. Let i and j be a postsynaptic HPC unit and a presynaptic cell in $ViPC$ or $PiPC$, respectively. New connections are formed from all simultaneously active $ViPC$ and $PiPC$ cells to the new HPC cells [according to (9)]. Then, Hebbian learning is used to establish the weight w_{ij} of those connections

$$\Delta w_{ij} = r_i \cdot r_j \cdot (1 - w_{ij}). \tag{14}$$

The firing rate r_i of a cell $i \in HPC$ is simply a weighted average of its $ViPC$ and $PiPC$ inputs ((10), (11)).

To interpret the ensemble place cell activity as a spatial location we apply population vector decoding [4]. Let $\vec{s}_i$ be the center of the place field of cell i. The population vector $\vec{p}(t)$ is the center of mass of the ensemble pattern of activity at time t:

$$\vec{p}(t) = \frac{\sum_i \vec{s}_i \cdot r_i(t)}{\sum_i r_i(t)}. \tag{15}$$

The encoded position $\vec{p}(t)$ is near, but not necessarily identical, to the robot's actual location $\vec{s}(t)$. The approximation $\vec{p}(t) \approx \vec{s}(t)$ is good for large neural populations covering the environment densely and uniformly [28].

4) Maintaining Allothetic and Idiothetic Signals Consistently Over Time: Exploratory Behavior: Within a novel environment, three issues are relevant to the robot's exploratory behavior: (i) a measure of "familiarity" for spatial locations must be defined; (ii) the environment must be explored uniformly; (iii) allothetic $(ViPC)$ and idiothetic $(PiPC)$ representations must be maintained coherent over time to yield stable place coding.

d) Estimating place familiarity: Place units are recruited incrementally as the robot explores new locations. The familiarity of a location $\vec{s}(t)$ is measured in terms of the response of the current hippocampal place cell population (HPC). New place units are recruited only if the following relation is false

$$\sum_{i \in HPC} \mathcal{H}(r_i(t) - \varepsilon) \geq C \tag{16}$$

where $\mathcal{H}$ is the Heaviside function, $r_i(t)$ is the activity of unit i at time t, and $\varepsilon = 0.75$ and $C = 10$ are fixed thresholds. Thus, at time t, the space representation is updated only if the number of place units active at location $\vec{s}(t)$ does not exceed a threshold C. Equation (16), which is a mere algorithmic implementation of novelty detection in the state space, enables the system to control the redundancy level of the spatial model.

e) Uniform coverage of the environment: The robot adopts a simple active-exploration strategy that helps to cover the environment uniformly. Given the current location $\vec{s}(t)$, it updates its direction of motion $\phi(t)$ based on place cell activity. According to (16), a relatively large number of currently active units indicates that $\vec{s}(t)$ is a familiar location of the environment. Then, a small directional change $\Delta\phi_s(t)$ implies that the robot moves nearly straight and induces a high probability of leaving that area. Conversely, a large variability of the robot's direction $\Delta\phi_l(t)$ is associated to low HPC activity, which results in a thorough exploration of that region. We randomly draw $\Delta\phi_s$ and $\Delta\phi_l$ from $[-5, +5]^\circ$ and $[-60, +60]^\circ$, respectively.

f) Coherence between allothetic and idiothetic maps: path integration calibration: The robot starts exploring an unfamiliar environment by relying upon path integration only. The entry location $\vec{s}(t_0)$ becomes the reference point (home) relative to which the idiothetic representation $PiPC$ is built. The continuous integration of translations $\Delta s(t)$ and rotations $\Delta\phi(t)$ over time generates a homing vector $\vec{h}(t) = (l_h, \theta_h)$ providing the robot with the distance $l_h(t)$ and direction $\theta_h(t)$ relative to the starting point $\vec{s}(t_0)$, at time t. As exploration proceeds, local views encoded by $ViPC$ activity are integrated into the spatial framework provided by the path integrator (as discussed in Section II-C.3). However, the idiothetic-based dynamics, which consists of integrating translational and rotational signals over time, induce systematic as well as nonsystematic errors that quickly disrupt the position representation [29]. Thus, to maintain the allothetic and idiothetic representations consistent over time, path integration needs to be occasionally calibrated. When available, stable allothetic cues (e.g., visual fixes) may be used to accomplish such a calibration process.

The robot adopts an exploration strategy that emulates the exploratory behavior of animals [29], [30]. At the very beginning, exploration consists of short return trips (e.g., narrow loops) centered at the home location $\vec{s}(t_0)$ and directed toward the principal radial directions (e.g., north, east, and so on). This allows the robot to explore the home region exhaustively. Afterwards, the robot switches to an open-field exploration strategy. It starts moving in a random direction recruiting a new subset of place units at each new location (16). After a while, the path integrator has to be recalibrated. We do not propose a specific uncertainty model for the dead-reckoning system. We simply assume that the path integration error grows monotonically as some function $n(t)$ of time t. Whenever $n(t)$ overcomes a fixed threshold n_{cal}, the robot stops creating place units and starts following its homing vector $\vec{h}(t)$ to return home $(\vec{s}(t_0))$. Thus, during homing, spatial learning does not occur. As soon as the robot arrives and recognizes a previously visited location (not necessarily $\vec{s}(t_0)$), it utilizes the learned allothetic representation $ViPC$ to realign the path integrator.

Let $\vec{p}_v(t)$ be the center of mass of the $ViPC$ ensemble activity computed by population vector coding at time t (15). Let $\sigma_v(t)$ denote the variance of the $ViPC$ ensemble activity around $\vec{p}_v(t)$. We take a fixed variance threshold Σ_v to evaluate the reliability of the $ViPC$ coding and we assume that only if $\sigma_v(t) \leq \Sigma_v$ the signal $\vec{p}_v(t)$ is suitable for recalibrating odometry. We define a weight coefficient $\alpha \in [0, 1]$ as

$$\alpha(t) = \begin{cases} \frac{1-\sigma_v(t),}{\Sigma_v} & \sigma_v(t) \leq \Sigma_v \\ 0, & \text{otherwise} \end{cases} \qquad (17)$$

and we use it to compute the calibrated robot position $\vec{p}_d^{\,*}(t)$

$$\vec{p}_d^{\,*}(t) = \alpha(t) \cdot \vec{p}_v(t) + (1 - \alpha(t)) \cdot \vec{p}_d(t) \qquad (18)$$

where $\vec{p}_d(t)$ is the population vector corresponding to $PiPC$ ensemble firing ((15)). Once the vision-based calibration of the path integrator is done, the open-field exploratory behavior is resumed. This technique (consisting of looped exploratory excursions) allows the robot to propagate exploration over the entire environment by keeping the dead-reckoning error bounded. This also means that the allothetic $(ViPC)$ and idiothetic $(PiPC)$ representations are maintained mutually consistent over time.

Note that during homing behavior, the robot might reach the starting location $\vec{s}(t_0)$ without having realigned its path integration, i.e., without having found a location where $ViPC$ activity is suitable for calibration. In this case the robot would resort to a spiral searching behavior centered around the home location $\vec{s}(t_0)$ and, after having found a calibration point, the open-field exploring behavior would be resumed.

D. Goal-Oriented Navigation: Reward-Based Action Learning

The spatial learning system described in Section II-C enables the agent to localize itself within the environment based on available sensory information. To support goal-oriented navigation [1] the model must also incorporate the knowledge about relationships between the environment, its obstacles and reward locations. We take a population $A = \{a | 1 \leq a \leq m\}$ of locomotor action units one synapse downstream from our place units $i \in HPC$. Each cell a provides an allocentric directional motor command (e.g., go north). Then, the navigation problem is: How can we establish a mapping function $\mathcal{M}\colon \mathcal{P} \to \mathcal{A}$ from the place cell activity space $\mathcal{P}$ to a continuous action space $\mathcal{A} \supseteq A$ to achieve goal-directed behavior? We employ reinforcement learning [31] to acquire $\mathcal{M}$ based on the robot's experience. The robot interacts with the environment and reward-related stimuli elicit the synaptic changes of the $i \to a$ projections w_{ai} to adapt the action-selection policy to the task. After training, the ensemble activity of units $a \in A$ provides a navigation map to support goal-oriented behavior and obstacle avoidance.

For the reinforcement learning paradigm, temporal difference (TD) learning was selected since it relies on sound mathematical foundations and represents a well understood approach [31]. In particular, we utilize Q-learning [32], a TD-based learning technique. Given a target $\vec{g}$, a discrete action set $A = \{\text{north}, \text{south}, \text{west}, \text{east}\}$, $A \subseteq \mathcal{A}$, is recruited. Each unit $a \in A$ receives inputs $\vec{w}_a = (w_{a1}, \ldots, w_{an})$ from all place units $i \in HPC$. Each state $\vec{s}(t)$ is encoded by the ensemble place unit activity $\vec{r}_h(\vec{s}) = (r_1(\vec{s}), r_2(\vec{s}), \ldots, r_n(\vec{s}))$, where n is the number of place units. The state-action value function $Q_w(\vec{s}, a)$ is of the form

$$Q_w(\vec{s}, a) = (\vec{w}_a)^T \cdot \vec{r}_h(\vec{s}) = \sum_{i=1}^{n} w_{ai} \cdot r_i(\vec{s}). \qquad (19)$$

Learning consists of updating the adjustable parameter vector $\vec{w}_a$ to approximate the optimal function $Q_w^*(\vec{s}, a)$. The state-value prediction error is

$$\delta_t = R_{t+1} + \gamma \cdot \max_a Q_t(\vec{s}_{t+1}, a) - Q_t(\vec{s}_t, a_t) \qquad (20)$$

where R_{t+1} is the immediate reward, and $0 \leq \gamma \leq 1$ is a constant discounting factor. The temporal difference δ_t estimates the error between expected and actual reward when at, time t, the agent takes action a_t at state location $\vec{s}_t$ and reaches new state $\vec{s}_{t+1}$ at time $t+1$. Training allows the system to minimize this error signal. The weight vector $\vec{w}_a$ changes according to

$$\Delta\vec{w}_a = \alpha \cdot \delta_t \cdot \vec{e}_t \qquad (21)$$

where $0 \leq \alpha \leq 1$ is a constant learning rate, and $\vec{e}_t$ is the eligibility trace vector [31].

Action learning consists of a sequence of training paths starting at random positions $\vec{s}(t_0)$. Every time the robot reaches the target $\vec{g}$, a new training path begins at a new random location. During training, the robot behaves in order to either consolidate goal-directed paths (exploitation) or find novel routes (exploration). This exploitation-exploration tradeoff is determined by an ϵ-greedy action selection policy, with $0 \leq \epsilon \leq 1$ [31]. At each time t, the agent might either behave greedily (exploitation) with probability $1-\epsilon$ by selecting the best action a_t with respect to the Q-value function, $a_t = \arg\max_a Q_t(\vec{s}_t, a)$, or resort to random action selection (exploration) with probability equal to ϵ. The update of the eligibility trace $\vec{e}_t$ depends on whether the robot selects an exploratory or an exploiting action. Specifically, vector $\vec{e}_t$ changes according to

$$\vec{e}_t = \vec{r}_h(\vec{s}_t) + \begin{cases} \gamma \cdot \lambda \cdot \vec{e}_{t-1}, & \text{if exploiting} \\ 0, & \text{if exploring} \end{cases} \tag{22}$$

where $0 \leq \lambda \leq 1$ is a trace-decay parameter, and $\vec{e}_0 = \vec{0}$.

After learning, population vector decoding is applied to map the discrete action space A into a continuous action space $\mathcal{A}$ by averaging the ensemble action cell activity. Given a location $\vec{s}(t)$ at time t, the robot's action $\vec{d}(t) \propto \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}$ is a direction in the environment defined by

$$\vec{d}(t) = \frac{\sum_{a\in A} \vec{d}_a \cdot Q_t(\vec{s}_t, a)}{\sum_{a\in A} Q_t(\vec{s}_t, a)} \tag{23}$$

where $\vec{d}_{\text{north}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\vec{d}_{\text{south}} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, $\vec{d}_{\text{west}} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, and $\vec{d}_{\text{est}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ are the four principal directions defined by actions $a \in A$. Equation (23) results in smooth trajectories.

The above action learning scheme also applies to multitarget navigation tasks. Let $\{\vec{g}_1, \ldots, \vec{g}_m\}$ be a set of distinct target types. Whenever the agent encounters a rewarding location $\vec{g}_j$, it recruits a new set of action cells A^j. Let $Q(\vec{s}_t, a^j)$ be the state-action value function when looking for the optimal policy associated with target $\vec{g}_j$, $\vec{w}_{a^j}$ denote the synaptic projections from place cells $i \in HPC$ to cell a^j, and $\{R^1, \ldots, R^m\}$ be the set of reward signals. The Q-learning algorithm defined by (20)–(22) can be applied to optimize the set of functions $Q(\vec{s}_t, a^j)$ for $j = 1, \ldots, m$.

The experimental results have been obtained by taking a learning rate $\alpha = 0.1$, a discount factor $\gamma = 1.0$, and a decay factor $\lambda = 0.9$. The reward-signal function $R(\vec{s})$ is

$$R(\vec{s}) = \begin{cases} 1, & \text{if } \vec{s} = \text{target state} \\ -0.5, & \text{if } \vec{s} = \text{collision state} \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

During training, target $\vec{g}$ is fixed and the robot receives a positive reward whenever it enters a square region G (goal region) centered at $\vec{g}$ and about twice the area occupied by the robot.

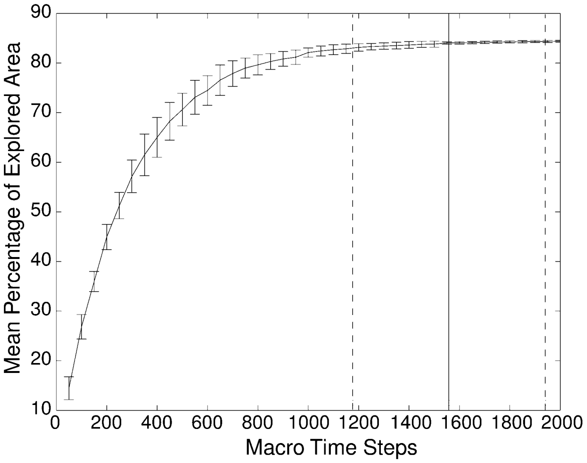

Fig. 4. Convergence of the exploration process averaged over $n = 10$ trials. The curve represents the mean percentage of the area explored by the robot over time. The continuous vertical line indicates when, in average, the robot considered the environment as sufficiently explored. The dashed vertical lines give the standard deviation.

A dynamically changing ϵ-probability is employed to increase the probability of exploring novel routes as the time to reach the target increases. The ϵ parameter is defined by

$$\epsilon(t) = \frac{\exp(\beta t) + k_1}{k_2} \tag{25}$$

where $\beta = 0.068$, $k_1 = 100$, $k_2 = 1000$, and where $t = 0, 1, 2, \ldots$ are macro time steps (defined in Section II-A). We consider a time window T_ϵ of 100 macro time steps. At time $t = 0$, $\epsilon(0) = 0.101$ and exploitation is enhanced, whereas at $t = 100$ a probability $\epsilon(100) = 1.0$ yields exploratory behavior. If after T_ϵ steps the target has not been reached, exploration is further enhanced by maintaining $\epsilon = 1.0$ for $T_\epsilon = 100$ more steps. Then, exploitation is resumed by reinitializing the time window T_ϵ which resets ϵ to 0.101. This process is iterated over time. Note that every time the target is reached the time window is also reinitialized. These are known heuristics to ensure a sufficient amount of exploration.

III. Experimental Results

For sake of clarity, the overall spatial task is considered as two-fold: First, to establish a spatial representation of an unfamiliar environment through exploration (Section III-A), and second to learn the appropriate sensorimotor mapping to perform goal-oriented navigation (Section III-B). The rationale behind this distinction relates to the so-called latent learning (i.e., animals establish a spatial representation even in the absence of explicit rewards [33]). It is shown that having a target-independent space representation (i.e., the HPC place fields) enables the robot to learn target-oriented navigation very quickly.

A. Learning a Place Field Representation

This task consists of placing the robot at a random starting location $\vec{s}(t_0)$ in a novel environment and let it build a map incrementally and on-line through exploration. The exploratory

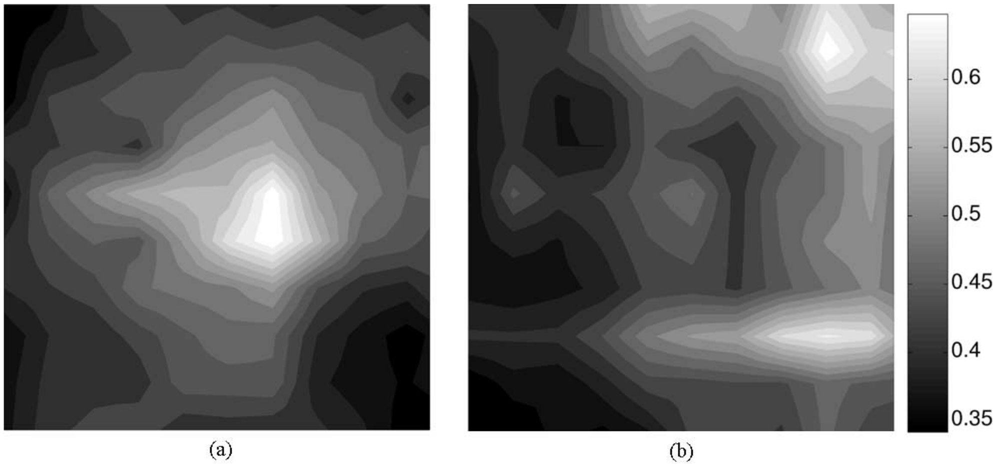

Fig. 5. Two samples of $ViPC$ place fields. Each square box represents the arena. For each position $\vec{s}$, the mean response r_i of a cell $i \in ViPC$ when the robot was visiting $\vec{s}$ is shown. White regions indicate high r_i activity whereas black regions denote low firing rates. (a) Example of single-peak VIPC receptive field. (b) Example of double-peak place field.

behavior depends on an internal familiarity measure which triggers the robot's curiosity and initiates an updating of the map whenever a novel location is detected (Section II-C.4).

1) Convergence of the Exploration Process: Spatial learning is potentially unlimited in time because environmental changes would trigger the robot's exploratory behavior indefinitely. However, for monitoring purposes, we endow the robot with a simple mechanism to self-evaluate the map building process. The robot memorizes the macro time step at which the last updating of the space representation occurred. Then, it considers the environment as sufficiently explored if the unsupervised growing scheme has not recruited new place cells for more than an empirically selected number of macro time steps $T = 100$. Fig. 4 shows the percentage of the environmental area explored by the robot over time. The diagram also illustrates when the robot considered the arena as sufficiently explored. The results were averaged over $n = 10$ spatial learning sessions. At the beginning of each session the robot was placed at a random entry position $\vec{s}(t_0)$ with no prior knowledge of the environment. In average, the robot interrupted the map building process after about 1560 macro time steps, which corresponds to an exploration of approximately 84% of the total state space. Note that, asymptotically, the exploration process does not converge to a 100% coverage. This is because the low-level reactive controller prevents collisions by always keeping the robot at a certain distance from walls.

2) Allothetic (Vision-Based) Place Fields: At each location the robot takes four views and encodes them neurally by means of four view cells (VC). Then, the unsupervised learning scheme combines the gaze-dependent activity of the four view cells to drive a downstream population of $ViPC$ cells. Due to the combination of multiple local views (forming a quasi-panoramic picture), $ViPC$ cells become location selective, i.e., their activity can be used by the system to discriminate locations based on vision information only.

Single $ViPC$ Cell Recordings. Fig. 5 shows two place fields obtained by recording $ViPC$ units when the robot was moving freely in the environment following learning. Fig. 5(a) shows a unit that is maximally active if the robot is in a localized region of the environment (place field center) and whose firing rate decreases (with a Gaussian-like law) as the robots leaves that area. More than 90% of the recorded cells showed this type of location-correlated firing. However, due to visual aliasing, some cells can have multiple subfields, i.e., they cannot differentiate spatial locations effectively. For instance, the cell of Fig. 5(b) has a double-peak receptive field, that is it encodes ambiguous visual inputs and indicates distinct spatial locations providing similar visual stimulation.

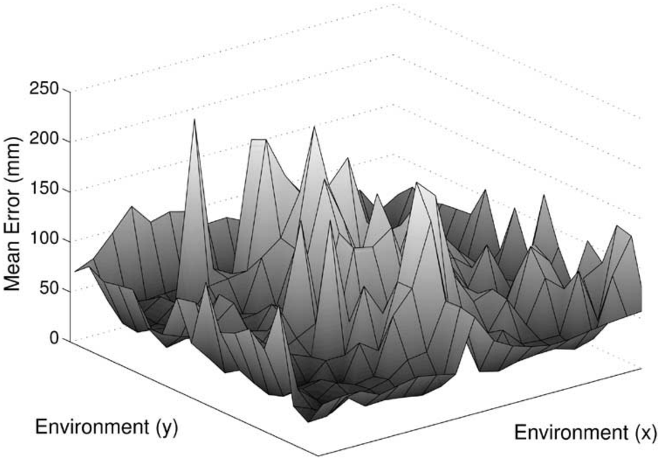

Fig. 6. Accuracy of the $ViPC$ representation over the 2-D environmental space. The diagram has been obtained by discretizing the environment by a 18×18 matrix, and then rastering uniformly over the grid. For each grid cell center visited by the robot, the z-axis is the position reconstruction error (averaged over 10 trials) when applying population vector coding to the $ViPC$ ensemble activity. The mean position error is about 60 mm.

Accuracy of the Allothetic Spatial Representation. Population vector decoding (15) is employed to interpret the ensemble activity $\mathcal{R}^{ViPC}(t) = \{r_i(t) | \forall\, i \in ViPC\}$ as spatial locations. Examining the population rather than single cell activity allows the system to compensate for inaccuracies in the $ViPC$ cell activity [e.g., Fig. 5(b)].

The $ViPC$ ensemble activity packet moves over the 2-D space tracking the robot's movements. The accuracy of the representation is not uniformly distributed over the arena

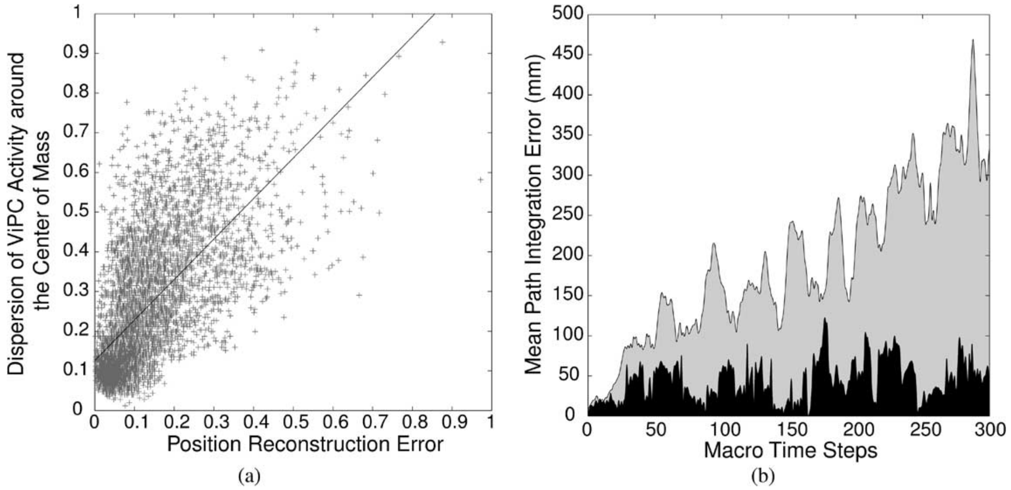

Fig. 7. (a) Correlation between the (normalized) vision-based position reconstruction error and the normalized dispersion σ of $ViPC$ ensemble activity around the center of mass $\vec{p}_v$. (b) Uncalibrated (light-gray curve) and calibrated (black curve) mean path integration error.

surface. The mapping from the visual input space to the 2-D view-manifold reflects the reliability of local visual stimuli, such that locations characterized by ambiguous local views will be poorly encoded by $ViPC$ activity. To measure the accuracy of the $ViPC$ representation as a function of the robot's position the mean quadratic tracking error e is employed

$$e = \frac{1}{n}\sum_{i=1}^{n}\Big((x'-x)^2 + (y'-y)^2\Big)^{1/2} \tag{26}$$

where $\vec{s} = \begin{pmatrix} x \\ y \end{pmatrix}$ is the robot's actual position (provided by the the camera above the arena) and $\vec{p}_v = \begin{pmatrix} x' \\ y' \end{pmatrix}$ is the estimate provided by the $ViPC$ ensemble activity. Fig. 6 shows a result obtained by averaging over $n = 10$ trials. Some regions of the arena are characterized by a rather precise representation, whereas others are poorly represented by the $ViPC$ firing pattern. The average error over the arena is about 60 mm.

3) Path Integration Calibration: The robot integrates translational and rotational self-movements in order to maintain an environment-independent representation $(PiPC)$ of its position relative to the starting point $\vec{s}(t_0)$ (Section II-C-II). However, this dead-reckoning process is prone to cumulative errors over time and needs to be reset occasionally. The vision-based $ViPC$ coding is employed to calibrate the path integrator. As discussed in Section II-C-IV, a criterion to evaluate the reliability of the $ViPC$ coding on-line has been defined to select those locations in the environment that are suitable for the vision-based calibration of the path integrator. As a first approximation, one way to penalize representations of the type found in Fig. 5(b) consists of measuring the dispersion σ_v of the $ViPC$ population activity around the center of mass $\vec{p}_v$. According to this technique, the robot assesses the $ViPC$ ensemble activity by simply employing a threshold Σ_v to detect improper $ViPC$ representations, i.e., those characterized by a dispersion $\sigma_v > \Sigma_v$ (17), (18). Fig. 7(a) indicates the correlation between the vision-based position reconstruction error (26) and the dispersion σ of the $ViPC$ population activity around $\vec{p}_v$. About 4600 data points are represented. The correlation coefficient is 0.67.

To test the vision-based calibration process, an odometry error function $e_d(t)$ is computed during exploration according to (26). At each step t, it measures the difference between the robot's actual position $\vec{s}(t) = \begin{pmatrix} x \\ y \end{pmatrix}$ and the estimate $\vec{p}_d(t) = \begin{pmatrix} x' \\ y' \end{pmatrix}$ provided by the ensemble $PiPC$ activity. We run two series of experiments consisting of $n = 10$ exploration trials each. At the beginning of each trial the robot enters the arena at the same starting position $\vec{s}(t_0)$, and with the same initial arbitrary heading $\phi(t_0)$. In the first series we do not employ vision to calibrate odometry (i.e., darkness conditions). The light-gray curve of Fig. 7(b) shows the mean uncalibrated error $e_d(t)$. The idiothetic representation $PiPC$ is affected by a cumulative shift over time. In the second series of trials we do apply (17), (18) to realign the path integrator occasionally. The black curve of Fig. 7(b) shows the mean calibrated error $e_d(t)$ when performing the vision-based calibration. The odometry error remains bounded over time and has an average value of about 45 mm.

4) Combining Vision and Path Integration: The allothetic $(ViPC)$ and idiothetic $(PiPC)$ maps are combined to drive the HPC population where the allocentric space representation used by the robot to solve spatial tasks is encoded. Similar to $ViPC$ units, HPC cells are recruited incrementally as the robot explores a novel environment.

Single HPC Cell Recordings. Fig. 8 shows two place fields recorded from the HPC layer of the system. Place fields are less noisy than those recorded from $ViPC$ and 97% of the recorded HPC units do not exhibit multiple subfields.

Multiple HPC Cell Recordings. Fig. 9(a) shows an example of HPC population activity after spatial learning. In this example the robot recruited approximately 1500 HPC units. As already mentioned, the purpose is to cover the environment by a large population of overlapping basis functions that can be used for the self-localization task. Redundancy helps in terms of stability and robustness of the place code. Note that place units

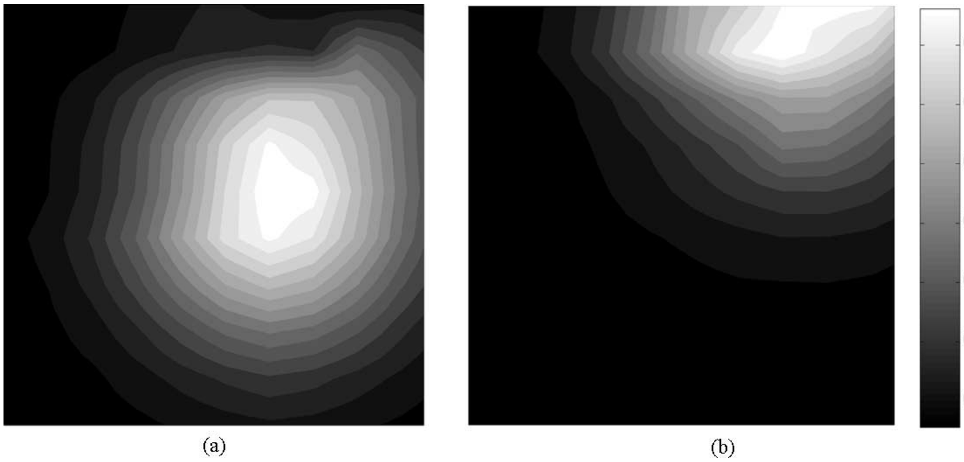

Fig. 8. Two samples of HPC place fields. Each diagram shows the mean firing rate r_i of a cell $i \in HPC$ as a function of the position of the robot $\vec{s}$ within the square arena.

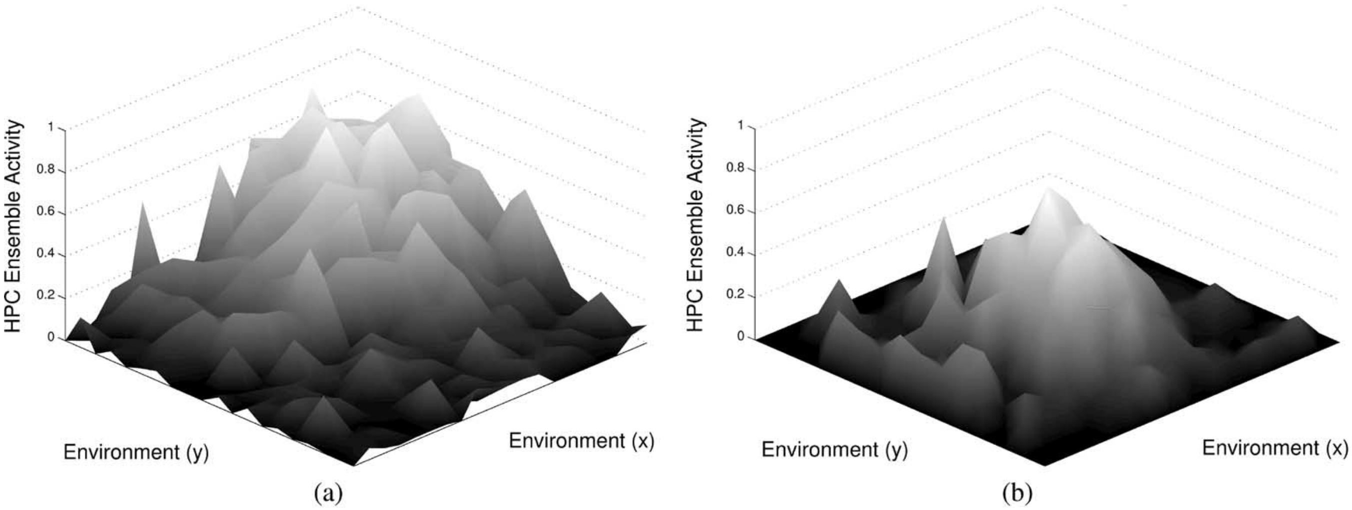

Fig. 9. (a) 3-D representation of the HPC population activity after learning. (b) HPC ensemble activity recorded in the dark (i.e., only the input from the path integrator was present). The robot was approximately at the center of the arena.

are not topographically arranged within the HPC layer of the model. That is, two cells i and j coding for two adjacent locations $\vec{s}_i$ and $\vec{s}_j$, respectively, are not necessarily neighboring neurons in the HPC network. In the image, HPC units are tied to their place field center only for monitoring purposes. Let $\mathcal{R}^{HPC}(t) = \{r_i(t) | \forall\, i \in HPC\}$ be the ensemble HPC activity at time t. We employ population vector decoding (15) to reconstruct the agent's current position based on $\mathcal{R}^{HPC}(t)$.

In the absence of visual information (i.e., in the dark), HPC firing can be sustained by the input provided by the path integration signal. Fig. 9(b) represents the HPC population activity recorded in the dark when the robot was approximately at the center of the arena.

B. Goal-Oriented Behavior

For the goal-oriented navigation task, specific target locations $\vec{g}$ (and the corresponding goal regions G, Section II-D) are defined and the robot has to learn appropriate action selection policies to reach them from any position $\vec{s}$.

Single Target Experiment. Fig. 10(a) shows the navigation map learned by the robot when the goal region G was in proximity of the upper-left corner of the arena. The map was obtained after five training trials. Each training trial starts at a pseudorandom location and ends either when the robot reaches the target or after a timeout of 200 macro time steps. Here, a pseudorandom location means a position randomly drawn from the subset $\mathcal{S}_D = \{\vec{s}_i\}$ of all possible locations having the same distance D from the goal $\vec{g}$, i.e., $\| \vec{s}_i - \vec{g} \| = D$, $\forall\, \vec{s}_i \in \mathcal{S}_D$. At the beginning of each trial the robot estimates its starting location based upon its spatial code HPC. Fig. 10(b) shows the mean search latency, i.e., the number of macro time steps needed by the robot to find the target, as a function of training trials.

The vector field representation of Fig. 10(a) was obtained by rastering uniformly over the whole environment: the ensemble responses of the action cells were recorded at 324 locations distributed over a 18×18 grid of points. Many of the sampled locations were not visited by the robot during goal learning, that is the robot was able to associate appropriate goal-directed

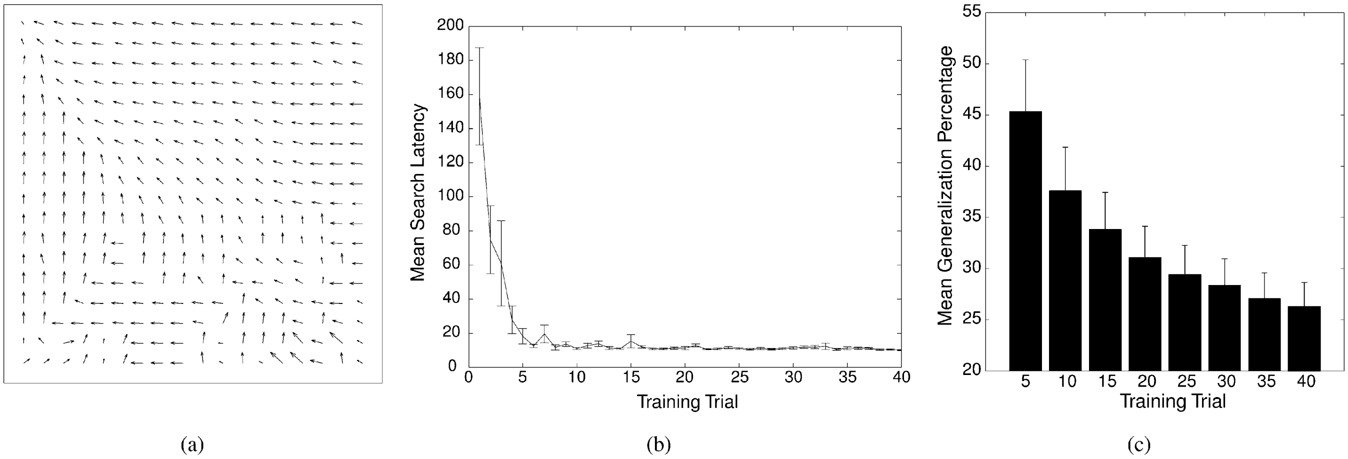

Fig. 10. (a) Navigation map learned after 5 training trials. The target location $\vec{g}$ is nearby the upper-left corner. Each arrow indicates the local action encoded by the action cell ensemble activity after learning. (b) Mean search latency, averaged over $n = 20$ experiments, as a function of training trials. The search latencies decrease rather rapidly and reach the asymptotic value (corresponding to appropriate goal-directed behavior) after approximately 10 trials. (c) Mean amount of generalization as a function of training trials.

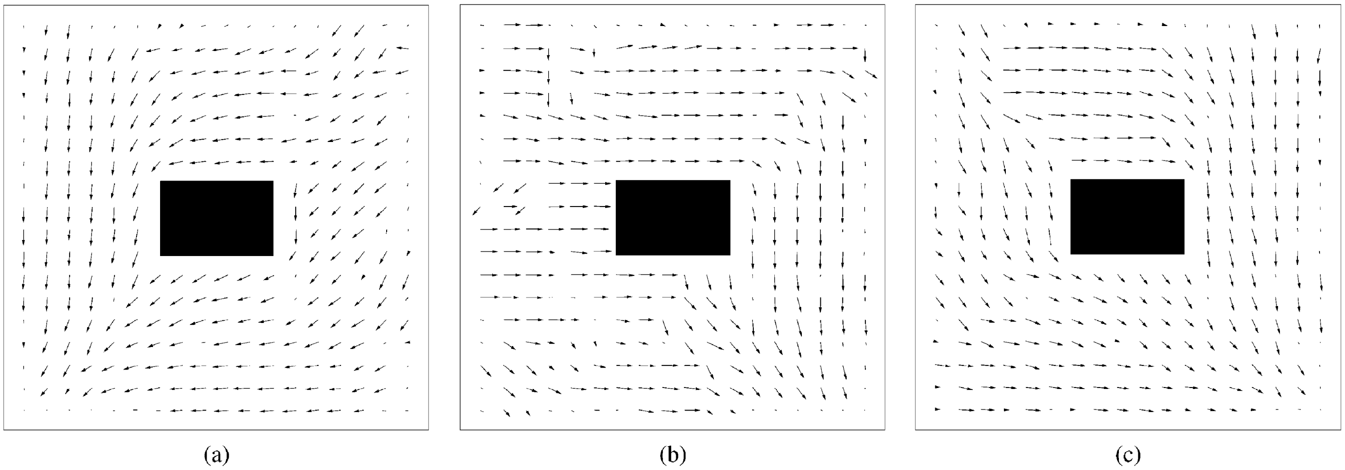

Fig. 11. Navigation task in the presence of two targets $\vec{g}_1$ and $\vec{g}_2$ located at the bottom-left and bottom-right corners, respectively, and one obstacle (black object). (a) Navigation map learned after 30 trials when looking for $\vec{g}_1$. (b) Partial navigation map for $\vec{g}_2$ learned by the robot when focusing on $\vec{g}_1$. (c) Final map acquired by the robot when focusing on $\vec{g}_2$.

actions to spatial states that had not been experienced during training. The amount of generalization is defined as the percentage of sampled positions that were not visited by the robot during learning. A sampled position $\vec{s}_1(t)$ (belonging to the 18×18 grid) is considered as visited during training if the robot was at a position $\vec{s}_2(t)$ such that $\| \vec{s}_2(t) - \vec{s}_1(t) \| \leq$ robot's radius. The amount of generalization was quantitatively measured as a function of the number of training trials. Fig. 10(c) shows the results averaged over $n = 10$ experiments. The diagram shows that for navigation maps acquired after 5 training trials only [as the one of Fig. 10(a)] the mean amount of generalization is about 45%. Then, as expected, the longer the training the lesser the generalization.

Multiple Target Experiment. In this experiment two distinct types of rewards $\vec{g}_1$ and $\vec{g}_2$ are considered. The corresponding goal regions G_1 and G_2 are the bottom-left and bottom-right corners of the arena, respectively. First, the robot is trained to navigate toward G_1. Thus, its primary task is to approximate the $Q(\vec{s}_t, a^1)$ functions to optimize the action-selection policy for G_1. Fig. 11(a) illustrates the navigation map learned by the robot after about 30 trials. When searching for G_1, the robot has a high probability of passing through region G_2 and receiving the positive reward signal $R^2 = 1$. Even if G_2 is not the current target, the robot can exploit this information to adjust weights $\vec{w}_{a^2}$ and start approximating $Q(\vec{s}_t, a^2)$. Thus, when optimizing the policy for G_1, the robot can partially learn a navigation policy to reach G_2. Fig. 11(b) shows the knowledge about target G_2 acquired by the robot while learning a navigation map for G_1. When the robot starts focusing on G_2 (to optimize $Q(\vec{s}_t, a^2)$), it does not start from zero knowledge. This results in a shorter training time for learning the optimal policy for G_2. Fig. 11(c) represents the navigational map acquired by the robot after ten training trials when searching for G_2.

IV. Discussion

This paper presents a neural architecture to endow a robot with space coding and goal-oriented navigation capabilities. The spatial code is established during exploration via two processing streams, an allothetic vision-based representation built by unsupervised Hebbian learning, and an idiothetic representation obtained by integrating internal self-motion signals over time (path integration). Correlational learning is applied to combine these two types of information based on the agent-environment interaction. This induces a mutual benefit in the sense that path integration disambiguates visual singularities and, conversely, visual cues are used for resetting the path integrator. The spatial learning system develops incrementally based upon an unsupervised growing network scheme. This allows the robot to recruit new place cells every time it explores novel regions of the environment. Of course, adding neurons and synaptic connections incrementally is not biologically plausible. Rather, this is a mere algorithmic solution to optimize the use of memory and time resources allocated to the system.

The place field representation HPC provides a basis for guiding goal-directed behavior. HPC cells project to a population of locomotor action cells whose ensemble activity guides navigation. Thus, solving the action learning task means establishing a mapping function from the continuous space of physical locations (encoded by HPC activity) to the activity space of action cells. Temporal-difference (TD) learning is employed to establish this mapping. Algorithms like TD learning are easy to implement for low-dimensional discrete problems. In real world applications, input data are rather high-dimensional and continuous. The most important practical issue for applications of reinforcement learning in these cases is probably the construction of a suitable representation of the input space. The HPC network provides a coarse coding representation suitable for applying reinforcement learning in continuous space. The model also solves the problem of partially hidden states [3], that is the current state is always fully known to the system. Standard reinforcement techniques imply a long training time when applied directly on high-dimensional input spaces. We show that, by means of an appropriate state space representation, the robot can learn goal-oriented behavior after few training trials (e.g., 5 as shown in Fig. 10(a)). This is similar to the learning time of rats in the Morris water-maze [34].

The model captures some properties of hippocampal place (HP) cells, neurons that seem to play a functional role in flexible spatial behavior of rats [1]. In experimental neuroscience, the issue of explicitly relating observations at the neuronal level (i.e., electrophysiological properties of HP cells) to those at the behavioral level (i.e., the animal's capability of solving spatial navigation tasks) remains an arduous task. The lack of experimental transparency at the intermediate levels (e.g., system level) is one of the factors that make it difficult to clearly identify the function of HP cells. It is one of the advantages of modeling that potential connections between findings on the neuronal level and on the behavioral level can be explored systematically. Furthermore, the fact that neuromimetic robots are simpler and more experimentally transparent than biological organisms makes them a useful tool to check new hypotheses concerning the underlying mechanisms of spatial behavior in animals. Synthesizing bio-inspired architectures may help to connect different levels explicitly (e.g., cellular, systemic, behavioral) and bridge the gap between the electrophysiological properties of HP cells and their functional roles in spatial behavior. Conversely, a bio-inspired approach to model spatial cognition offers the prospect of developing robots that can emulate the navigation capabilities of animals and this may lead to an immediate applicational payoff in designing more powerful and adaptive robots.

Acknowledgment

The authors would like to thank D. Floreano, J. del R. Millán, and S. Hug for their contributions.

References

[1] J. O'Keefe and L. Nadel, *The Hippocampus as a Cognitive Map*. Oxford, U.K.: Clarendon, 1978.

[2] J. S. Taube, R. I. Muller, and J. B. Ranck Jr, "Head direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations," *J. Neurosci.*, vol. 10, pp. 436–447, 1990.

[3] R. A. McCallum, "Hidden state and reinforcement learning with instance-based state identification," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 464–473, 1996.

[4] A. P. Georgopoulos, A. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Sci.*, vol. 233, pp. 1416–1419, 1986.

[5] D. J. Foster, R. G. M. Morris, and P. Dayan, "A model of hippocampally dependent navigation, using the temporal difference learning rule," *Hippocampus*, vol. 10, no. 1, pp. 1–16, 2000.

[6] A. Arleo and W. Gerstner, "Modeling rodent head-direction cells and place cells for spatial learning in bio-mimetic robotics," in *Animals to Animats VI*, J. A. Meyer, A. Berthoz, D. Floreano, H. Roitblat, and S. W. Wilson, Eds. Cambridge, MA: MIT Press, 2000, pp. 236–245.

[7] ——, "Spatial orientation in navigating agents: Modeling head-direction cells," *Neurocomput.*, vol. 38–40, no. 1–4, pp. 1059–1065, 2001.

[8] ——, "Spatial cognition and neuro-mimetic navigation: A model of hippocampal place cell activity," *Biolog. Cybern., Special Issue Navigation Biolog. Artif. Syst.*, vol. 83, pp. 287–299, 2000.

[9] A. Arleo, F. Smeraldi, S. Hug, and W. Gerstner, "Place cells and spatial navigation based on 2D feature extraction, path integration, and reinforcement learning," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, pp. 89–95.

[10] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robotics Automation*, 1985, pp. 116–121.

[11] B. J. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robot. Autonomous Syst.*, vol. 8, pp. 47–63, 1991.

[12] R. Chatila and J. P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proc. IEEE Int. Conf. Robotics Automation*, 1985.

[13] S. Thrun, "Learning maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, pp. 21–71, 1998.

[14] A. Arleo, J. del R. Millán, and D. Floreano, "Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 990–1000, 1999.

[15] N. Burgess, M. Recce, and J. O'Keefe, "A model of hippocampal function," *Neural Networks*, vol. 7, pp. 1065–1081, 1994.

[16] H. A. Mallot, M. O. Franz, B. Schölkopf, and H. H. Bülthoff, "The view-graph approach to visual navigation and spatial memory," in *Proc. Artificial Neural Networks—ICANN'97. 7th Int. Conf.*, W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, Eds., Lausanne, Switzerland, 1997, pp. 751–756.

[17] O. Trullier and J.-A. Meyer, "Animat navigation using a cognitive graph," *Biolog. Cybern.*, vol. 83, pp. 271–285, 2000.

[18] 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," *Biolog. Cybern.*, vol. 86, no. 1, pp. 15–28, 2002.

[19] K. Balakrishnan, O. Bousquet, and V. Honavar, "Spatial learning and localization in rodents: A computational model of the hippocampus and its implications for mobile robots," *Adapt. Behav.*, vol. 7, no. 2, pp. 173–216, 1999.

[20] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.

[21] M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff, "Learning view graphs for robot navigation," *Autonomous Robots*, vol. 5, pp. 111–125, 1998.

[22] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Res.*, vol. 20, pp. 847–856, 1980.

[23] H. Knutsson, *Filtering and Reconstruction in Image Processing*: Linköping University, 1982, vol. 88, Linköpings Studies Sci. Technol.: dissertations.

[24] F. Smeraldi, J. Bigün, and W. Gerstner, "On the role of dimensionality in face authentication," in *Proc. Symp. Swedish Soc. Automated Image Analysis, Halmstad*, Sweden, 2000, pp. 87–91.

[25] F. Smeraldi, O. Carmona, and J. Bigün, "Saccadic search with gabor features applied to eye detection and real-time head-tracking," *Image Vision Comput.*, vol. 18, no. 4, pp. 323–329, 2000.

[26] F. Smeraldi and J. Bigun, "Retinal vision applied to facial features detection and face authentication," *Pattern Recogn. Lett.*, vol. 23, pp. 463–475, 2002.

[27] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.

[28] E. Salinas and L. F. Abbott, "Vector reconstruction from firing rates," *J. Computat. Sci.*, vol. 1, pp. 89–107, 1994.

[29] A. S. Etienne, J. Berlie, J. Georgakopoulos, and R. Maurer, "Role of dead reckoning in navigation," in *Spatial Representation in Animals*, S. Healy, Ed. Oxford: Oxford Univ. Press, 1998, ch. 3, pp. 54–68.

[30] B. Leonard and B. L. McNaughton, "Spatial representation in the rat: Conceptual, behavioral, and neurophysiological perspectives," in *Neurobiology of Comparative Cognition*, R. P. Kesner and D. S. Olton, Eds. Hillsdale, NJ: Erlbaum, 1990, pp. 363–422.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning, an Introduction*. Cambridge, MA: MIT Press-Bradford, 1998.

[32] C. J. C. H. Watkins, "Learning From Delayed Rewards," Ph.D. dissertation, Univ. Cambridge, U.K., 1989.

[33] E. C. Tolman, "Cognitive maps in rats and men," *Psycholog. Rev.*, vol. 55, pp. 189–208, 1948.

[34] R. G. M. Morris, P. Garrud, J. N. P. Rawlins, and J. O'Keefe, "Place navigation impaired in rats with hippocampal lesions," *Nature*, vol. 297, pp. 681–683, 1982.

Angelo Arleo received the M.S. degree in computer science from the University of Mathematical Science of Milan, Italy, in 1996, and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2000.

From 2001 to 2003, he worked as an Associate Researcher with the Collége de France-CNRS, Paris. He is currently an Associate Researcher at the Neuroscience Group, SONY Computer Science Laboratory, Paris, France. His research topics concern the neural bases of spatial memory and cognitive navigation in animals, neuromimetic systems, and artificial neural networks.

Fabrizio Smeraldi received the M.S. degree in physics from the University of Genova, Italy, in 1996 and the Ph.D. degree in science from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2000.

He was with the University of Halmstad, Sweden, and is currently with Queen Mary, University of London, U.K.

Wulfram Gerstner received the Ph.D. degree in theoretical physics from the TU Munich, Germany, in 1993, after studies in Tübingen, Berkeley, and Munich.

He is a Professor and Head of the Laboratory of Computational Neuroscience, EPFL, Lausanne, Switzerland, the School of Computer and Communication Sciences, and Brain-Mind Institute. He has been an Invited Researcher in numerous institutions, including Brandeis University, Boston University, and the Courant Institute of Mathematical Sciences.