Biological Inspired System for Localization and Mapping in Underwater Environments

Luan Silveira[1], Felipe Guth[1], Diones Fisher[1], Felipe Codevilla[1], Paulo Drews-Jr[1], Silvia Botelho[1]

Abstract— **One of the most important challenge in robotic is teach a robot how to locate itself in one unknown environment. Nowadays, the most applied solution to this problem relies in Probabilistic Filters. Lead by the discovery of neurons in the mammalian brain associated with navigation tasks, this paper presents a bio inspired system to solve the problem in three-dimensional environments, such as underwater scenarios. Preliminaries results in simulated environments shows the relevance of the proposed method, which is highly parallelizable and capable of running in real time applications.**

I. INTRODUCTION

The robotic mapping has been a widely exploited area in robotics for at least two decades [1], including in the underwater environment [2], [3], [4] The quality of the representative model obtained depends on the characteristics of the environment and sensors. There are several issues that are involved that make a general solution very complex, such as high dimensionality, sensors uncertainty, dynamic of the environment [5]. The localization problem has being addressed together mapping, called Simultaneous Localization and Mapping - SLAM. This term can be defined as build a spatial representation of the environment using data provided by sensors without a priori knowledge, and simultaneously locate the robot in the environment using this representation [6].

Historically, probabilistic approaches dominated the solution to the SLAM problem [7]. Although, in the last decades neuroscientists discovered many neurons in the mammalian brain related to navigation tasks. In 1971, [8] discovered especial cells in rat Hippocampus. This neurons increase their firing rate when the rat cross certain locations in an environment, being called Place Cells. Inspired by this discovery, [9] developed a system called RatSLAM, which uses a Continuous Attractor Neural Network (CANN) to simulate the neuronal activity and represent the pose of a mobile robot in a 2D environment. Together with the network, there is a vision system that provides visual and odometry information. The RatSLAM algorithm build maps in an online fashion, leading to loop-closure and localization through sequences of visual cues.

Here we present an original algorithm, dubbed Hippo 3D. Expanded from the RatSLAM system, this algorithm is based in the *Hippocampus* cells and performs SLAM in 3D environments. The idea is to develop a bio-inspired method to treat the problems of localization and mapping in one of the most appealing and difficulty scenarios: the seabed.

[1]The authors are with Computational Sciences Center, Federal University of Rio Grande - FURG, Rio Grande - RS, Brazil. Email: luansilveira@furg.br.

II. BRAIN CELLS AND LOCALIZATION

The neuroscience is characterized by the scientific study of the nervous system. Although traditionally seen as a branch of biology it is currently an interdisciplinary discipline, collaborating with other fields such as computer science. The techniques used by neuroscientists have evolved, enabling the creation of artificial intelligence methods based on brain functioning, like the artificial neural networks.

During the past decades, but also very recently, a variety of cell types have been identified in certain areas of the mammalian brain that seem to be involved in spatial navigation. Results from experiments with primates or humans are rare, but support the findings from rodents, suggesting that primate and human navigation adheres to the same principles as rodent navigation [10].

The first cell type discovered was the Place Cell, [8] reported results of experiments of neuronal recordings of the activity of nearby cells in the hippocampus. While being inactive most of the time, suddenly the cells fire whenever the rat enters a certain area in the environment, the so called place field of that cell. These place fields differ in size and shape. The place cell activity is driven by visual cues (or landmarks), i.e. the rat is able to recognize places in the environment by visual information. Several experiments proved that the place fields could be moved or otherwise influenced by moving visual landmarks (e.g. [11], [12]). Place cells are still able to represent and update the animals position in a known environment even when no landmarks are visible anymore [13]. This strongly indicates that self-motion cues are used to perform path integration, which corresponds to dead reckoning in the SLAM-context [10].

In a different way to the behavior of place cells, Hafting discovered in Entorhinal Cortex another cell type, with multiple firing fields in the same environment [14]. These locations form a regular tessellating pattern that covers the whole recorded environment. Because of the grid-like structure of the firing fields, the cells were named grid cells. Similar to place cells, the firing fields of grid cells are influenced by visual landmarks.

While place cells and grid cells provide a representation of the animal's position in allocentric space, Head Direction Cells represent the global orientation of the animals head (See Fig. 1). Just like place cells and grid cells, the activity of head direction cells is driven both by visual cues and ego-motion signals. The animal is able to maintain and update the activity in its head direction cells even when moving around in darkness. Although without being updated by visual

978-0-933957-40-4 ©2013 MTS

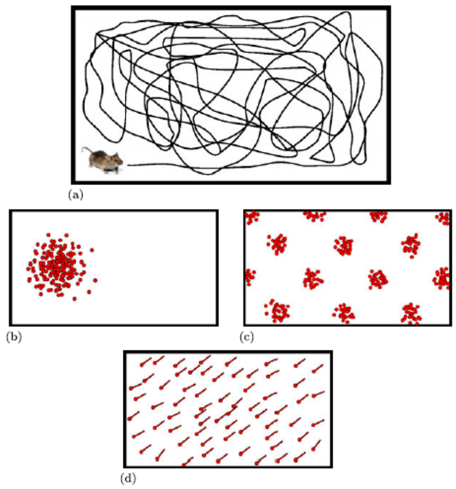


Fig. 1. (a) a rat moving in a rectangular enclosed environment. (b) Place cells firing when the rat is in a certain area of the environment. (c) A grid cell has multiple firing locations that form a grid covering the whole environment. (d) A head direction cell fires whenever the animal looks into a certain direction[10].

landmarks, the head direction represented by the cells begins to drift [15]. [16], [17] related the use of self organizing continuous attractor networks with one and two dimensions to realize path integration with models of Head Direction Cells.

Continuous Attractor Neural Networks - CANN

The generic model of a continuous attractor is a recurrent attractor network with global inhibition [18]. There is a set of neurons that are connected to each other by synaptic weights that are a simple function, for example Gaussian, of the distance between the states of the agent in the physical world represented by the neurons. Neurons that represent similar states have strong connections. One way to set up the weights between the neurons in the continuous attractor network is to use an associative (Hebb-like) synaptic modification rule. In relation to biology, the idea here is that neurons close together in the state space (the space being represented) would tend to be co-active due to the large width of the firing fields [16], [17].

CANN presents not only associatively modifiable recurrent connections between nodes, but also a continuous representation, in which neighboring nodes have overlapping (typically Gaussian) receptive fields. The state space is inherently continuous. These network models are capable of maintain one local activity package.

The generic Continuous Attractor model is a recurrent network with global inhibition, as shown in Figure 2. This network presents a set of *fire* neurons with recurrent connections, which are driven by synaptic weights(w_{ij}^{HD}). Neurons with similar states in the physical world have stronger connections [16]. In addition to these connections, external input(I_i^V) can be applied into the system.

Continuous attractor neural networks usually are used to simulate the behavior of brain cells, like place, grid and head direction cells, in the bio-inspired methods developed with artificial intelligence techniques like in [16], [17], [9].

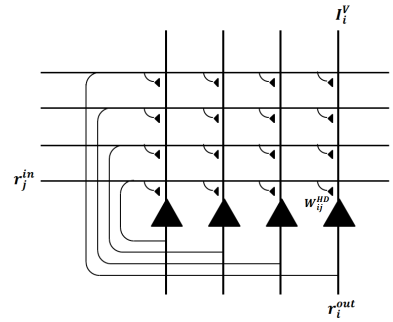


Fig. 2. Continuous Attractor Neural Network model with recurrent connections. In this network each output neuron is feedback to the network through recurrent connections, modulated by synaptic weights. Besides, external inputs also work for the activation of neurons in the network. *Leaky integrator* [18].

III. THE HIPPO3D SLAM SYSTEM

The Hippo 3D algorithm is an extended version of RatSLAM [9] for 3D environments.

The system is divided into three modules, Local View Cells, Pose Cells and Experience Map, as shown in Fig. 3.

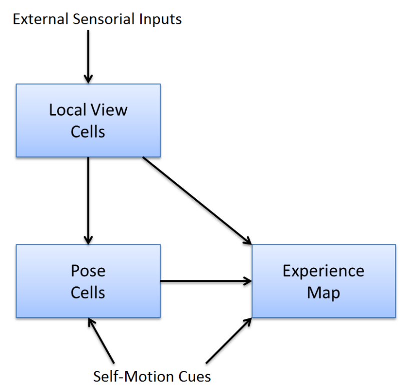


Fig. 3. System architecture.

A. Pose Cell

The Pose Cell is the core of the algorithm. It is composed of a CANN designed to simulate the behaviour of brain mechanisms of navigation, presented in Section II. In our approach, we extended the RatSLAM Pose Cell, adding another layer in the Neural Network, to make the estimation of position in four Degrees Of Freedom. From now on, it's composed of a 4D CANN to estimate the pose xyz and yaw orientation.

B. Local View Cells

The accumulated error in the path-integration process is reset by learning associations between pose cell activity and local view cell activity, while simultaneously recalling prior associations. The local view is represented as a vector V , with each element of the vector representing the activity of a local view cell. During a loop-closure event, the familiar visual scene activates local view cells with learn connections to the pose cells representing the pose where the visual scene was first encountered. Because of the attractor dynamics of the pose cells, a single visual scene is not enough to force an immediate change of pose; several consecutive and consistent views are required to update the pose.

The Data Association was solved using the Bag of Features (BoF) [19] algorithm and each phase of the process will be explained below.

1) Extracting and Recognizing Places: Concerning the tasks of learning and recognizing places visited by the robot in the environment, it was developed an algorithm using Bag of Visual Features [19]. BoF is derived from the well known algorithm in document classification that called Bags of Words. It plays the role of dictionary of visual features, extracted from the camera images of the environment.

The first step is the dictionary creation, which corresponds to a set of every characteristics points in the image. The keypoints extraction is performed by the Speeded Up Robust Features Algorithm (SURF) [20], a robust local feature detector.

A clustering algorithm is required to build a discrete vocabulary from millions (or billions) of local features sampled from the training data [21]. This work uses K-Means algorithm to group similar features [22]. Each cluster region forms a word in the dictionary. This dictionary is the final result of the training phase, when an "*a-priori*" database of environment features. The Figure 4 depicts the general process to represent and recognize places of the environment through the use of visual features.

In the recognizing phase, firstly the SURF algorithm is applied to extract the keypoints. After that, we create a histogram of features, representing the number of features of each group in the image. Then, the histogram is normalized and compared to every templates (histogram) already stored. In the case that one template match with another already created, is injected energy in the neuron associated with this Local View Template of the CANN of Pose Cells, leading to loop closures if the energy is enough to shift the activity packet of the network. Otherwise, when a histogram does not match any created template, a new template is created and assigned to a particular neuron of the CANN of Pose Cells.

2) Training Synaptic Connections: The synaptic association between a Local View Cell with the neurons of a Pose Cell must be created in a automatic form, without supervising, ensuring the creation of new Local View Templates in real time execution.

The chosen training algorithm is the Hebbian rule, that simply consists: - If two neurons connected by a synapse are activated simultaneously, then the force of this synapse is strengthened; - If two neurons connected by a synapse are activated in an asynchronous way, then then the force of this synapse is weakened or eliminated.

At each external input applied in the Pose Cell network,

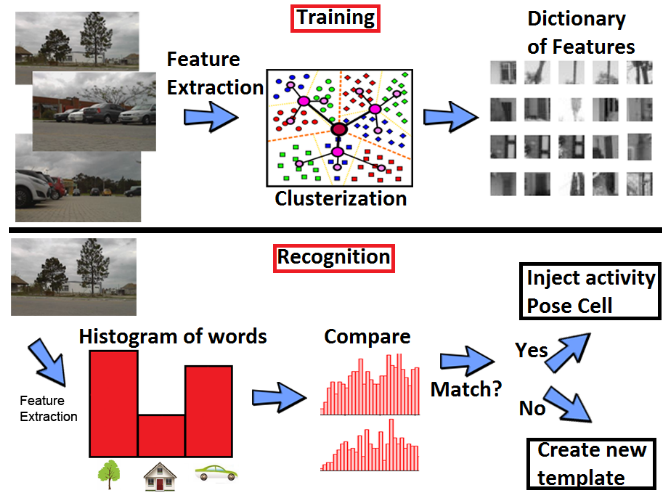


Fig. 4. In the training phase the keypoints extracted from the images with the SURF algorithm are cluster in groups using the K-means algorithm. These groups form the dictionary of the bag of words. In the recognition phase, keypoints are extracted from the image, a histogram of words is generated according to the dictionary. If the histogram, that represents a local view, match with another one stored in the local view module, it means that the robot is in a place seen before and these place in the Pose Cell Network receive energy, leading to loop closure. If there is no match a new template is created in the Pose Cell Network associated with the current local view template.

an update is performed on the synaptic weights associated with this input, according the Hebian rules. The Equation 1 show how is calculated this learning.

$$\Delta\beta^{m}_{ijkl} = k\beta^{m}_{ijkl}p_{ijkl} \tag{1}$$

The variable $\delta\beta_{ijkl}$ represents the variation of the synaptic weights that connect the Local View m to the neuron $ijkl$, where β^{m}_{ijkl} is the current synaptic weight, and p_{ijkl} the activation rate of the neuron $ijkl$ and k a constant for the update process.

In an original way, for the proposed method the Hebbian learning rule was employed, as explained before.

3) Normalization of Inputs: This Hebbian learning has as characteristic the continuum increasing of the injected energy in the Neural Network, this effect must be limited in order to avoid the domain of an external input, i.e., if an external input is repeatedly insert in the network the synaptic connection between it and the active neurons will be increasingly strong. Thus, after a certain time, every time this input is inserted, the activation packet will be shifted instantly, make impossible the competition of the activity packet of the neurons in the network.

In order to maintain the competition in the network, it was implemented a limitation of the maximum energy injected. When an external input, the connection weighs are modified using Eq. 1. If the total energy, i.e., the sum of all the synaptic weights related with this input, is equal to the maximum energy MAX_{ENERGY}, the synaptic weights are modified by the Equation 2.

$$sum = \sum_{i=0}^{N_i} \sum_{j=0}^{N_j} \sum_{k=0}^{N_k} \beta_{ijlk}, \tag{2}$$

$$\beta_{ijkl} = MAX_{ENERGY} * \frac{\beta_{ijkl}}{sum}, \tag{3}$$

4) External Application in the Pose Cell: Every time a sensory information is acquired in the system, the synapses related with that input are activated, applying energy to the Pose Cell, according to Eq. 4.

$$p_{ijkl} = p_{ijkl} + \beta^m_{ijkl}, \tag{4}$$

where, p_{ijkl} represents the actual activation rate of the neuron $ijkl$ and β^m_{ijkl} the link weight of the Local View m with the neuron $ijkl$. This update is performed to all neurons of the network.

C. Experience Map

The experience map is the module where the most improvements are made, because now it is necessary to create a map in three dimensions. For this, the processes of create, update experiences and close loops were modified.

An experience remains a tuple, as shown in Eq. 5.

$$e_i = \{P_i, V_i, pos_i\}, \tag{5}$$

where the new representation of the position pos_i is a 3D pose, as shown in Eq. 6. This pose has an origin and orientation, the last one been coded as a quaternion [23].

$$pos_i = [Origin; Orientation], \tag{6}$$

where

$$Origin = [x, y, z]. \tag{7}$$

$$Orientation = [x, y, z, w]. \tag{8}$$

In the same way, the link between two experience stores the transformation between the experience position, as shown in Eq. 9.

$$l_{ij} = \{T_{ij}\}. \tag{9}$$

This transform has the same components of a pose (see Eq. 6.

The position of a new experience pos_j is calculated multiplying the current experience position pos_i by the current Transform T_{ij}.

1) Loop closure on the experience map: One of the most important events that take place on the map of experiences is the loop closure, where a known experiment is revisited again. When this occurs, it is possible to make an adjustment on the map, in order to reduce odometry errors.

In the proposed method, the odometry error correction is done in a different way, based on the average of estimations for experience position, being more stable than the previous form in RatSLAM, based on error correction through a refresh rate.

$$\begin{gathered} pos_i.origin = pos_i.origin + \textstyle\sum_{j=1}^{n_i}(pos_j * T_{ji}).origin \\ \cdots + \textstyle\sum_{k=1}^{n_o}(pos_k * T_{ik}^{-1}).origin/(1 + ni + no) \end{gathered} \tag{10}$$

where ni represents the connections that arrive in the node i and no the connections from node i to another node in the generated map.

The updated orientation is calculated similarly. The first step is find the yaw angle from the quaternions of all estimates. After that, a mean between these angles is made, transforming them up to Cartesian coordinates, adding vectors, and finding the angle of the resultant vector.

IV. RESULTS

In this section, we will briefly describe the simulator used in the tests, and them present the results of the prototype.

A. The Simulated Scenario

Aiming the execution of the algorithm, our Lab developed a subaquatic environment with characteristics founded in an oil and gas extraction scenario. The ambient was developed using the Blender plataform for rendering and the Robot Operation System for comunication between sensors, algorithm and simulated robot. More information about the simulator can be found in [24].

A snapshot of the environment can be found in Fig. IV-A. The simulated robot was Girona 500, from the University of Girona. [25]. External sensory data was acquired by an image camera, located in front of the robot. A Doppler Velocity Logger and a Compass provide navigation data.

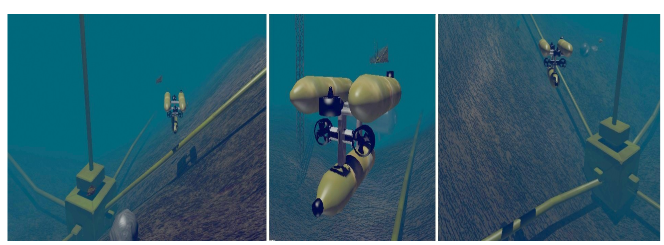


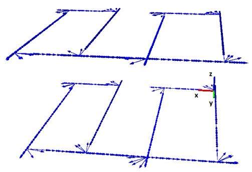
Fig. 5. Screen shots of the underwater simulator. The robot is performing a inspection mission. In the middle the Girona 500 AUV is shown in details.

B. Experiments

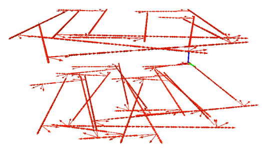
We made two experiments to validate the prototype. The first one was intended to test the neural network and the map formation. In the second one the full algorithm was tested.

1) Case Study One: This test goals validate the Bio-inspired part of the system. For this, we designed a simulated path to the robot. In each position of the path, the visual template was automatically generated, with an uniform probability of 20% to make a wrong template.

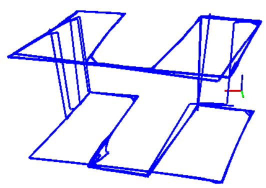
The ground truth data was shown in Fig. 6-a. Fig. 6-b shows the odometry generated on the simulator. This data was generated by inserting a Gaussian noise on the speed developed by the robot. The preliminary results are shown in Fig. 6-c. The position error of the final map is plotted on Fig. 6-d.



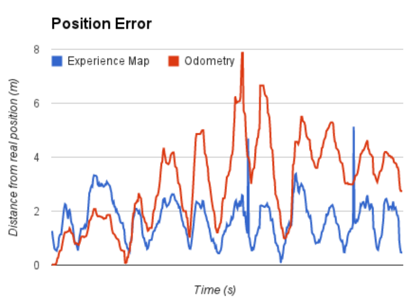
(a) Ground Truth



(b) Odometry information



(c) Map of Experiences



(d) Position Error

Fig. 6. Hippo 3D Results

The Map of Experiences created by the prototype shows a visible improvement compared to the map generated by dead reckoning(Fig. 6-b). The plot of Fig. 6-d shows that the position error decreases when a loop closure events occurs.

2) Case Study Two: This experiment aims to validate the system in a more realistic environment. The underwater robot was assigned to follow the pipes in the Simulator described in Subsection IV-A. The ground truth path was show in Fig IV-B.2 and the generated experience map was show in Figure IV-B.2.

This is an initial result and shows the capacity of the system in create a 3D map of an unknown environment, where only inputs from velocity control and visual images are applied to the system. The loop closure has to be improved. In this experiment the matching algorithm fail to recognize patterns in the scenario.

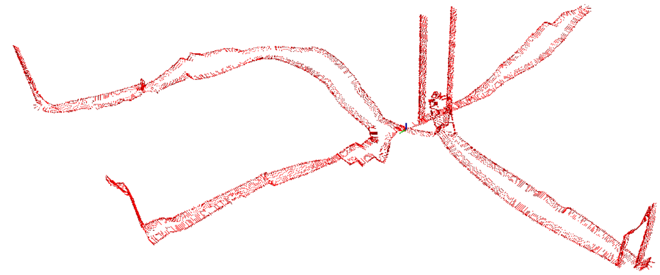


Fig. 7. Ground Truth

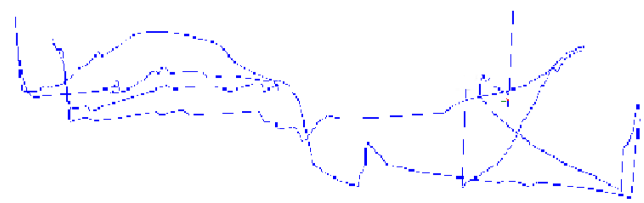


Fig. 8. Map of Experiences

The proposed method has a high scalability, as shown in experiments with the RatSlam system [9], where a suburb with 66 Km was mapped, without increasing the number of neurons in the neural network. Although, a large scale scenario would increase the uncertainty in the estimate activity packet in the network, since the space mapped by each neuron also increase. Conversely, the increasing in the number of neurons in the network provides a better estimation since the discretization is improved, which also increases the processing time to update the network.

V. CONCLUSIONS

This work presented an SLAM System designed for three dimensional environments, called Hippo 3D. The algorithm was tested in a simulated environment and demonstrated its ability to create a 3D map of an unknown environment.

The next steps of our research will focus on test different simulated scenarios in order to calibrate the system parameters, mainly the related with dictionary size to improve the data association.

Afterwards, we will conduct experiments with real robots in order to verify the effectiveness of the method against the uncertainties of the underwater environment. As future work, we propose the algorithm parallelization, to make possible for real time applications. It also intends to verify the biological plausibility of the Hippo 3D method established by the presence of place, grid and head cells in underwater mammals like dolphins and whales.

ACKNOWLEDGMENT

The authors would like to thank to the Brazilian National Agency of Petroleum, Natural Gas and Biofuels(ANP), to the Funding Authority for Studies and Projects(FINEP) and to Ministry of Science and Technology (MCT) for their financial support through the Human Resources Program of ANP to the Petroleum and Gas Sector - PRH-ANP/MCT.

There is also a contribution of the Brazilian National Institute of Science and Technology - INCT-Mar COI funded by CNPq Grant Number 610012/2011-8.

REFERENCES

[1] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002, pp. 163–175.

[2] S. BOTELHO, P. D. Jr., and C. Madsen, "A visual system for distributed mosaics using an auvs fleet," in *IEEE Oceans*, Aberdeen, Scotland, 2007, pp. 332 – 337.

[3] S. BOTELHO, G. Oliveira, and P. D. Jr., "A new approach for visual underwater mapping using topological shell maps." in *IEEE/OES Oceans*, Bremen, Germany, 2009.

[4] M. Figueiredo, S. Botelho, P. Drews, and C. Haffele, "Spatial and perceptive mapping using semantically self-organizing maps applied to mobile robots," in *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*. IEEE, 2012, pp. 245–250.

[5] S. Botelho, P. Drews-Jr, M. Figueiredo, C. Haffele, and G. Oliveira, "Appearance-based odometry and mapping with feature descriptors for underwater robots," *Journal of the Brazilian Computer Society*, vol. 15, pp. 47–54, 2009.

[6] S. BOTELHO, P. D. Jr., and S. Gomes, "Slam in underwater environment using sift and topologic maps," in *5th Latin American Robotics Symposium*, Salvador, Bahia, Brazil, 2008, pp. 91 – 96.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[8] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat," *Brain Research*, vol. 34, no. 1, pp. 171–175, 1971.

[9] M. Milford and G. Wyeth, "Mapping a suburb with a single camera using a biologically inspired slam system," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1038 –1053, oct 2008.

[10] N. Sünderhauf and P. Protzel, "Learning from nature: Biologically inspired robot navigation and slam - a review," *KI - Künstliche Intelligenz*, vol. 24, no. 3, pp. 215–221, 2010.

[11] J. O'Keefe and D. H. Conway, "Hippocampal place units in the freely moving rat: why they fire where they fire." *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, vol. 31, no. 4, pp. 573–590, 1978.

[12] J. O'Keefe and A. Speakman, "Single unit activity in the rat hippocampus during a spatial memory task," *Experimental Brain Research*, vol. 68, pp. 1–27, 1987.

[13] R. Muller, J. Kubie, and J. Ranck, "Spatial firing patterns of hippocampal complex-spike cells in a fixed environment," *The Journal of Neuroscience*, vol. 7, no. 7, pp. 1935–1950, 1987.

[14] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, no. 7052, pp. 801–6, aug 2005.

[15] E. Rolls, "Head direction and spatial view cells in primates, and brain mechanisms for path integration." in *Head Direction Cells and the Neural Mechanisms of Spatial Orientation*, S. Wiener and J. Taube, Eds. The MIT Press, 2005, ch. 14, pp. 299–318.

[16] S. Stringer, T. Trappenberg, E. x. Rolls, and I. de Araujo, "Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells," *Network: Computation in Neural Systems*, vol. 13, no. 2, pp. 217–242, 2002.

[17] S. Stringer, T. Trappenberg, E. Rolls, and I. de Araujo, "Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells," *Network: Computation in Neural Systems*, vol. 13, no. 4, pp. 429–446, 2002.

[18] T. P. Trappenberg, *Fundamentals of Computational Neuroscience*. OUP Oxford, 2002.

[19] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[20] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[21] S. O'Hara and B. A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," *arXiv preprint arXiv:1101.3354*, 2011.

[22] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[23] L. Dorst, D. Fontijne, and S. Mann, *Geometric Algebra for Computer Science (Revised Edition): An Object-Oriented Approach to Geometry*. Morgan Kaufmann, 2009.

[24] F. Gugh, S. Botelho, and M. Amaral, "Underwater simulator for robotics applications in high depths," in *OCEANS 2013 MTS/IEEE Conference*, 2013, in press.

[25] P. R. M. C. D. Ribas, N. Palomeras and A. Mallios., "Girona 500 auv, from survey to intervention." vol. 17, no. 1, 2012, pp. 46–53.