2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Unsupervised Learning of Depth during Coordinated Head/Eye Movements

Marco Antonelli[1], Michele Rucci[2] and Bertram Shi [1]

Abstract— **Autonomous robots and humans need to create a coherent 3D representation of their peripersonal space in order to interact with nearby objects. Recent studies in visual neuroscience suggest that the small coordinated head/eye movements that humans continually perform during fixation provides useful depth information. In this work, we mimic such a behavior on a humanoid robot and propose a computational model that extracts depth information without requiring the kinematic model of the robot. First, we show that, during fixational head/eye movements, proprioceptive cues and optic flow lie on a low dimensional subspace that is a function of the depth of the target. Then, we use the generative adaptive subspace self-organizing map (GASSOM) to learn these depth-dependent subspaces. The depth of the target is eventually decoded using a winner-take-all strategy. The proposed model is validated on a simulated model of the iCub robot.**

I. Introduction

Autonomous robots are expected to work in unstructured environments. To do that, they need to create a coherent representation of their surroundings. Vision provides a remarkable amount of information about an inspected scene, so it is not surprising that it is extensively used throughout the animal kingdom. On the other hand, the 3D structure of the environment is lost when the observed scene is projected onto the 2D camera sensor. Since this information is essential for several tasks, such as object manipulation or image segmentation, several machine vision techniques to estimate depth have been proposed. Stereopsis is probably the most broadly known methods [1], but various other approaches have been proposed, such as motion parallax [2], depth from shading and defocus [3]. The work presented in this paper falls within the range of depth-from-motion techniques, where the camera is actively moved to produce depth-dependent image motion [4], [5].

Most work in this field has focused on large displacements of the agent [4], [6], [7], [8], because they facilitate extraction of depth information and reduce noise sensitivity [9]. However, in humans, useful motion parallax also emerges during much smaller movements, such as the minute involuntary head and body movements that humans continually perform during fixation [10], [11]. Small relocations of the gaze are particularly interesting, as they yield relatively small changes in the images, which greatly facilitate the task of creating a dense depth map. First of all, the long range correspondence problem, i.e. the determination of the positions of identical features in the images acquired from cameras at different locations, can be substituted with estimation of the optic flow, which is an easier problem to solve. Second, small movements reduce the number of occlusions in the image. Finally, small movements enable target depth to be approximate as a invariant during the fixation, which simplifies the temporal integration of sequential depth estimates [12], [13].

Motivated by these observations, our goal is to create a framework to recover depth information in a robot that replicates human fixational head/eye behavior, without prior knowledge of the kinematic model of the robot. In the literature, small movements similar to those performed by humans, including small isolated camera rotations [14], [15] and coordinated head/camera rotations [16], have been exploited to provide useful 3D information in robotic systems. However, in these studies [14], [16] the authors used triangulation to estimate distance from only two images acquired at successive times during fixation. They did not integrate information over time, as humans do [17]. Tagawa proposed a probabilistic model to obtain a dense depth map by integrating visual cues that emerge during drift eye movements [15]. However, he did not consider the advantage of using proprioceptive cues, so the achieved depth map was only unique up to a scale factor [18]. Thus, the achieved 3D representation could not reliably estimate egocentric distance. It can only be used to perform a qualitative analysis of a scene, such as segmentation of objects in the foreground, but not to perform tasks that require physical interaction, such as manipulation.

In previous work [12], [13], we extended the models presented in [14], [16] to progressively refine a 3D representation of a scene by integrating information acquired by the visual and proprioceptive cues over the period of fixation [17]. Our results show that this approach yields accurate and robust 3D representations of the observed scene within the peripersonal space of a humanoid robot [12], [13]. The main limitation of this previous work was that it required the exact kinematic model of the robot and the focal length of the camera in order to estimate the distance. Unfortunately, the calibration process is time consuming and must be repeated every time the geometry of the system and the parameters of the camera change.

In this work, we present a model that learns how to integrate visual and proprioceptive cues to obtain a coherent representation of the depth of the scene. First, we show that the input cues lie on a family of linear subspaces, each one characterized by the depth of the target (see Section II). We find the subspace on which an input pattern lies by using

[1] M. Antonelli and B. Shi are with Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. {eeantonelli,eebert}@ust.hk

[2] M. Rucci is with Department of Psychological and Brain Sciences and the Graduate Program in Neuroscience, Boston University, Boston, MA 02215, USA. mrucci@bu.edu

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

an extension of the Generative Associative Self-Organizing Map (GASSOM) (see Section III). Desired features of this algorithm are the following: 1) it is unsupervised; 2) it is capable of online learning; and 3) it preserves the topology of the subspaces [19]. We implemented the proposed framework on a simulated model of the iCub robot (see Fig. 1). At this stage, we learned the association between visual and proprioceptive cues only for the fixation target, but the model can be extended to estimate the scene of the whole image. Results reported in Section IV show that the algorithm successfully encodes depth information and that decoding can be performed by means of a winner-take-all strategy. A discussion of our main conclusion follows in Section V.

II. Fixational head/eye movements

In this section, we describe the fixational head/eye behavior considered in this study and the relationship among the cues that we took into account. In humans, fixation is the interval between two successive rapid gaze shifts (saccades). Drifts during fixation produces small linear displacements of the visual input around the fixation point [20]. In our setup, these displacements were generated by random movements of the neck that were compensated by eye rotations in order to keep the fixation target in the center of the image.

These neck and eye movements cause the camera to move with velocity $\vec{v}$. The variable $\vec{v}$ is a six-dimensional vector composed of translational, $\vec{t} = [t_x, t_y, t_z]^T$, and rotational, $\vec{\omega} = [\omega_x, \omega_y, \omega_z]^T$, velocities. If the kinematic model of the robot is known, the velocity of the camera can be estimated from the angular position of the head motors, $\vec{\theta}$, and their velocity, $\dot{\vec{\theta}}$ [13]. The relationship between camera and motor velocity can be approximated using the Jacobian matrix $\mathbf{J}(\cdot)$:

$$\vec{v} = \mathbf{J}(\vec{\theta})\dot{\vec{\theta}} \tag{1}$$

The camera velocity generated by the head rotation produces an apparent motion on the image. Considering a pinhole camera with focal length f, at the image location $\vec{p} = [x, y]^T$, the relation between the optic flow and the camera velocity is described by the following equation [21]:

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f & 0 & x \\ 0 & f & y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} \frac{x \cdot y}{f} & -\frac{x^2+f^2}{f} & y \\ \frac{y^2+f^2}{f} & -\frac{x \cdot y}{f} & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \\ = \frac{1}{Z(x,y)} \mathbf{T}(x,y)\vec{t} + \mathbf{R}(x,y)\vec{\omega} \tag{2}$$

where, u_x and u_y are the horizontal and vertical components of the optic flow, respectively; $Z(x, y)$ is the depth of the point in the scene corresponding to the image location $\vec{p}$; $\mathbf{T}$ and $\mathbf{R}$ are matrices that depend only on the image location and the intrinsic parameters of the camera.

In the following, we assume that fixation behavior always starts with the robot looking ahead. Thus, due to the small motion considered in this work, we can assume that the Jacobian is almost constant, that is $\mathbf{J}(\theta) \approx \hat{\mathbf{J}}$. From Eq. 1 and 2 we have:

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} \approx \left(\begin{bmatrix} \frac{1}{Z(x,y)} \mathbf{T}(x,y) & \mathbf{R}(x,y) \end{bmatrix} \hat{\mathbf{J}} \right) \dot{\vec{\theta}} = \mathbf{W}(x,y,Z)\dot{\vec{\theta}} \tag{3}$$

where $\mathbf{W}(x, y, Z)$ is the matrix that represents the relationship between visual and somatosensory cues and it depends on the depth of the scene at the image point $[x, y]$.

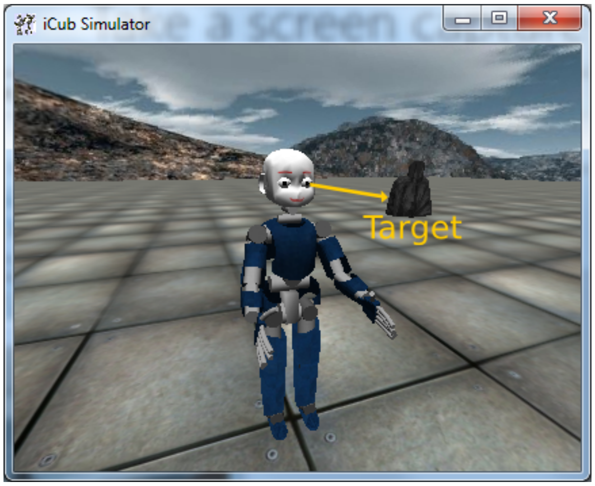


Fig. 1. Simulated environment. The iCub robots is looking at a textured object in front of it.

In this work, we assume the matrices $\mathbf{T}$, $\mathbf{R}$ and $\hat{\mathbf{J}}$ to be unknown. Our goal is to learn the family of matrices $\mathbf{W}(\cdot)$ that link the optic flow with the motor velocity. In this work we focus on learning the matrix $\mathbf{W}(\cdot)$ at only the center of the image. However, the model can be used for any arbitrary image location. The behavior used to move the head produces very small motions of the camera, so that the depth of target can be considered to be constant during each fixation. Thus, the input vectors $u_x(t)$, $u_y(t)$ and $\dot{\vec{\theta}}(t)$, which are observed during a fixation, lie on a linear subspace that is characterized by the target depth. As described in the next section, these subspaces can be learned using the GASSOM [19].

III. The Adaptive Subspace Self-Organizing Map

The Generative Adaptive Subspace Self-Organizing Map (GASSOM) was developed as an unsupervised way to learn invariant feature detectors for natural images. It is based upon the idea that vectors in a high-dimensional input space are distributed along many lower-dimensional manifolds or subspaces [22], [23].

The GASSOM consists of a fixed set of S nodes indexed by $i \in 1, ..., S$ organized in a D-dimensional topological map (latent space). Each node, which represents a subspace, is described by H orthonormal basis vectors specified by the columns of the matrix $\mathbf{B}_i = [\vec{b}_{i1}\, \vec{b}_{i2} \ldots \vec{b}_{iH}]$. Each vector belong to $\mathbb{R}^N$ where N is the dimensionality of the input space. The goal of the GASSOM is to find the basis vectors that describe the observed input patterns using an on-line learning method. To achieve this goal it exploits the concepts of sparsity, *i.e.*, only a single node is responsible to describe the observed input, and temporal slowness, *i.e.*, consecutive inputs are likely to be generated by the same node. The algorithm consists of two steps: the identification of the subspace and the basis vector updates.

In the subspace selection step, the algorithm searches for the node i that maximizes the probability $\mathcal{P}(z_i(t) = 1|\vec{x}(t), \dots, \vec{x}(0))$, where $z_i(t)$ is a binary variable that takes the value 1 if the input $\vec{x}(t)$ is generated by the node i; 0 otherwise. This probability is calculated from the following:

$$\mathcal{P}(z_i(t)|\vec{x}(t), \dots, \vec{x}(0)) = \frac{\alpha_i(t)}{\sum_{j=1}^{S} \alpha_j(t)} \tag{4}$$

where, the joint probability $\alpha_i(t) = \mathcal{P}(\vec{x}(0), \dots, \vec{x}(t), z_i(t))$ is updated recursively using a hidden Markov model (HMM) [24]:

$$\alpha_i(t) = \mathcal{P}(\vec{x}(t)|z_i(t)) \sum_{j=1}^{S} \alpha_j(t-1)\mathcal{P}(z_i(t)|z_j(t-1)) \tag{5}$$

The transition probability between the nodes j and i, $\mathcal{P}(z_i(t)|z_j(t-1))$, can be set as a mixture of a uniform probability and a discrete delta distribution [19]:

$$\mathcal{P}(z_i(t)|z_j(t-1)) = \rho \cdot \frac{1}{S} + (1-\rho) \cdot \delta(i-j) \tag{6}$$

where $\rho \in [0, 1]$ is the parameter that controls the slowness of the model. In this work, we set ρ equal to zero and we reset the node probability at the beginning of each fixation. Although transitions between nodes are disallowed, the HMM formulation is still useful in enabling online updates.

The probability $\mathcal{P}(\vec{x}(t)|z_i(t))$ is calculated by assuming that the observation $\vec{x}(t)$ is the sum of two independent Gaussian variables with diagonal covariance matrix, one lying on the subspace and the other orthogonal to the subspace:

$$\mathcal{P}(\vec{x}(t)|z_i(t)) = \mathcal{N}(\mathbf{B}_i^T \vec{x}(t), \sigma_w^2 \mathbf{I}) \cdot \mathcal{N}({\mathbf{B}_i^{\perp}}^T \vec{x}(t), \sigma_n^2 \mathbf{I}) \tag{7}$$

where σ_w^2 and σ_n^2 are the variance parameters of the Gaussian distribution $\mathcal{N}(\cdot)$. Herein, we set σ_w to infinity, so that the observation function is a single two-dimensional Gaussian.

Once the winning node c has been identified, the original algorithm updates every the basis vectors in order to minimize the reconstruction error [19]. In this work, instead of learning the basis $\mathbf{B}_i$, we learned the basis of the nullspace $\mathbf{B}_i^{\perp}$ because it has a lower dimensionality which allows us to reduce the number of parameters. The basis $\mathbf{B}_i^{\perp}$ can be written as:

$$\mathbf{B}_i^{\perp} = \begin{bmatrix} \mathbf{I}_{2\times 2} \\ -\mathbf{W}_i^T \end{bmatrix} \tag{8}$$

where $\mathbf{I}_{2\times 2}$ is the identity matrix and $\mathbf{W}_i$ is the matrix $\mathbf{W}(x, y, Z)$ introduced in Eq. 3 associated to the subspace with index i. The matrices $\mathbf{W}_i$ are updated using the gradient descent in order to minimize the squared error of the predicted optic flow $\vec{u}(t) = [u_x, u_y]^T$:

$$\Delta \mathbf{W}_i(t) = \gamma h_i(t) \left[\vec{u}(t) - \mathbf{W}_i(t) \dot{\vec{\theta}}(t) \right] \dot{\vec{\theta}}(t) \tag{9}$$

where $\gamma \in [0, 1]$ is the learning rate; $h_i(t)$ is a Gaussian function centered in the winning node c, which ensures neighboring nodes encode similar subspaces:

$$h_i(t) = \frac{g(i|c, \sigma_h)}{\sum_{j=1}^{S} g(j|c, \sigma_h)} \tag{10}$$

In order to make the algorithm more robust to outliers, we saturated the prediction error, $\vec{u}(t) - \mathbf{W}_i(t)\dot{\vec{\theta}}(t)$, when it exceeded $3 \times \sigma_n$.

IV. Experiments and Results

A. Setup

The system was tested on the simulated model of the iCub robot [25] (Fig. 1). The robot is equipped with a six degrees of freedom (d.o.f.) head, enabling yaw/roll/pitch rotations of the neck, and tilt/vergence/version rotations of the eyes. Since our architecture relies on monocular depth cues, in this study we worked with the left camera only. Monochromatic images were acquired by the default camera that has a resolution of 320×256 pixels and a focal length of 257.34 times the pixel size.

Dense optic flow was extracted from two consecutive images using the Farneback's algorithm provided in OpenCV [26]. The optic flow in the center of the image was taken by averaging the dense optic flow in a window of 10-by-10 pixels.

The motors of the robot were controlled in velocity. To replicate human fixational head movements, the neck moved following trajectories generated by an Ornstein-Uhlenbeck process, a constrained random walk [27]. The velocity $\dot{\theta}$ at time t of the three motors in the neck was updated as:

$$\dot{\theta}(t) = \lambda[\mu - \theta(t-1)]\Delta t + \sigma\sqrt{\Delta t}\mathcal{N}(0, 1) \tag{11}$$

where μ is the mean of the process, σ is the standard deviation of the random walk and λ is the drift of the process. In the experiments described below, the parameters were: $\mu = 0°$, $\lambda = 1$ and $\sigma = 1°$. The velocity tilt/version motors of the cameras were controlled to maintain approximate fixation by counteracting neck rotations. The vergence motor was not used in this study.

The size of the matrices $\mathbf{W}_i$ of Eq. (3) and (9) was 2×5: five motor velocities and two components of the optic flow (horizontal and vertical). The number of subspaces was set to 90. The learning rate γ was set to 0.1, while the standard deviation σ_h of the Gaussian function h_i was annealed smoothly from an initial value of 12.6 and to a final value of6.3. The parameter that regulated the transition probability ρ was set to 0, so assuming the depth of the target as constant during a fixation. We set $\sigma_w = \infty$ and $\sigma_n = 0.2$ in Eq. 7.

An object with an applied texture taken from a natural image from the van Hateren database [28] was placed in front of the robot at varying distance was used to train the algorithm (see Fig. 1. The training set consisted of 4544 fixations, whose duration was randomly chosen between 10 and 90 frames. The total number of frames used for training

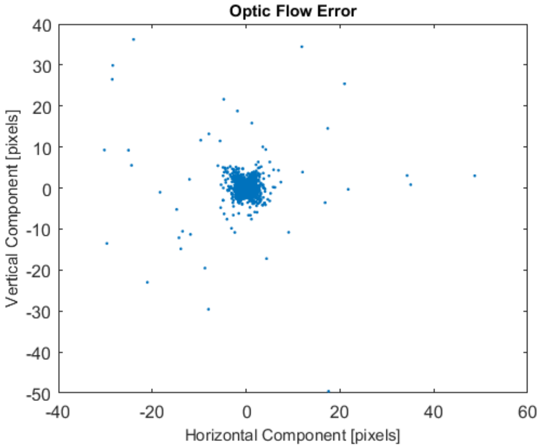


Fig. 2. Distribution of the optic flow error.

the algorithm was 225826. At the beginning of each fixation, we changed randomly the distance of the target and we reset the probability of the nodes to $\mathcal{P}(z_i(0)) = \frac{1}{S}$. The depth of the target was sampled from an uniform distribution between 10 and 100 cm. During the fixation, the estimated optic flow at the center of the image and the velocity of the controlled motors were used to train the GASSOM using the online method described in Section III. Figure 2 shows the distribution of the optic flow error in the training set. The ground truth optic flow was computed from Eq. 2 using the known target depth and the camera velocity provided by the kinematic model of the robot. The dataset contained several outliers. If outliers were removed, the error could be approximated as a Gaussian random variable with parameters $\mathcal{N}(0, 0.2)$.

B. Results

After the training stage, we tested the algorithm on a dataset composed of ten objects with different applied textures also taken from the van Hataren dataset [28], but separate from that used in training. These objects, in turn, were placed at a varying distance between 10 and 100 cm with a step size of 1 cm. For each position of the target we performed a average of 78 fixations, each one with a fixed duration of 30 frames. At the end of each fixation we extracted the index of the winning node. We ran experiments either with and without saturation of the prediction error in Eq. 9. Without saturation, some units did not encode any depth and some depths were encoded by two cells. A total of 16 units out of 90 did not provide useful information, reducing the sensitivity of the algorithm. For this reason, in the remainder of this section we describe the results obtained by saturating the prediction error.

Figure 3 shows the conditional density of the winning node given each input depth for the range of input depths. We found at every depth, the distribution of the winning node can be approximated as a unimodal Gaussian.Fig. 4 plots the mean and the standard deviation of index of the winning node as a function of the target depth. The mean index of the winning node is approximately monotonic in the depth of the target. This behavior is due to the topological organization of the GASSOM: closer cells encode similar target distances. As expected, the standard deviation of the of the winning node, which provides a measure of the accuracy of the method, increases with the depth of the target. This is because the amount of motion parallax is inversely proportional to depth. Note also that the slope of the mean index depends on the depth. The algorithm used more units to encode nearby targets and fewer for farther targets.

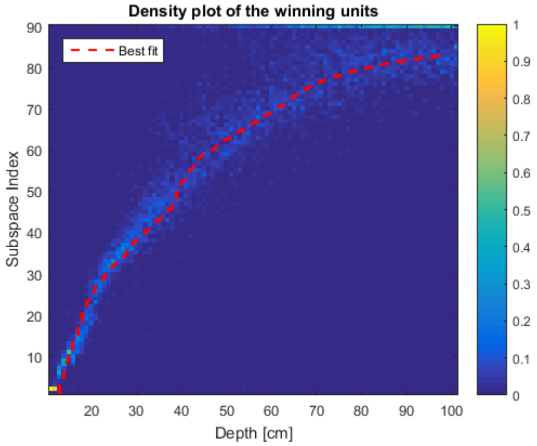


Fig. 3. Conditional density of the winning node of the GASSOM as a function of the input depth range. The density function at each depth is calculated on an average of 78 fixations for each range of input depths (resolution: 1 cm), each one with the fixed duration of 30 frames. Red dashed line is obtained by finding the depth the minimizes the difference between the basis obtained from Eq. 3 and the ones learned by the GASSOM (see the text).

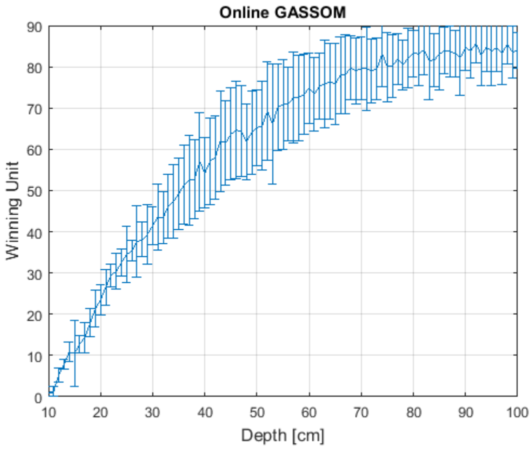


Fig. 4. Mean and the standard deviation of the winning node as a function of the target depth.

Another way to analyze data in Fig. 3, is to plot the conditional probability of the target depth given the winning unit c: $\mathcal{P}(Z|c)$. In Fig. 5 we show the conditional probability for the cells with index starting from 5 to 80, with a step size of 15. We can note that each probability distribution has

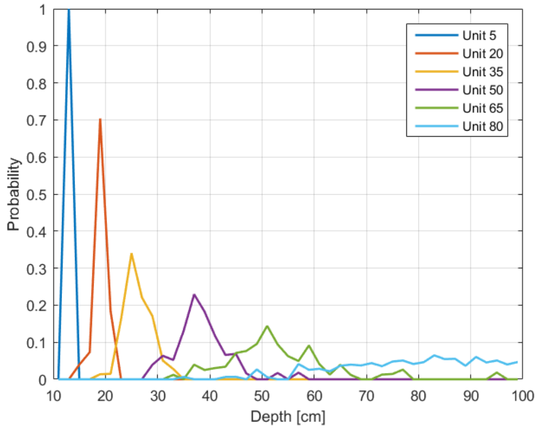


Fig. 5. Conditional probability of the input depth given the winning unit. Fig. shows the probability for the cells with index 5, 20, 35, 50 and 65.

a Gaussian like shape, with a peak value corresponding to increasing target depths for larger indices and a spread that increases with the depth of the target. These plots show that the winning of the GASSOM can be used to estimate the depth, e.g. by mapping the winning unit to the maximum a posteriori estimate.

An alternative way to calculate the depth encoded by the units, is to find the depth Z_i^* that minimizes the distance between the optic flow predicted by the matrix $\mathbf{W}_i$ learned by the GASSOM and the optic flow predicted by the matrix $\mathbf{W}(Z_i^*)$ of Eq. (3). The red dashed line in Fig. 3 shows Z_i^* as a function of the subspace index. We can observe that Z_i^* matches quite well with the peak of the distributions. The valued of Z_i^* for units with index between 84 and 90 are not shown in Fig. 3 because they exceeded 100 cm. For these units Z_i^* increased monotonically from 104.7 to 185.6 cm. For any unit, the root mean squared error between the optic flows predicted by the learned matrices and the fitted matrices was lower than 0.4. Highest errors were obtained for the first 10 basis. The lower accuracy of the first 10 basis can also be observed in Fig. 6, where the difference between the learned and the fitted parameters is higher.

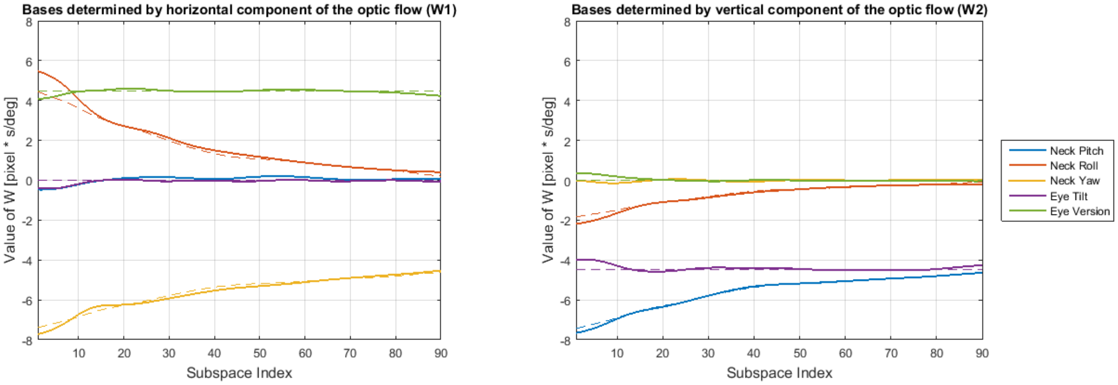
Figure 6 shows the values of the basis $\mathbf{W}$ learned by the GASSOM as a function of the subspace index (solid line). Dashed lines show the fitted parameters. The learned parameters are close to the fitted parameters and change smoothly as a function of the depth due to the topological updates in the GASSOM. These values indicate the transformation that converts head motor velocity into optic flow. Let us focus on the basis of the horizontal component of the optic flow (see Fig. 6(a)). First, we can observe that the horizontal flow depends only on three motors, the neck roll/yaw and eye version, while the parameters of the other two motors are almost zero. The parameter related to the eye version is nearly constant at 4.45 degrees/pixel independently of the target. This suggests that the eye version angle only rotates the camera. The learned value is consistent with 4.49, the product of the focal length (257.34 pixels) and the factor $\frac{\pi}{180}$ converting the radians to degrees. The neck roll motor parameter is inversely proportional to the depth and tends to zero when the object is far away. The neck yaw parameter changes with depth, but reaches a constant value, suggesting that it leads both rotations and translations of the camera. Moving the neck roll joint leads to highest motion parallax. A similar analysis can be also performed for the vertical component of the optic flow. This analysis shows that the algorithm has learned about the kinematic parameters of the robot without supervision, simply by observing the sensorimotor contingencies generated by fixational eye movements.

V. Conclusions

This work is part of our research to develop a sensorimotor framework that allows humanoid robots to create an implicit representation of the peripersonal space based visual and somatosensory cues [29]. While our previous framework created a space representation using binocular cues [29], the model presented here focuses on monocular cues. The goal is to increase robustness by adding a new source of depth information. The estimate of the target depth was obtained by integrating visual and proprioceptive cues that emerge during fixational head/eye movements [12], [13], a behavior that has been proposed to provide reliable depth information within the peripersonal space in humans [10].

In this work, we showed that a faithful representation of the depth is also possible without prior knowledge of the kinematic model of the robot. The model exploits the observation that, during fixation, optic flow and motor velocities lie on a low dimensional subspace that is characterized by the depth of the target. The family of these depth-dependent subspaces can be effectively discovered using a variant of the GASSOM algorithm [19], which learned the null-space imposed by the optic flow equation. In contrast to other unsupervised learning approaches [30], in our approach depth information can be extracted from the learned basis using a winner-take-all strategy. This depth representation is not explicit, i.e., it does not give a value in an artificial Cartesian coordinate system, but is implicit in the activation of the neural network. Nonetheless, it is an absolute representation of the 3D space: the winning unit can be mapped to a corresponding depth. Finally, due to the winner-take-all strategy, just one unit is responsible to describe the observed depth. This decoding strategy is particularly suitable to be integrated with other static and self-motion-based depth cues using reinforcement learning [31].

To conclude, this paper presents a method to learn a representation of the depth at a single image location. Even if results have been provided for the fixation target, the underlying principle can be applied to any image positions. Future work will extend this work to obtain a dense depth map. A parallel implementation of the algorithm will be applied at every image location and the transition probability will be modified to include information from neighboring pixels.



(a) (b)

Fig. 6. Values of the basis $\mathbf{W}$ as a function of the subspace index. Basis for the horizontal and vertical components of the optic flow are shown in panels (a) and (b), respectively. Solid lines: parameters learned by the GASSOM; dashed lines: fitted parameters.

REFERENCES

[1] O. D. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The geometry of multiple images - the laws that govern the formation of multiple images of a scene and some of their applications.* MIT Press, 2001.

[2] B. Rogers and M. Graham, "Motion parallax as an independent cue for depth perception." *Perception*, vol. 8, no. 2, pp. 125–134, 1979.

[3] M. Tao, P. Srinivasa, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from Shading, Defocus, and Correspondence Using Light-Field Angular Coherence," in *Computer Vision and Pattern Recognition (CVPR)*, jun 2015.

[4] Y. Aloimonos and Z. Duric, "Estimating the heading direction using normal flow," *International Journal of Computer Vision*, vol. 13, no. 1, pp. 33–56, 1994.

[5] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.

[6] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 865–880, 2002.

[7] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.

[8] M. Ramachandran, A. Veeraraghavan, and R. Chellappa, "A fast bilinear structure from motion algorithm using a video sequence and inertial sensors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 1, pp. 186–193, 2011.

[9] G. Sandini and M. Tistarelli, "Active tracking strategy for monocular depth inference over multiple frames," *Pattern Anal. Machine Intell., IEEE Trans.*, vol. 12, no. 1, pp. 13–27, 1990.

[10] M. Aytekin and M. Rucci, "Motion parallax from microscopic head movements during visual fixation," *Vision Res*, vol. 70, pp. 7–17, 2012.

[11] M. Poletti, M. Aytekin, and M. Rucci, "Head-eye coordination at a microscopic scale," *Current Biology*, vol. 25, no. 24, pp. 3253–3259, Dec 2015.

[12] M. Antonelli, A. del Pobil, and M. Rucci, *Depth Estimation during Fixational Head Movements in a Humanoid Robot*, ser. Lect. Notes Comput. Sc. Springer Berlin Heidelberg, 2013, vol. 7963, pp. 264–273.

[13] M. Antonelli, A. P. del Pobil, and M. Rucci, "Bayesian multimodal integration in a robot replicating human head and eye movements," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2868–2873.

[14] F. Santini and M. Rucci, "Depth perception in an anthropomorphic robot that replicates human eye movements," in *IEEE International Conference on Robotics and Automation*, Orlando, FL, May 2006.

[15] N. Tagawa, "Depth perception model based on fixational eye movements using bayesian statistical inference," *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1662–1665, 2010.

[16] X. Kuang, M. Poletti, J. D. Victor, and M. Rucci, "Temporal encoding of spatial information during active visual fixation," *Current Biology*, vol. 22, no. 6, pp. 510–514, PMCID: PMC3 332 095, 2012.

[17] K. Hosokawa, K. Maruya, and T. Sato, "Temporal characteristics of depth perception from motion parallax," *Journal of Vision*, vol. 13, no. 1, pp. 16–16, jan 2013.

[18] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 4, pp. 523–535, 2002.

[19] T. N. Chandrapala and B. E. Shi, "Learning Slowness in a Sparse Model of Invariant Feature Detection," *Neural Computation*, vol. 27, no. 7, pp. 1496–1529, jul 2015.

[20] M. Rolfs, "Microsaccades: small steps on a long way," *Vision res*, vol. 49, no. 20, pp. 2415–2441, 2009.

[21] L. H. C. Higgins and K. Prazdny, "The Interpretation of a Moving Retinal Image," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 208, no. 1173, pp. 385–397, 1980.

[22] T. Kohonen, "Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map," *Biological Cybernetics*, vol. 75, no. 4, pp. 281–291, 1997.

[23] A. Hyvrinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[24] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[25] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 57–61.

[26] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image analysis*. Springer, 2003, pp. 363–370.

[27] C. W. Gardiner *et al.*, *Handbook of stochastic methods*. Springer Berlin, 1985, vol. 3.

[28] J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," *Proceedings: Biological Sciences*, vol. 265, no. 1394, pp. 359–366, Mar 1998.

[29] M. Antonelli, A. Gibaldi, F. Beuth, A. Duran, A. Canessa, M. Chessa, F. Solari, A. del Pobil, F. Hamker, E. Chinellato, and S. Sabatini, "A hierarchical system for a distributed representation of the peripersonal space of a humanoid robot," *Autonomous Mental Development, IEEE Transactions on*, vol. 6, no. 4, pp. 259–273, Dec 2014.

[30] K. Konda and R. Memisevic, "Unsupervised learning of depth and motion," *arXiv preprint arXiv:1312.3429*, 2013.

[31] B. J. Grzyb, V. Castelló, M. Antonelli, and A. P. del Pobil, "Integration of static and self-motion-based depth cues for efficient reaching and locomotor actions," in *Artificial Neural Networks and Machine Learning–ICANN 2012*. Springer, 2012, pp. 322–329.