Application of the Visuo-Oculomotor Transformation to Ballistic and Visually-Guided Eye Movements

Marco Antonelli  Angel J. Duran  Angel P. Del Pobil

Abstract—Active eye movements can be exploited to learn the oculomotor command that allows a humanoid head to gaze at a visual target. In the field of adaptive robotics, this transformation is commonly encoded by means of neural networks that are trained using ballistic eye movements.

In this work, we encode the visuo-oculomotor transformation using radial basis function networks and we derive the equations to use such a transformation in visual servo control tasks. We study how the choice of the input (monocular vs binocular) influences the execution of ballistic and visually-guided eye movements.

Analytic analysis and experimental results on a simulated model of the robot show that the proposed framework can be employed to execute both kinds of movements.

I. INTRODUCTION

Human and primate’s eyes are controlled by concurrent mechanisms that can be either voluntary (saccade and smooth pursuit) or not (vestibular-oculomotor reflex, optokinetic reflex and vergence). This work focuses on the implementation of voluntary eye movements in a robotic head.

Saccades are fast eye movements that are executed to gaze at a specified target. While, in general, the target can be provided both by auditory and visual stimuli, herein we refer only to the visual ones. Once a saccadic target is detected, the oculomotor system triggers an eye movement to bring it in the center of the visual field. The movement is ballistic, that is, it is not affected by the visual perception during the motion. In the field of control theory, this movement is considered as open loop and its implementation requires a precise model of the plant, herein the oculomotor system.

Smooth pursuits are relatively slow eye movements that are used to actively track a moving target. The visual velocity of the target is used as feedback signal to modify the velocity of the eyes, so that it is considered a closed-loop control law. The implementation of these movements requires to know the relationship between eye movement and optic flow and to predict the motion of the target.

In neuroscience and bio-inspired robotics, the model of the plant is built up on the integration of the sensory cues, and learning such a model is known as sensorimotor transformation problem. In previous works we presented a framework that allows the robot to learn the internal model of the robot and build an implicit sensorimotor representation of the peripersonal space [1], [2]. The framework grounds on radial basis function networks (RBFN) to encode and update the sensorimotor transformation.

The visuo-oculomotor transformation is learned during the visual exploration of the environment. The on-line learning strategy is the following: the robot localizes a visual stimulus, triggers a saccade toward it and then uses the new visual location of the stimulus to update the internal model [3].

In this work, we exploit the visuo-oculomotor transformation to control the eye velocity to perform smooth pursuit movements. In order to do that, we compute the Jacobian of the transformation to obtain the relationship between the velocity of the eyes and the subsequent motion of the image. This relationship, which is the basis of the visual servo control, can be used to predict the apparent motion of the image due to eye motion and vice-versa.

Moreover, we tested two approaches to encode the visuo-oculomotor transformation. The first one takes as input the visual position of the stimulus on a single camera and retrieves an updating of the eye position [4], [5], [6]. The second approach employs binocular information, that is, it takes as input the visual location of the target in both the left and right images [7], [8], [9], [10], [1].

The binocular approach is more precise than the monocular one because it implicitly encodes the distance of the target. Indeed, depth information is required because the nodal point of the camera does not lies on the center of rotation of the motors [11]. On the other hand, the binocular approach requires the correspondence problem, which is an ill-posed problem, to be solved [12].
In this work, we consider the correspondence problem has already been solved and we compare the performance of the two approaches in the execution of a ballistic and visually-guided eye movements. Experimental results on a simulated model of the robot shows that the RBFN framework is able to perform accurate saccades and smooth pursuits movements.

The remainder of this paper is organized as follows. In Section II we describe the model of the robotic head that we use to study the visuo-oculomotor transformations. These transformations, which are introduced in Section III, are encoded by RBFNs. Radial basis function networks, the learning algorithm and the computation of the Jacobian of the networks are described in Section IV. Section V reviews the equations of the image-based visual control using the proposed transformations while Section VI describes the experiments and the achieved results. Finally, in Section VII we compare the proposed approach with related works and we discuss the main results of this study.

II. ROBOTIC HEAD MODEL

This section describes the geometrical model of the robotic head of the humanoid torso of the University Jaume I, namely Tombatossals (Fig. 1). The robotic head mounts two cameras with a resolution of $1024 \times 768$ pixels that can acquire color images at 30 Hz. The pixel size is $4.65 \mu m$ and the focal length was set to $5 \ mm$ to obtain a wide field of view. The cameras can actively move by means of a common tilt and two independent pan motors. The baseline between the cameras is $270 \ mm$. The center of rotation of the motors (tilt, left pan and right pan) do not lie on the nodal point of the camera lens, so that, their rotation produces both rotation and translation of the nodal point (see Fig. 2). Due to the translation, the visuo-oculomotor transformation depends on the distance of the target. The misalignment between the nodal point and the center of rotation depends on the focal length of the camera, is virtually present in every robotic system and also in humans’ eyes [11].

Through this paper, we describe the proposed framework using the two dimensional model of the robot that is sketched in Fig. 2. However, the framework can be easily generalized with the full 3D model. The notation is the following:

- $u_l$ and $u_r$ represent the projection of the visual target on the left and right images, respectively;
- $\theta_l$ and $\theta_r$ represent the angular position of the left and right cameras;
- $\gamma_l$ and $\gamma_r$ represent the angular position of the left and right cameras that allow the robot to gaze at the target.

That is, when $\theta_l = \gamma_l$ and $\theta_r = \gamma_r$, the visual position of the stimulus is zero.

Moreover we also use the matrix-notation: $u = [u_l, u_r]^T$, $\theta = [\theta_l, \theta_r]^T$ and $\gamma = [\gamma_l, \gamma_r]^T$.

Using this notation, the visuo-oculomotor transformation can be mathematically defined as

$$\gamma = T_{V\rightarrow O}(u, \theta)$$  \hspace{1cm} (1)

III. VISU-OCLUDOMOTOR TRANSFORMATION

In the proposed framework, the spatial position of an object is maintained by a spherical-like coordinate system that is centered in the head of the robot (head-centered f.o.r.). In particular, the head-centered coordinate system is defined by the version and vergence angles:

$$\text{version} = \frac{\gamma_l + \gamma_r}{2}, \text{vergence} = \gamma_l - \gamma_r. \hspace{1cm} (2)$$

Using this spherical-like coordinate system, the robot implicitly encodes the spatial position of the gazed object by means of its sensory-motor cues [1].

Shifting the gaze at a target requires to integrate the current position of the eyes with the visual position of the stimulus, which provides just a local information of the target location (retinotopic-centered f.o.r.).

Such an integrative process can be conducted independently for each eye (monocular approach) or jointly (binocular approach). The monocular approach is simpler but it is less precise because it does not consider the distance of the target, which is necessary to compute the exact gaze shift. The error that is made by omitting distance information depends on the distance of the target itself. Its effect is stronger for closer points while it is negligible for distant points. It can be formulated as follow:

$$\gamma_x = T_{V\rightarrow O}^{M}(u_x, \theta_x) \hspace{1cm} (3)$$

where $x$ can be $l$ (left) or $r$ (right) depending on which eye we are using.

Conversely, the binocular approach implicitly encodes the distance in the disparity of the visual target. The binocular approach can be formulated as follow:

$$\gamma_x = T_{V\rightarrow O}^{B}(u_x, u_r, \theta_l, \theta_r) \hspace{1cm} (4)$$
IV. RADIAL BASIS FUNCTION NETWORKS

Learning the visuo-oculomotor transformation \((TV \rightarrow O)\) can be treated as a function approximation problem. Radial basis function networks can potentially approximate any function with the desired precision, so they are especially suitable for encoding sensorimotor transformations [13, 14].

Basis function networks are three-layer feed forward neural networks whose hidden units \(h_i\) perform a non-linear transformation of the input data, whereas the output \((y)\) is computed as a linear combination \((w_i)\) of the hidden units:

\[
y = \sum_{i=1}^{n} w_i \cdot h_i(x) = w^T h(x).
\]

Learning in the context of the radial basis function networks can be divided into two phases. An unsupervised phase sets the parameters of the network, such as the number of hidden units or the position of their centers, whereas a supervised phase adjusts the weights. In the proposed framework, we employ fixed centers, whose receptive fields do not move according to the input data.

We model the activation of the hidden neurons by using Gaussian functions. Each unit is characterized by its center of activation \((c_i)\), whereas the spread of the activation \((\Sigma)\) is equal for every unit:

\[
h_i(x) = h(||x - c_i||) = e^{-(x - c_i)^T \Sigma^{-1} (x - c_i)}
\]

Using this setup, the learning process consists in finding the weights that better approximate the sensorimotor transformation. Given a set of \(m\) input-output samples of the target function, the weights of the network can be calculated by minimizing a cost function \(E\), defined as

\[
E = \frac{1}{2} \sum_{d=1}^{m} (\hat{y}_j - y_j)^2 = \frac{1}{2} \sum_{d=1}^{m} (e_d)^2
\]

where the error \(e_d\) is the difference between the actual output of the transformation \((\hat{y}_j)\) and that calculated by the network \((y_j)\) for the \(j\)-th input \((x_j)\).

We trained the visuo-oculomotor transformation using an on-line algorithm that adapts the map each time a new training point is available. In previous work [1], [2], [3] and in other bio-inspired studies [13] RBFNs were trained using the delta rule. In this work, we preferred to employ the recursive least square algorithm [15], because it is more suitable to be implemented on a real robot. Indeed, it converges faster than the delta rule, so saving robot interactions and it does not require to set empirical parameters such as the learning rate, which is a possible factor of instability [16]. At each step, the algorithm updated the state matrix \(P\) and the weights \(w\), using the following equations:

\[
\delta = (1 + h^T \cdot P \cdot h)^{-1}
\]

\[
w = w + \delta \cdot e \cdot h^T \cdot P
\]

\[
P = P - \delta \cdot h \cdot h^T \cdot P
\]

where \(e\) is the error of the network. At the start up of the system, the matrix \(P\) was initialized to \(10^3 \cdot I\), where \(I\) is the identity matrix, whereas \(w\) was initialized to zero.

The implementation of visual servo control requires to compute the partial derivative of the RBFN with respect to the input \(x\). Considering that the matrix \(\Sigma\) is symmetric, the Jacobian is given by the following equation:

\[
J(x) = \sum_{i=1}^{n} w_i \cdot \frac{\partial h_i(\cdot)}{\partial x} = -2 \sum_{i=1}^{n} w_i \cdot h_i(\cdot) \cdot (x - c_i)^T \cdot \Sigma^{-1}
\]

V. VISUAL SERVO CONTROL WITH RBFNS

The aim of the visual servo control is to drive the velocity of the nodal point to bring the target in the center of the visual field following a certain trajectory. In general, the variation of the visual position of the target \((\hat{u})\) depends on its own movement \((\dot{u})\) and on the motion of the nodal point \((\nabla H)\), which in turn depends on the velocity of the motors. So that, the visual velocity of the target can be written as:

\[
\dot{u} = J_f \cdot \dot{\theta} + \dot{\hat{t}}.
\]

In equation (10), \(J_f\) is feature Jacobian matrix that links the image motion to the movement of the motors [17], [18].

In this section we provide the equations for the monocular case, which can be easily extended to the binocular approach. The visuo-oculomotor transformation encoded by the RBFNs can be written as:

\[
\gamma = T_{V \rightarrow O}^h (u, \theta) = \sum_{i=1}^{n} w_i \cdot h_i(\theta, u)
\]

By deriving equation (11) we obtain:

\[
\dot{\gamma} = \sum_{i=1}^{n} w_i \cdot \left( \frac{\partial h_i(\cdot)}{\partial \theta} \cdot \dot{\theta} + \frac{\partial h_i(\cdot)}{\partial u} \cdot \dot{u} \right)
\]

\[
= J_{\theta} \cdot \dot{\theta} + J_u \cdot \dot{u}
\]

where \(J_{\theta}\) and \(J_u\) are the Jacobian of \(T_{V \rightarrow O}\) with respect \(\theta\) and \(u\), respectively. By substituting Eq. (12) into Eq. (10) we obtain:

\[
\dot{u} = -J_u^{-1} \cdot J_{\theta} \cdot \dot{\theta} + J_{\theta}^{-1} \cdot \dot{\gamma}
\]

so that, \(J_f = J_u^{-1} \cdot J_{\theta}\).

Once we obtained the feature Jacobian matrix we can use it as in the standard theory of the visual servo control. For instance, if we design the control law to decrease exponentially the error \((\dot{u} = -\lambda \cdot u)\), the velocity of the camera can be computed using the following equation:

\[
\dot{\theta} = -\lambda \cdot J_{\theta}^{-1} \cdot u + J_u^{-1} \cdot \dot{\gamma}
\]

The advantage of using the RBFNs to compute feature Jacobian matrix is that we do not need to explicitly calibrate the robotic head. That is, we do not need to rectify the images and to find the intrinsic and extrinsic parameters of the cameras.
VI. EXPERIMENTAL RESULTS

In this study, we conducted three experiments to compare the monocular approach with the binocular one. The first experiment tests the performance of $T_{V \rightarrow O}$ with ballistic eye movements. The second experiment employs the same transformations to predict the apparent motion of the target in the image due to the motion of the camera. Finally, the third experiment tests the two approaches on a visual servo control task.

For these experiments, we implemented a RBFN for each approach. We encoded each input signal with a fixed number of Gaussian units that were distributed uniformly in the input space. We employed 6 units for the visual input and 4 for proprioceptive cues. The standard deviation of the Gaussian was set to three times the distance between two consecutive centers.

Using this setting, the monocular transformation has two inputs and $24(6 \times 4)$ hidden units while the binocular one has four inputs and $570(6 \times 6 \times 4 \times 4)$ hidden units.

The two RBFNs were trained on the same dataset using the recursive least square algorithm. The dataset were obtained as follow. We placed $390(30 \times 13)$ visual targets uniformly distributed in the distance/version space (black points of Fig. 3). Then, we generated 200 fixation points (red circles of Fig. 3). From each fixation point the robot executed a saccade to every visible target point. In this way we obtained $20495$ input-output samples that were used to train the visuo-oculomotor transformations.

In general, the visual system has not direct access to the training points (pairs of visual position of the stimulus and associated gaze movement), which is the issue of the lacking of a teacher. However, this problem is not the focus of this paper and it can be solved in several ways such as feed-back learning [19], distal teacher [20] or linear approximation [3].

A. Ballistic movement

In this experiment we generated 20969 input-output samples to test the performance of the networks. The dataset was created similarly to that used to train the networks.

Moreover, we compared the results of the two transformations with those obtained using an approximated model of the robot. The approximated model does not take into account the translation between the center of rotation of the camera and the nodal point of the lens. This model allows us to understand how much this translation is relevant in the control of eye movements.

The results of the experiment obtained with the left camera are shown in Table I. The table shows the mean value and the standard deviation of the visual position of the stimulus after the saccadic movement. The approximated model provides the worst performance, because omitting the translation of the nodal point is similar to assume that the target point is located at an infinite distance. The monocular transformation outperforms the approximated model because it implicitly learns the average distance of the stimuli. Finally, the binocular transformation provides the best results and the standard deviation of the visual position of target stimulus is about 2 pixels on an image with a resolution of 1024 pixels. Indeed, it implicitly encodes the distance of each point, and not just the mean distance of the distribution.

<table>
<thead>
<tr>
<th>Transf.</th>
<th>Input</th>
<th>Units</th>
<th>Error: $\mu \pm \sigma$ [pixels]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{V \rightarrow O}$</td>
<td>$u_{1}, \theta_{1}$</td>
<td>$24(6 \times 4)$</td>
<td>$-0.25 \pm 10.49$</td>
</tr>
<tr>
<td>$T_{V \rightarrow O}$</td>
<td>$u_{1}, \phi_{1}$</td>
<td>$24(6 \times 4)$</td>
<td>$-0.11 \pm 10.07$</td>
</tr>
<tr>
<td>$T_{V \rightarrow O}$</td>
<td>$u_{1}, \phi_{1}, \theta_{r}$</td>
<td>$576(6 \times 6 \times 4 \times 4)$</td>
<td>$-0.03 \pm 2.14$</td>
</tr>
</tbody>
</table>

In our framework the target position is encoded by the gaze direction thus we need to perform accurate saccades. From this perspective, the results suggest that the binocular transformation is more suitable to implicitly encode the position of the target and should be preferred in such a task. In order to verify that, we computed the fixation error of the two approaches. In the monocular case, the saccades of the left and right eyes are computed separately by two independent transformations. On the other hand, the binocular transformation produces coordinated saccades.

The error is computed as Euclidean distance between the Cartesian position of the target and the fixation point. The mean error on the testing set is 8.5 mm for the binocular approach and 47.6 mm for the monocular one. The histograms of the error are showed in figure 4. In order to make clearer the figure we cut the errors greater than 10 cm. From the histogram it is possible to appreciate that the binocular approach provides a more coherent representation of the space.

B. Estimation of image motion

We placed a stimulus in the center of the visual field and we started to move the left camera with a constant velocity of $10^\circ/s$. At the end of the simulation (2 s), the target appears almost at the border of the visible space (around 430 pixels).

In order to evaluate the performance of transformations, we measured the root-mean square error (RMSE) between
the actual velocity of the stimulus and the predicted image motion. The results are reported in Fig. 5 for a target placed at a different distance in a range between 0.5 m and 2 m. As expected, the Jacobian of the approximated model improves its performance with the increasing distance of the target because the translational component of the velocity became smaller. The monocular transformation provides its best result around 1 m, which is the mean distance of the training points. Finally, the binocular transformation works well independently on the distance.

These effects can also be observed in the velocity profile predicted by the Jacobian of the visuo-oculomotor transformations (Fig. 8). Indeed, the profiles provided by the approximated model and by the monocular transformation are independent from the distance of the target. Moreover, the result of the approximated model is biased by the lack of the translational velocity. This effect is reduced with the distance.

C. Visual servo control

Finally, we tested the sensorimotor transformations in a visual servo control task which consists in tracking a moving stimulus. At time zero, the stimulus begins to move on the $x-$axis with a constant velocity of $0.50 m/s$.

Fig. 4. Histogram of the gazing error. The error is computed as euclidean distance between the target point and the fixation point after the saccade.

Fig. 5. Root mean square errors for a target placed at a different distances.

The velocity of the camera was controlled to reduce exponentially the visual error, as explained in Section V. The position of the stimulus in the image as a function of time is shown in Fig. 6. At the time zero the camera is stopped and the stimulus starts to move. The visual system detects the motion of the stimulus and controls the velocity of the camera to reduce the visual error. After 0.8 s, the camera and the stimulus move with the same velocity.

Fig. 6. Trajectory of visual stimulus in the image for a target placed at 50 cm in front of the robot.

VII. DISCUSSION

This work is part of a sensorimotor framework that creates an implicit representation of the environment and simultaneously learns the sensorimotor associations among vision, oculomotor system and arm-motor system by means of reaching and gazing [1], [2], [3].
In this paper we have focused on the learning of the visuo-oculomotor transformation. With respect to the previous works, we have employed a faster learning algorithm (recursive least square vs delta rule). Moreover, we have compared two different approaches that can be found in the literature to represent such a transformation: the monocular system [4], [6] and the binocular one [8], [9], [2].

The achieved results show that both the binocular and the monocular approach provided good results in closed loop eye movements. Given that the monocular approach does not require to solve the correspondence problem, it could be preferred for active tracking. On the other hand, the
computation of the saccadic movement is more precise with the binocular approach and should be preferred to maintain an implicit representation of the space.

In previous related works [4], [6], [8], [9], [2], even though the visuo-oculomotor transformation was learned using different approaches (inverse forward model vs direct inverse model) and using different encoding (look-up table vs neural networks), it was always learned using saccadic movements. Herein, the transformation learned by means of saccadic movement is exploited to track a moving target (smooth pursuits). In the robotic literature, works related to smooth pursuits usually focus on the prediction of the target motion. On the other hand, the relationship between camera velocity and image velocity is provided by an empirical proportional gain [19], [21]. Conversely, we obtain such a relationship by means of the Jacobian of the neural network and we validated the method in a simulated environment. The use of the Jacobian of the RBFN in robotics is not novel, but it was mainly used in eye-arm coordination [22], [23].

Finally, our framework uses RBFNs to encode the sensorimotor transformations. RBFNs were chosen to model the gain field effect that is typical of the populations of neurons in posterior parietal cortex, which are involved in the sensorimotor transformation task [13], [14], [24], [25]. The approach proposed in this work can be implemented with other feed-forward neural networks that are not necessarily RBFNs, such as the locally weighted projection regression network (LWPR) [26]. However, given that the input space is small and that the implemented RBFN approximates correctly the visuo-oculomotor transformation, we preferred to avoid the use of complex networks that require to tune a considerable number of parameters.

The RBFN was trained using the recursive least square algorithm that is also employed in state-of-the-art algorithms, such as LWPR [26]. This learning strategy allows the system to speed up its convergence time and makes it suitable for the implementation on the real robot.

Conclusions

This work is part of a sensorimotor framework that links the visual position of the target with the gaze direction. We tested two approaches for encoding the visuo-oculomotor transformation that are based on the monocular and binocular information, respectively. In both approaches the transformation is learned by means of ballistic movements of the cameras and is subsequently employed to track a moving target. Simulation results show that the proposed approach approximates accurately the behavior of the desired visual servo controller. Moreover, we have verified that a joint control of the eye movements is more suitable to maintain an implicit representation of space. Finally, in this work we have provided a common framework to represent the surrounding space, perform saccadic movements and track moving targets.

REFERENCES


