2nd Workshop Toward Robot Avatars, IEEE International Conference on Robotics and Automation (ICRA), London, UK, 2023.

Gaze teleoperation for the iCub3 humanoid robot

Stefano Dafarra[1], Daniele Pucci[1,2]

***Abstract*— We present an algorithm for controlling the motion of the bulbs of a humanoid robot in a teleoperation system. The operator controls the bulbs by simply gazing inside the VR headset. This additional control allows the operator to have a more immersive experience while the robot appears more human-like. We tested the algorithm while on the stage of the "We Make Future" Festival in front of about 2000 spectators. The operator and the robot were about 400 km apart.**

I. INTRODUCTION

A telexistence system transfers the skills of the human operator to a robotic avatar [1]. Intuitiveness is a key feature of the system. Humanoid robot avatars show great potential for existing and future applications of teleoperation systems: the robot's human-likeness feature increases its acceptability, its social interaction performances, and the clarity of its intentions [2].

Humanoid robots thus represent an optimal starting point for a platform to embody humans in terms of locomotion, manipulation, verbal and non-verbal interaction, allowing an operator to have direct control over the whole-body of the robot [3], [4], [5].

The immersivity of a teleoperation system can be further enhanced by allowing the operator to actively control the robot's face expressions. One possible approach consists in using a classical display over which the operator's face expressions are retargeted [6]. In the case of a humanoid robot, the motion of the robot bulbs can be directly controlled[7]. This paper presents an application of gaze control on the iCub3 humanoid robot [8], tested while on the stage of the "We Make Future" Festival[1] in front of 2000 spectators.

II. BACKGROUND

A. Notation

- Vector and matrices are expressed in bold symbols.
- The transpose operator is denoted by $(\cdot)^\top$.
- The superscript $(\cdot)^*$ indicates desired values.
- $\boldsymbol{e}_i$ is the canonical base in $\mathbb{R}^n$.
- The L2-norm of a vector $\boldsymbol{v} \in \mathbb{R}^n$ is denoted by $\|\boldsymbol{v}\|$.
- ${}^A\boldsymbol{R}_B \in SO(3)$ and ${}^A\boldsymbol{H}_B \in SE(3)$ denote the rotation and transformation matrices which transform a vector expressed in the B frame, ${}^B\boldsymbol{x}$, into a vector expressed in the A frame, ${}^A\boldsymbol{x}$. $\boldsymbol{R}^j(\alpha)$ with $j \in \{x, y, z\}, \alpha \in \mathbb{R}$, is a rotation around a cartesian axis of a given angle α.

[1]Artificial and Mechanical Intelligence at Istituto Italiano di Tecnologia, Center for Robotics and Intelligent Systems, Genova, Italy. name.surname@iit.it

[2]Machine Learning and Optimisation, The University of Manchester, Manchester, United Kingdom.

[1]https://en.wemakefuture.it/

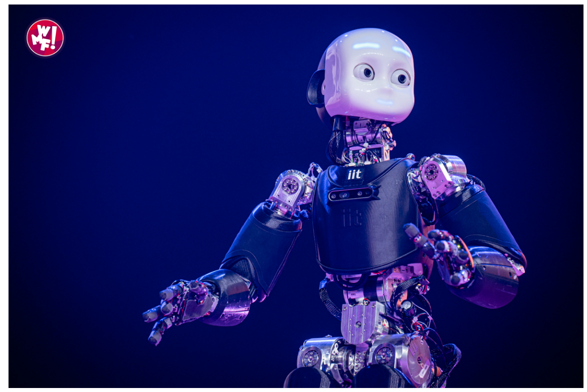

Fig. 1. iCub3 while on the stage of the We Make Future Festival.

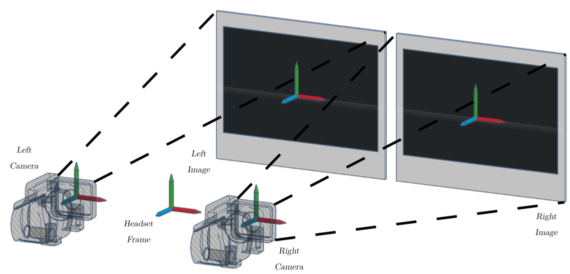

Fig. 2. A simplified representation of the stereo rendering.

B. iCub3 Eyes Kinematics

The iCub3 humanoid robot, depicted in Fig. 1, possesses two RGB cameras placed in the bulbs. The two bulbs are coupled in the tilting direction (i.e. up-down direction), while they are independent in the axial plane. The tilt angle is indicated by $\theta_t \in \mathbb{R}$, positive when looking upwards. We define the position of the two bulbs in the axial direction with $\theta_{a,l}, \theta_{a,r} \in \mathbb{R}$, positive when looking left. In this context, it is possible for the eyes to look in opposite directions. This behavior would appear unnatural and unaesthetic. Hence, the robot low-level control does not allow to control directly $\theta_{a,l}$ and $\theta_{a,r}$, but rather the vergence $\theta_{vg} \in \mathbb{R}^+$ and version $\theta_{vs} \in \mathbb{R}$ angles, with θ_{vg} non-negative. They are as follows:

$$\theta_{vg} = \theta_{a,r} - \theta_{a,l}, \quad \theta_{vs} = -\frac{1}{2}\left(\theta_{a,r} + \theta_{a,l}\right). \tag{1}$$

III. STEREO CAMERA VISUALIZATION

The video stream generated by the robot eyes is visualized in a Virtual Reality (VR) environment. The two images are visualized via two different rectangles, each visible only to the corresponding eye. In other words, the operator left eye sees only the image captured by the left camera.

Fig. 2 depicts a representation of the system used for visualization. The *headset frame* is placed in the middle of the headset with the y axis pointing up and the z axis pointing backward. We place the origin of the camera frames along the x axis, symmetric with respect to the origin, and at a distance equal to the robot interpupillary distance p_x. Hence, we have ${}^h\boldsymbol{p}_l = \begin{bmatrix}-p_x/2 & 0 & 0\end{bmatrix}^\top$ and ${}^h\boldsymbol{p}_r = \begin{bmatrix}p_x/2 & 0 & 0\end{bmatrix}^\top$. Then, the two rectangles are placed along the cameras' z axis, at a distance dependent on the camera focal length. In particular, we have $wf = Wp_z$ where w is the width of the image in the VR space, f is the focal length expressed in pixel quantities, W is the pixel width of the image, and p_z is the position of the image along the depth direction.

We can rotate the fictitious cameras around the x and y axis. The corresponding angles are defined as $\phi_{t,e}, \phi_{a,e} \in \mathbb{R}$, respectively, with e a placeholder for left and right, thus moving the rectangles in the 3D VR space accordingly. The relative rotation between the headset and the eye image frame is ${}^h\boldsymbol{R}_e = \boldsymbol{R}^x(\phi_{t,e})\boldsymbol{R}^y(\phi_{a,e})$. Thus, the relative pose between the headset and the e image frame is

$$ {}^h\boldsymbol{H}_e = \begin{bmatrix} {}^h\boldsymbol{R}_e & {}^h\boldsymbol{p}_e + {}^h\boldsymbol{R}_e \begin{bmatrix}0 & 0 & p_z\end{bmatrix}^\top \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{2} $$

IV. Gaze control

The gaze control iterates on the following steps:

1) Measure the operator gaze.
2) Compute the desired eye velocities.
3) Apply the desired eye velocity to the robot.
4) Update the VR layer positions according to the robot eye angles.

First, we measure the operator gaze in the headset frame and we express it in the image frame through ${}^h\boldsymbol{H}_e$. Define ${}^g\boldsymbol{o}$ and ${}^g\boldsymbol{d}$ as the gaze origin and direction expressed in the image frame coordinates, and ${}^e\boldsymbol{p}$ the intersection point. This can be computed as follows. Consider a generic point $\boldsymbol{p}$ on the gaze line, $\boldsymbol{p} = {}^g\boldsymbol{o} + \gamma {}^g\boldsymbol{d}$. In order to find ${}^e\boldsymbol{p}$, it is sufficient to compute γ such that the z coordinate of $\boldsymbol{p}$ is null, i.e.

$$ \gamma = -\frac{\boldsymbol{e}_3^\top {}^g\boldsymbol{o}}{\boldsymbol{e}_3^\top {}^g\boldsymbol{d}} \tag{3} $$

Given the intersection of the gaze with the VR image ${}^e\boldsymbol{p}$, we apply a dead zone as follows

$$ {}^{\tilde{e}}\boldsymbol{p} = \max\left(1.0 - \frac{d}{\|{}^e\boldsymbol{p}\|}, 0\right) {}^e\boldsymbol{p} \tag{4} $$

where $d \in \mathbb{R}$ is a user-chosen dead zone. We consider two different levels to activate and deactivate the gaze control. This is necessary because the gaze measurement is noisy, and continuous motions of the VR images can cause motion sickness. Hence, we keep the images perfectly still when the gaze is within a circle around the image frame origin.

The desired eye velocities are computed by using the small-angle approximation (i.e. $sin(x) \approx x$), with the goal of having the operator gaze at the center of the image:

$$ \dot{\phi}^*_{t,e} = k\frac{\boldsymbol{e}_2^\top {}^{\tilde{e}}\boldsymbol{p}}{|p_z|}, \quad \dot{\phi}^*_{a,e} = -k\frac{\boldsymbol{e}_1^\top {}^{\tilde{e}}\boldsymbol{p}}{|p_z|} \tag{5} $$

Fig. 3. The operator controlling the robot of Fig. 1. On the top left, it is possible to see the VR visualization displayed on a TV.

where $k \in \mathbb{R}$ is a user-defined gain. Assume the intersection to be positive along the y axis on the image. This means that the operator is looking up, thus the tilt velocity needs to be positive. On the other hand, if the operator is looking right, the axial velocity needs to have a negative sign to move in the same direction.

The desired robot eye velocities are computed following Eq. (1). In particular:

$$ \dot{\theta}^*_t = \frac{1}{2}(\dot{\phi}^*_{t,r} + \dot{\phi}^*_{t,l}) \tag{6} $$

$$ \dot{\theta}^*_{vs} = -\frac{1}{2}\left(\dot{\phi}^*_{a,r} + \dot{\phi}^*_{a,l}\right) \tag{7} $$

$$ \dot{\theta}^*_{vg} = \dot{\phi}^*_{a,r} - \dot{\phi}^*_{a,l}. \tag{8} $$

They are saturated to consider the eyes' kinematic limits. This is achieved by applying the method presented in [9, Sec. III.B]. The saturated desired robot velocities are commanded to the robot. Then, by accessing the robot eye encoder values, we update the desired orientation of the VR images. This mechanism has the following effect. If the operator stares at an object that is not in the center of the image, the gaze control moves the robot's eyes to bring that object to the center of the image, while keeping its position constant in the VR space.

V. Results

We validated the gaze control on the iCub3 humanoid robot while on the stage of the "We Make Future" Festival, Fig. 1 The video of the demonstration: `https://youtu.be/FkBXzLaO7W0?t=4027`. Via an HTC Vive Pro Eye headset[2], a remote operator controls the gaze of the robot while being about 400 Km away. From Fig. 3, it is possible to spot the visualization of the images in a pair of rectangles. The rectangles are rotated since the operator is looking left, and the eyes of the robot move accordingly, as in Fig. 1. The code is available online[3].

[2] `https://www.vive.com/us/product/vive-pro-eye/overview/`

[3] Gaze control: `https://github.com/robotology/walking-teleoperation/tree/v1.3.3/modules/SRanipal_module`, VR visualization `https://github.com/ami-iit/yarp-device-openxrheadset`

REFERENCES

[1] S. Tachi, "Telexistence," in *Virtual Realities*. Springer, 2015, pp. 229–259.

[2] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 301–308.

[3] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *IEEE Transactions on Robotics (Accepted)*, 2023.

[4] K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, and D. Pucci, "Whole-body geometric retargeting for humanoid robots," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 679–686.

[5] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot," *IEEE Robotics & Automation Magazine*, vol. 26, no. 4, pp. 73–82, 2019.

[6] A. Rochow, M. Schwarz, M. Schreiber, and S. Behnke, "Vr facial animation for immersive telepresence avatars," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2167–2174.

[7] R. Cambuzat, F. Elisei, G. Bailly, O. Simonin, and A. Spalanzani, "Immersive teleoperation of the eye gaze of social robots-assessing gaze-contingent control of vergence, yaw and pitch of robotic eyes," in *ISR 2018; 50th International Symposium on Robotics*. VDE, 2018, pp. 1–8.

[8] S. Dafarra, K. Darvish, R. Grieco, G. Milani, U. Pattacini, L. Rapetti, G. Romualdi, M. Salvi, A. Scalzo, I. Sorrentino, *et al.*, "icub3 avatar system," *arXiv preprint arXiv:2203.06972*, 2022.

[9] L. Rapetti, Y. Tirupachuri, K. Darvish, S. Dafarra, G. Nava, C. Latella, and D. Pucci, "Model-based real-time motion tracking using dynamical inverse kinematics," *Algorithms*, vol. 13, no. 10, p. 266, 2020.