Hands-Free: A Robot Augmented Reality Teleoperation System

Cristina Nuzzi*, Stefano Ghidini, Roberto Pagani, Simone Pasinetti and Giovanna Sansoni

***Abstract*— This electronic document is a live template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.**

I. INTRODUCTION

Despite advances in robotic perception are increasing autonomous capabilities, the human intelligence is still considered a necessity in unstructured or not predictable environments. Typical scenarios concern the detection of random shape objects, manipulation, or custom robot motion. In such context, human and robots must achieve mutual human-robot interaction (HRI) [1].

HRI can be both physical (pHRI) or not, depending on the assigned task. For example, when the robot is constrained in a dangerous environment or must handle hazardous materials, pHRI is not recommended. In this cases, robot teleoperation may be necessary. A teleoperation system concerns with the exploration and exploitation of spaces which do not allow user presence, thus, the operator acts by remotely move the robot [2]. A plenty of human-machine interfaces for teleoperation have been developed considering a mechanical interface, this includes exoskeleton [3] or gloves [4]. Such systems are particularly helpful to achieve bilateral teleoperation [5], where they can transmit or reflect back to the user reaction forces from the task being performed. In this case, a high perception with complete haptic feedback [6] is required. Other interfaces includes mouse, switchbox, keyboard, touch-screen and joystick, which is usually a better control device than others because the operators can identify better with the task [7]. Electromyography (EMG) is widely used to implement teleoperation systems by mean of muscular activity signals [8], [9]. However, as reminded recently in [10], EMG could be affected by difficulties in processing EMG signals for amplitude and spectral analysis, reducing their efficiency for many applications. Moreover, all the interfaces described still act by contact, hindering the movement of the operator or cause him to act through unnatural movements.

Therefore, if bilater interaction is not required, a vision-based interface is preferable. A vision-based interface does not require physical contact with external devices such as cables, connectors and objects outside of the user working area. This grants a more natural and intuitive interaction, which is reflected on the task performance: as shown in [11], the accuracy of object gripping tasks is improved by mean of a contactless vision-based robot teleoperation method, while in [12] a stereo vision system improved the performance of a mobile robot teleoperation application.

Furthermore, if a vision-interface is integrated with virtual and augmented reality techniques, it translates in a greater level of immersion for the user. Such techniques are used to enhance the feedback information; in fact, the operator feels like being physically present in the remote environment, enhancing the immersion level. The notion of immersion is one of the most important reasons for using virtual and augmented reality [7]. An augmented reality system for teleoperation based on the Leap Motion (LM) controller is presented in [13]. For its application domain, the LM controller is considered an accurate sensor [14], however it is limited to a relatively small measuring distance if compared with other sensors. In this sense, the LM controller introduces spatial constraints that clash with the previously stated concept of the high level user immersion.

For these reasons, we present a novel robot augmented reality teleoperation system that exploits RGB cameras, which provide greater measuring distance if compared with the LM controller. A ROS-based framework has been developed to provide hand tracking and hand-gesture recognition features, exploiting the OpenPose software [15], [16] based on the Deep Learning framework Caffe [17]. This, in combination with the ease of availability of an RGB camera, lead the framework to be strongly open-source oriented and highly replicable on all the ROS-based platform. The proposed system includes: neural network for hand−gesture recognition (Section III), a rigorous procedure for robot workspace calibration and a mapping policy between the coordinate system of the user and the robot (Section III). Different experiments were performed on a *Sawyer (Rethink Robotics)* industrial collaborative robot to evaluate repeatability and accuracy of the proposed system. Section IV reports the results.

II. WORKSPACE CALIBRATION AND MAPPING

Our set-up is composed of two workspaces: the *user workspace* and the *robot workspace*. Cartesian points in the user workspace, which can be reached by the user hand and are correctly viewed by the camera, correspond to precise robot end-effector Cartesian points in the robot workspace. To obtain a mapping between the hand positions and the

This work was not supported by any organization.

Cristina Nuzzi, Roberto Pagani, Simone Pasinetti and Giovanna Sansoni are members of the Department of Mechanical and Industrial Engineering at University of Brescia, Via Branze 38, 25123 Brescia, Italy.

Stefano Ghidini is a member of the STIIMA-CNR, Via Alfonso Corti 12, 20133 Milan, Italy. He is also a member of the Department of Mechanical and Industrial Engineering at University of Brescia, Via Branze 38, 25123 Brescia, Italy.

* Corresponding author, e-mail: c.nuzzi@unibs.it

robot end-effector positions, it is necessary to perform a set of calibration procedures described in the following sections.

A. User Workspace Calibration

In the user workspace an RGB camera is used to recognize the hand skeleton in real-time. Therefore, it is necessary to properly calibrate the camera relative to the user-defined reference system. This procedure is called *camera calibration*, and can be easily realized following standard procedures, such as the one detailed in [18]. The projection mapping for a generic point $\mathbf{P}_{0,C} = (u, v)$ in the camera image plane with reference frame C to its corresponding real world coordinate point $\mathbf{P}_{0,H} = (x, y, z)$ in reference frame H is defined by the following Equation. Homogeneous coordinates are required:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R}|\mathbf{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{1}$$

However, since we are looking for the point position $\mathbf{P}_H$ in the frame H by back-projecting a 2D point to 3D it is necessary to invert Equation 1:

$$\mathbf{P}_{0,H} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left(s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \mathbf{K}^{-1} - \mathbf{t} \right) \mathbf{R}^{-1} \tag{2}$$

In the equations above, the scalar $s \in \mathbb{R}$ is the scale factor of the image, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera matrix containing the intrinsic parameters of the camera such as focal length and optical center obtained through the calibration procedure, $[\mathbf{R}|\mathbf{t}] \in \mathbb{R}^{4\times3}$ is the rigid transformation matrix containing the exstrinsic parameters for rotation ($\mathbf{R} \in \mathbb{R}^{3\times3}$) and translation ($\mathbf{t} \in \mathbb{R}^3$) of the camera reference frame C relative to the calibration master reference frame H. To obtain matrix $\mathbf{K}$ it is necessary to perform a calibration procedure. A well-performed calibration procedure allows to obtain a satisfactory estimation of the camera parameters. To correctly map image points to the corresponding real world coordinates, the rigid transformation matrix must be estimated with respect to the user-defined reference system of the calibration master. Thus, if reference frame H changes or the camera frame C moves, it is necessary to estimate again the correct rigid transformation matrix.

Images acquired by the camera are processed, then the hand skeleton joints coordinates are calculated. By using Equation 1 for each frame, it is possible to obtain the real world coordinates of the hand skeleton joints in real time (Fig. 1).

B. Robot Workspace Calibration

The robot workspace refers to the space in which the robot moves (purple square of Fig. 2) with respect to the user workspace (green square of Fig. 1). In this case, the user hand real-world position in reference system H is mapped to the new reference system W. The mapping between reference system H and reference system W is obtained easily if the two workspaces have the same dimension (matrix **[R|t]** is the identity matrix) or if one workspace is a scaled version of the other one (matrix **[R|t]** is the identity matrix multiplied by the scale factor).

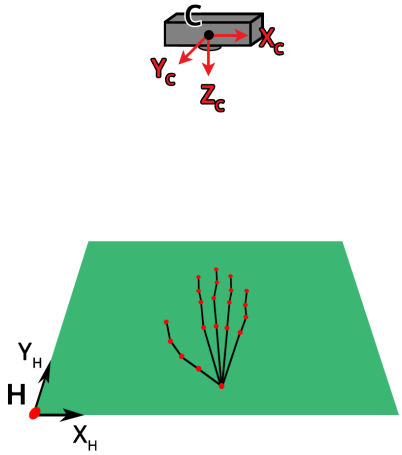

Fig. 1. Scheme of the user workspace, where the camera reference system C and the user-defined reference system H are calibrated to find the correspondence between image points and real world coordinates.

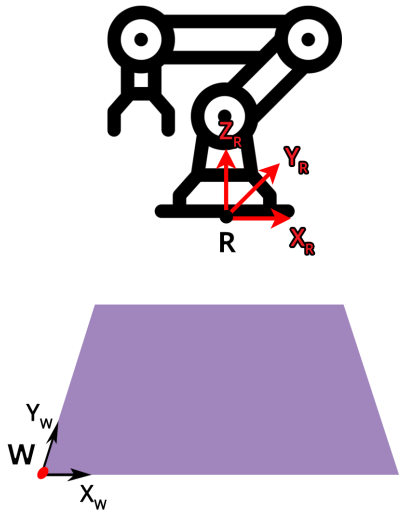

Fig. 2. Scheme of the robot workspace, where the second user-defined reference system W and the robot reference system R are calibrated to find the correspondence between the workspace positions and the end-effector joint coordinates.

To correctly move the robot in a cartesian position of reference system W, it is necessary to perform a calibration between reference system W and the robot reference system R. This procedure has been carried out experimentally by moving the robot (using its manual guidance mode) in different Cartesian positions of reference system W (Fig. xx). The robot correct positioning on top of each calibration position has been assured by using a 3D-printed centering tool (Fig. xx). The tool must be centered manually on each calibration marker and secured in place, then the robot end effector can be moved on it and carefully positioned inside the purposely made circular cavity of the tool. When the positioning is complete, the robot coordinates (both in the

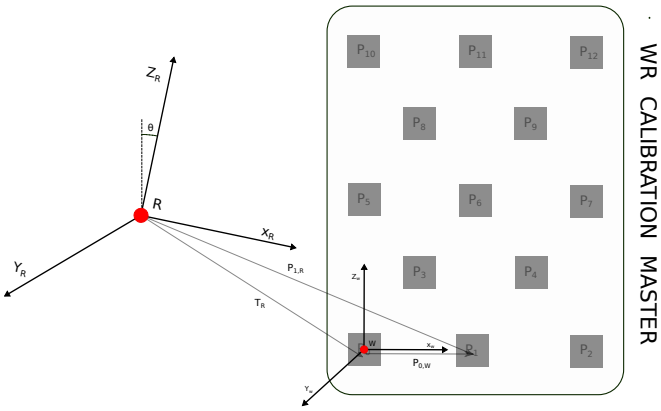

Fig. 3. Master used to calibrate the second user-defined reference system W with the robot reference frame R. The Figure illustrate the position of the point $\mathbf{P_1}$ in both reference frames. To properly calibrate the system, the position of each point is required, both for frame W and R. In the context, 13 calibration points have been used.

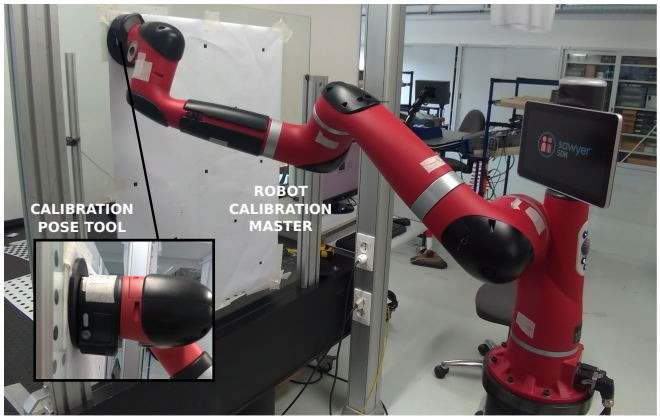

Fig. 4. Master used to calibrate the second user-defined reference system W with the robot reference frame R. The Figure illustrate the position of the point $\mathbf{P_1}$ in both reference frames. To properly calibrate the system, the position of each point is required, both for frame W and R. In the context, 13 calibration points have been used.

Cartesian space and in the Joints space) corresponding to that particular marker (of which the positioning is known with respect to reference system W) can be extracted using ROS or the robot proprietary software. When a satisfactory number of calibration positions has been acquired, it is possible to estimate the rigid transformation matrix between workspaces W and R as follow.

Considering a generic point $\mathbf{P}_1 \in \mathbb{R}^2$ relative to the frame W ($P_{1,w}$) and the frame R ($P_{1,R}$) as shown in Figure 4. Consider $\mathbf{T}_R \in \mathbb{R}^2$ as the distance between the two frame, it is possible to affirm:

$$\mathbf{P}_{0,R} = \mathbf{P}_{0,W} + \mathbf{T}_R \tag{3}$$

Using homogeneous coordinates it is possible to rewrite the previous equation as matrix products:

$$\mathbf{P}_{0,R} = \mathbf{M}_R^W \mathbf{P}_{0,W} \tag{4}$$

Where $\mathbf{M}_R^W \in \mathbb{R}^{3\times3}$ is the rigid transformation matrix composed both by translational and rotational coordinate from frame R to frame W. By evidencing equations from Equation 4, it is possible to write:

$$\begin{aligned} x_{P_{0,W}} &= x_{P_{0,R}} \cos\theta + y_{P_{0,R}} \sin\theta + x_{T_R} \\ y_{P_{0,W}} &= -x_{P_{0,R}} \sin\theta + y_{P_{0,R}} \cos\theta + y_{T_R} \end{aligned} \tag{5}$$

The aim of the calibration procedure is to identify the terms of $\mathbf{M}_R^W$ in order to find the correct position and orientation of frame W with respect to the frame R. However, considering only one calibration position point $\mathbf{P_1}$, the system in Equation 5 results underdetermined, hence, a minimum of $n > 2$ calibration position points is required to solve the system. To minimize the calibration error $n = 13$ points have been considered. Thus, the system in Equation 5 becomes an overdetermined system $\mathbf{Ax} = \mathbf{b}$ that will be solved by mean of the least square method. Such that:

$$\mathbf{A} = \begin{bmatrix} x_{P_{0,R}} & y_{P_{0,R}} & 1 & 0 \\ -y_{P_{0,R}} & x_{P_{0,R}} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{P_{n-1,R}} & y_{P_{n-1,R}} & 1 & 0 \\ -y_{P_{n-1,R}} & x_{P_{n-1,R}} & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ x_{T_R} \\ y_{T_R} \end{bmatrix} \mathbf{b} = \begin{bmatrix} x_{P_{0,W}} \\ y_{P_{0,W}} \\ \vdots \\ x_{P_{n-1,W}} \\ y_{P_{n-1,W}} \end{bmatrix} \tag{6}$$

The rigid transformation matrix $\mathbf{M}_R^W$ used to identify the reference frame W from R is defined by the components of $\mathbf{x}$.

Now, considering the schema in Figure (FIGURA COMPLESSIVA RIFERIMENTI), the generic point $\mathbf{P}_0$ in the robot reference frame R with respect to the camera frame C is calculated as follow:

$$\mathbf{P}_{0,R} = \mathbf{M}_R^W \mathbf{P}_{0,W} \tag{7}$$

$$\mathbf{P}_{0,W} = K_s \mathbf{P}_{0,H} \tag{8}$$

Where $K_s \in \mathbb{R}$ is the a scaling factor between the robot and user's workspace. At the end, considering Equations (7,8) and Equation 2, the point $\mathbf{P}_0$ in the robot reference frame using the camera coordinates results:

$$\mathbf{P}_{0,R} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = K_s \mathbf{M}_R^W \left(s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \mathbf{K}^{-1} - \mathbf{t} \right) \mathbf{R}^{-1} \tag{9}$$

The space coordinates (u, v) will be the output of the hand-gesture recognition algorithm, while the coordinates (x, y, z) are the position set-points for the robot.

III. Hand-Gesture Recognition

The proposed teleoperation method is based on the recognition of the user hands skeleton.

Each frame acquired by the RGB camera (in our set-up, a Kinect v2 camera) is processed by the software, which leverages the OpenPose hand skeleton recognition network to predict the hand skeleton, following the details of [15]. The gesture recognition procedure is based on the position of the reference keypoint (red keypoint 0 in Fig. 5) and on the position of the four knuckles keypoints (blue keypoints $5, 9, 13, 17$ in Fig. 5). According to the knuckles and the thumb (x, y) keypoints positions (from keypoint 1 to 4), the

hand orientation can be hypotized as upright, left oriented, right oriented, or upside down. This allows the software to recognize the gestures regardless of the hand orientation.

To recognize if a certain finger is opened or closed, we consider the Euclidean distance between the reference keypoint 0 and the last keypoint of each finger (pink keypoints $8, 12, 16, 20$ in Fig. 5). If the last keypoint of a finger is not recognized by the network, we consider the corresponding Euclidean distance equal to 0. Using this logic, we defined two gestures used to carry out basic teleoperation tasks: the **open hand** gesture, where all the fingers are detected as opened, and the **index** gesture, where the index finger is detected as open and with a corresponding Euclidean distance much greater than the Euclidean distances of the other fingers. This requirement has been proved useful to reduce the recognition error of the index gesture due to a wrong prediction of the fingers keypoints.

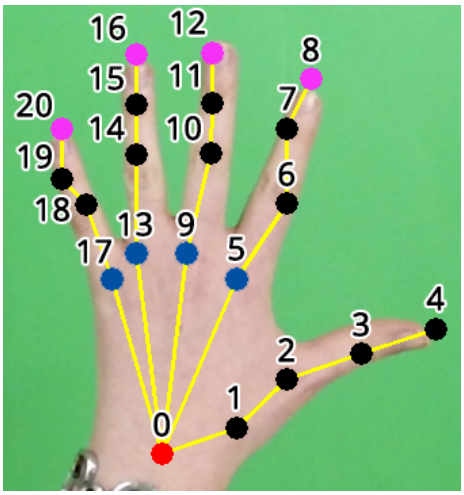

Fig. 5. Example of an open hand gesture correctly recognized skeleton. The red keypoint is the reference keypoint, the blue keypoints are the knuckles keypoints and the pink keypoints are the fingertips keypoints.

Considering the calibration procedure detailed in Section II, a certain position P_H of user workspace H corresponds to a certain robot end-effector position P_W in workspace W. Hence, to move the robot end-effector in position P_W using the software, users must:

1) place their hand in position P_H (corresponding to position P_W), using the real-time visualization of the software as guidance (Fig. xx a);
2) perform the open hand gesture to allow the coordinate extraction (Fig. xx b);
3) perform the index gesture, carefully pointing the index finger to position P_H (Fig. xx c).

It is worth noting that, since the hand skeleton is obtained by a neural network which estimates the joints coordinates frame per frame, their position in consectuive frames may vary. Therefore, our software extracts N different P_H coordinates from N consecutive index gestures recognized in consecutive frames. The average coordinates are extracted to reduce positioning errors introduced by the hand skeleton recognition network. The higher the value of N, the higher the error reduction, at the cost of a higher delay before the final P_H coordinates are extracted. In our set-up, we set $N = 7$. After a position P_H is obtained, the corresponding robot position P_R is calculated and the robot is moved there using ROS. To perform a new robot movement, the procedure in Fig. xx must be repeated from the start.

IV. EXPERIMENTAL EVALUATION

A reliable teleoperation system is obtained if the robot correctly moves to the desired position with a low positioning error. In the case of the proposed set-up, the positioning error is obtained as a sum of different positioning errors, obtained as follows:

1) thanks to the camera calibration procedure, a certain index finger position P_C [px] in the acquired image frame corresponds to a certain Cartesian point P_H [mm] in workspace H. Since the camera calibration procedure introduces an estimation error E_1, we obtain $P_H = P_C + E_1$;
2) our software extracts N consecutive finger positions estimated by OpenPose neural network and calculates the average of them. The resulting point is $P_O = \frac{\sum_{n=1}^{N}(P_{Hn}+E_{2n})}{N}$, where E_2 is the error between the real position of the index finger in the image and the index keypoint estimation made by OpenPose;
3) the extracted index finger position P_O corresponds to a certain Cartesian point in workspace W. According to the mapping between workspace H and W, we obtain $P_W = P_O + E_3$, where E_3 is the positioning error caused by this mapping;
4) the robot end-effector will move in cartesian point P_R, obtained considering the rigid transformation between workspaces W and R. This transformation introduces the positioning error E_4, thus we obtain $P_R = P_W + E_4$.

Hence, we obtain the final end-effector position P_R as:

$$P_R = \frac{\sum_{n=1}^{N}(P_{Cn} + E_{1n} + E_{2n})}{N} + E_3 + E_4 \qquad (10)$$

Considering this formula, and that in our set-up E_3 can be assumed equal to zero (because we kept workspace H and workspace W dimensions except for the scaling factor), we designed two experiments to assess if the positioning error obtained depends (i) on the camera calibration (E_1), (ii) on the estimation of the hand skeleton (E_2), or (iii) on the robot calibration (E_4).

A. Evaluation of the skeleton estimation error

The positioning error due to the estimation of the hand skeleton joints made by OpenPose neural network has been evaluated considering the theoretical position of the index in the image T and the index joint position in the image A calculated by the software (Fig. xx).

B. Evaluation of the robot positioning error

In our set-up, reference system H is placed horizontally with the camera mounted still at a $1m$ distance (Fig. xx). Reference system W, however, has been placed vertically on a glass pane (Fig. xx). To reliably assess the positioning of the robot end-effector, a red laser has been mounted on the end effector (laser name bla bla) (Fig. xx). When the robot is moved to a certain theoretical position T, the laser will point to its actual positioning A. To correctly visualize and measure the robot workspace and the laser positioning, an RGB camera (camera serial name) has been mounted behind the glass pane. A measuring software has been developed using LabVIEW to measure the distance between the theoretical position T (calculated as the barycenter of the black dot of the experimental master as in Fig. xx) and the actual positioning A (red dot in the image as in Fig. xx).

We moved the robot using the proposed system in xx theoretical positions. To do that, we used the Cartesian positions corresponding to the circular markers barycenters (theoretical positions) to move the robot using ROS. Hence, this procedure avoids considering the hand skeleton estimation errors.

V. Figures and Tables

TABLE I
AN EXAMPLE OF A TABLE

One	Two
Three	Four

VI. CONCLUSIONS

Conclusioni sul progetto/esperimenti ottenuti. Problematiche incontrate e come sono state risolte. Future developments.

References

[1] H. A. Yanco and J. L. Drury, "A Taxonomy for Human-Robot Interaction," *Engineering*, p. 9, 2002.

[2] J. Vertut and P. C. Coeffet, *Robot Technology; vol. 3A Teleoperation and Robotics Evolution and Development*. 1986.

[3] J. Rebelo, T. Sednaoui, E. B. Den Exter, T. Krueger, and A. Schiele, "Bilateral robot teleoperation: A wearable arm exoskeleton featuring an intuitive user interface," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 62–69, 2014.

[4] X. Lv, M. Zhang, F. Cui, and X. Zhang, "Teleoperation of robot based on virtual reality," *Proc. - 16th Int. Conf. Artif. Real. Telexistence - Work. ICAT 2006*, pp. 400–403, 2006.

[5] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[6] C. Glover, B. Russell, A. White, M. Miller, and A. Stoytchev, "An effective and intuitive control interface for remote robot teleoperation with complete haptic feedback," *Proc. 2009 Emerg. Technol. Conf. (ETC), Ames, IA, USA*, 2009.

[7] R. Boboc, H. Moga, and D. TALAB, "A Review of Current Applications in Teleoperation of Mobile Robots," *Bull. Transilv. Univ. Brasov Ser. I Eng. Sci.*, vol. 5, no. 54, pp. 9–16, 2012.

[8] J. Vogel, C. Castellini, and P. van der Smagt, "EMG-based teleoperation and manipulation with the DLR LWR-III," pp. 672–678, 2011.

[9] H. F. Hassan, S. J. Abou-Loukh, and I. K. Ibraheem, "Teleoperated robotic arm movement using electromyography signal with wearable Myo armband," *J. King Saud Univ. - Eng. Sci.*, no. xxxx, 2019.

[10] L. Roveda, S. Haghshenas, A. Prini, T. Dinon, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Fuzzy Impedance Control for Enhancing Capabilities of Humans in Onerous Tasks Execution," in *2018 15th Int. Conf. Ubiquitous Robot. UR 2018*, pp. 406–411, Institute of Electrical and Electronics Engineers Inc., aug 2018.

[11] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Trans. Ind. Electron.*, vol. 52, no. 5, pp. 1206–1219, 2005.

[12] S. Livatino, G. Muscato, and F. Privitera, "Stereo viewing and virtual reality technologies in mobile robot teleguide," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1343–1355, 2009.

[13] L. Peppoloni, F. Brizzi, C. A. Avizzano, and E. Ruffaldi, "Immersive ROS-integrated framework for robot teleoperation," *2015 IEEE Symp. 3D User Interfaces, 3DUI 2015 - Proc.*, pp. 177–178, 2015.

[14] H. Hedayati, M. Walker, and D. Szafir, "Improving Collocated Robot Teleoperation with Augmented Reality," *ACM/IEEE Int. Conf. Human-Robot Interact.*, pp. 78–86, 2018.

[15] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *CVPR*, 2017.

[16] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[18] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2001.