



Gesture-Based Feedback in Human-Robot Interaction for Object Manipulation

Leandro Filipe^{1,2}(✉), Ricardo Silva Peres^{1,2}, Francisco Marques^{1,2}, and Jose Barata^{1,2}

¹ Centre of Technology and Systems (CTS), UNINOVA, Caparica, Portugal
{leandro.filipe, ricardo.peres, fam, jab}@uninova.pt

² School of Science and Technology, NOVA University of Lisbon, Caparica, Portugal

Abstract. Human-Robot Interaction is a currently highly active research area with many advances in interfaces that allow humans and robots to have bi-directional feedback of their intentions. However, in an industrial setting, current robot feedback methods struggle to successfully deliver messages since the environment makes it difficult and inconvenient for the user to perceive them. This paper proposes a novel method for robot feedback, leveraging the addition of social cues to robot movement to notify the human of its intentions. Through the use of robotic gestures, we believe it is possible to successfully convey the robots' goals in interactions with humans. To verify this hypothesis, a proof of concept was developed in a simulated environment using a robotic arm manipulator that notifies the user using gestures when it needs to correct the pose of an object.

Keywords: Robotics · Human-Robot Interaction · Simulation · Object manipulation · Machine Learning

1 Introduction

With the emergence of Industry 4.0, the role of robotics in the industry has shifted from big unmovable scary machinery to socially accepted anthropomorphic robots with the goal of improving productivity and producing higher quality products at reduced costs [1]. There is also an increased effort in having both humans and robots in the same workspace, since both have their one strengths and limitations, creating a collaborative and safe working environment will result in higher productivity and decreased production times [2].

To better achieve this collaboration, some sort of communication is needed. This should be an intuitive method without the need for expert knowledge. Although existing interfaces meet these requirements, it is believed that in an industrial setting these would not be enough as they would struggle to deliver the intended message [3]. The need then arises for a better interface for Human Robot Interaction (HRI).

When talking about this subject it is difficult to leave out the existing social component. According to Erel et al. [4], there are implicit social cues in robot movement automatically interpreted by the human being, and these need to be taken into account

and leveraged by the programmer. Taking this and the previously described challenges into account, the following research question comes naturally.

RQ: *How can social cues be leveraged in robotic movement to improve communication in Human-Robot interaction in a more natural way?*

As a hypothetical solution to this problem, it is proposed the use of robotic gestures to feedback the user of the robot's goals, problems, and intentions, thus improving communication in HRI. Additionally, given the social component of this type of interaction, it is expected that the use of robotic gestures will allow for a more natural and comfortable interaction for the user. Furthermore, it will also allow a greater and easier adoption by users without a more technical background. To validate our hypothesis, a simple proof of concept was developed where a robotic arm is in charge of a pick-and-place task after notifying the user of an incorrect object pose.

With the fourth industrial revolution, many new possibilities came along. More specifically, the recent technological advances in the computational area made possible the use of virtualization as an easier, cheaper, faster way of development. Recent simulation software focuses on the development of life-like scenarios to increase the generability of a solution to the real world.

Moreover, recent changes triggered a shift in development methods, proving the efficiency and usefulness of digitalization and remote work. Teams are now able to develop and collaborate without the pricey overhangs associated with logistics.

Our work takes advantage of simulation as a virtualization method to more easily test and validate our solution. Not only that, the use of life-like scenarios enables the generation of synthetic data to train the used Machine Learning modules, both speeding up the process and reducing costs. Additionally, given the novelty of this subject, the discussion of ideas and possible collaboration will greatly favor this project.

The rest of the document is structured as follows: In Sect. 2 previous work related to this project is presented. Section 3 describes the materials and methods involved in the implementation of this project, containing the system design process (Sect. 3.1), the implementation (Sect. 3.2), and the tests and validation methods used (Sect. 3.3). Section 4 is where the results are analyzed and discussed. Some limitations of this proof of concept are exposed in Sect. 5. Finally, Sect. 6 presents the drawn conclusions and some additional future work is proposed.

2 Related Work

The idea of collaboration between humans and robots it's not unheard of, HRI has been a highly active research topic with the emergence of the Industry 4.0 paradigm [1], and rightly so. There are many reasons for choosing collaborative systems like economic motivations, efficiency in the use of space, and flexibility [2]. Collaborative cells can also adapt well to situations where a constant change in production layout is required since a rigid safety system is not necessary and can more easily be repurposed [5].

Recent research has pointed several different approaches for HRI that pass through speech [6], gestures [7], Augmented Reality (AR) [8], or multimodal systems [9]. Solutions like these are well researched but most of them focus on robot control which we

believe covers only half of the problem. For successful collaboration, there needs to be communication from both sides.

On this note, previous work was done regarding situations where the robot needs to notify the user. Berg et al. [10] feedback the user about robot information using a projector to display it. A verbal approach was developed by St. Clair et al. [11] where three types of situated verbalizations aimed at providing useful information are dynamically generated by the robot. However, in many cases like the ones previously listed, the approaches are not tested in a real manufacturing environment, and so there is a lack of assurance that these methods would play well in an industrial setting. This assumption comes from the existence of obstacles such as loud noises that would pose a problem for audio-based solutions, and the additional equipment required for other solutions would be frowned upon by the robot operators.

A new idea then emerges of using robotic gestures to notify the user with information, guide it through difficult tasks, and help to complete objectives. Robot gestures have been previously shown to be socially interpreted by humans and that this phenomenon should be leveraged [4]. Lohse et al. [12] take advantage of this behavior in an experiment where a Nao robot attempts to give route directions with and without robot gestures. The obtained results show that the use of robotic gestures increases user performance and indicates a promising means to improve HRI tasks.

Taking this into account the authors of this paper believe that the use of robotic gestures can be beneficial in HRI tasks in a manufacturing setting. This method is believed to not have to face as many obstacles as different solutions and would improve social acceptance of robots and both assurance and comfortability while collaborating with one. It is then proposed the implementation of a proof of concept that would help validate this hypothesis. Although not tested and validated in a real environment, the goal of this project is to verify the usefulness of an HRI framework of said nature before it is tested in a real industrial setting.

3 Materials and Methods

In this section, the overall planning and execution of the project will be discussed. This includes the architecture design (Sect. 3.1), implementation of the desired features and behaviors (Sect. 3.2), and the methods used to test and validate our solution (Sect. 3.3).

3.1 System Design

For this case study, a simple proof of concept was envisioned to confirm our hypothesis that a robot can give feedback to its human user using only gestures. The scenario that was chosen consists of a pick-and-place example using a robotic arm and a cube. The manipulator's objective is to pick up the cube and place it in a goal position. There is, however, a constraint, the arm can only pick up the cube at a specific orientation. To surpass this challenge the robot needs to ask the user to rotate the cube, using only gestures, until the desired orientation is achieved, after that the cube can be placed in the goal position.

To ensure correct behavior, the robot needs to be able to estimate the cube’s position and orientation and plan its movement accordingly. To meet this requirement, the architecture of this project (Fig. 1) will require a pose (position and orientation) estimation model that will receive an RGB image and output the desired information. This information is then passed to a motion planner that is responsible for planning the movement of the robot depending on the position and orientation of the cube. This module is also responsible for deciding whether the cube can be picked up or if the robot needs to inform the user that some adjustments to the cube’s orientation are necessary.

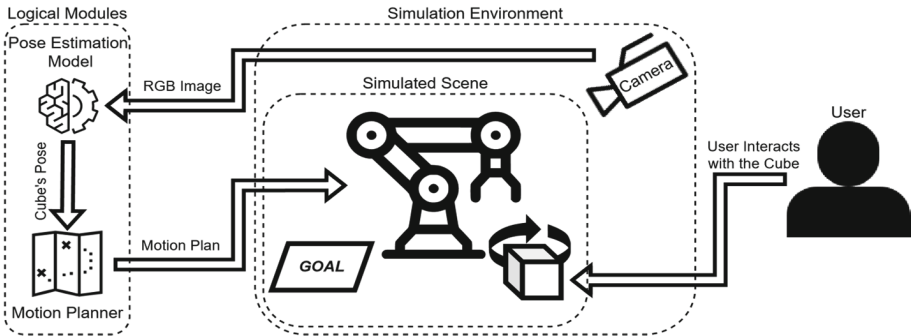


Fig. 1. Proposed architecture for this project.

As can be seen in Fig. 1, the required components are split between logic and interaction modules. The modules responsible for logic operations are not inserted in the simulation environment, since many existing frameworks do not support the necessary tools for robot control. This also serves the purpose of encapsulating similar modules together and isolating the simulation environment, offering greater generalization of our solution and enabling the use of differently implemented modules in conjunction with existing ones.

3.2 Implementation

To implement this project the Unity platform was used since it meets all the requirements imposed for the simulation environment and offers great community support and documentation. Unity also has native support for robotics projects with Unity Robotics Hub which enables the integration with Robotic Operating System (ROS)¹. Conveniently, Unity Robotics Hub offers an Object Pose Estimation demonstration [13] that already meets most of the requirements for this project. Taking this into account, it was decided to take advantage of the given opportunities and use the aforementioned solution as a starting point for our project.

¹ <https://www.ros.org/>.

As can be seen in Fig. 2, the overall architecture of the Object Pose Estimation tutorial is very similar to the proposed architecture for this project. In both architectures, there is a separation of the logic and interaction modules, wherein Fig. 2 both the pose estimation model and the motion planner modules are implemented within ROS, designed specifically for robotics projects.

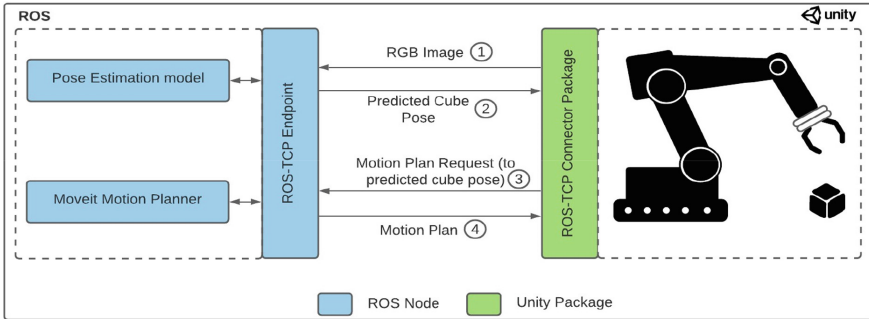


Fig. 2. Unity’s architecture for the Object Pose Estimation tutorial. Taken from [13].

Overall, three packages are used for implementing this project, the Unified Robot Description Format (URDF) Importer package² to import the robot model into the simulation scene, the TCP Connector package (see footnote 2) so that Unity can communicate with ROS and vice-versa via a TCP endpoint, and the Perception package (see footnote 2) that provides a toolkit for generating large-scale datasets for computer vision training and validation. The ROS workspace comes already configured inside a Docker container with all the necessary dependencies and uses the Moveit [14] motion planner and a custom Convolutional Neural Network (CNN) for pose estimation (CNNs are frequently used in the literature for object pose estimation [15–17]). Figure 3 shows a representation of this model’s architecture.

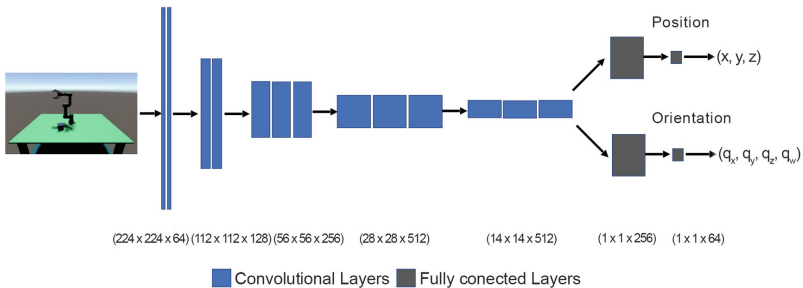


Fig. 3. The pose estimation model. Taken from [13].

² These packages and the version used are referenced in [13].

This model is a modified implementation of the one presented by Tobin et al. [18] that given an RGB image of the scene, outputs the position and orientation of the cube. To make the model more robust and generalizable to the real world, domain randomization was added by randomizing the pose of the cube, the pose of the target goal, and the lighting of the scene, using the Perception package. The same package is also responsible for labeling each image with a bounding box containing the pose of the cube. The model was trained with a dataset containing 30000 training images and 3000 validation images.

The motion planner that was used is Moveit, one of the most widely used software for robotic manipulation. This module receives the information containing the pose of the cube and the target goal from Unity and plans the motion of the robot accordingly so it can pick up the cube and place it in the goal position. It is in this module that the necessary features to achieve the behavior explained in the previous section were implemented in the Python programming language.

Firstly, an improvement to the overall motion planner was necessary. Although the accuracy regarding the pick-and-place behavior was sufficiently high, the movements produced by the robot were somewhat awkward, resulting in the robot having to do a lot of unnecessary movements. The example utilizes the Open Motion Planning Library (OMPL)³ with its default RRTConnect [19] algorithm. The replacement of this algorithm for RRT* [20], the additional planning time, and the increased number of concurrent planning jobs were sufficient to achieve a much better result with cleaner motion.

Secondly, to integrate robot feedback through gestures, a verification of the incoming message containing the cube's pose from Unity is needed. The objective here is to check whether the cube has the correct orientation for the manipulator arm to pick it up. For simplification, the desired orientation chosen was 0° with a tolerance of 10° in both directions, for the z-axis. If the cube's orientation does not meet this requirement the robot is instructed to perform a gesture above the cube. For this purpose, the Pilz Industrial Motion Planner was used in place of the OMPL planner, which enables the generation of circular paths around a center point. The direction of the rotation of the robotic arm depends on the orientation of the cube, the arm always rotates in the direction of the least necessary adjustment, making the user's life a little bit easier. Otherwise, if the cube is in the correct orientation (or inside the allowed interval) the robot can pick it up and place it in the target position.

Finally, a C# script was developed so the user can rotate the cube using a keyboard. The desired plan of action is after being notified by the robot that the cube needs to be rotated, the user will adjust the cube's orientation (preferably rotating it in the optimal direction, as alerted by the robot) and inform the robot that the cube is ready to be placed inside the goal. A shot from the implemented simulation can be seen in Fig. 4 where the robot just finished the circular motion above the cube.

³ <https://ompl.kavrakilab.org/>.

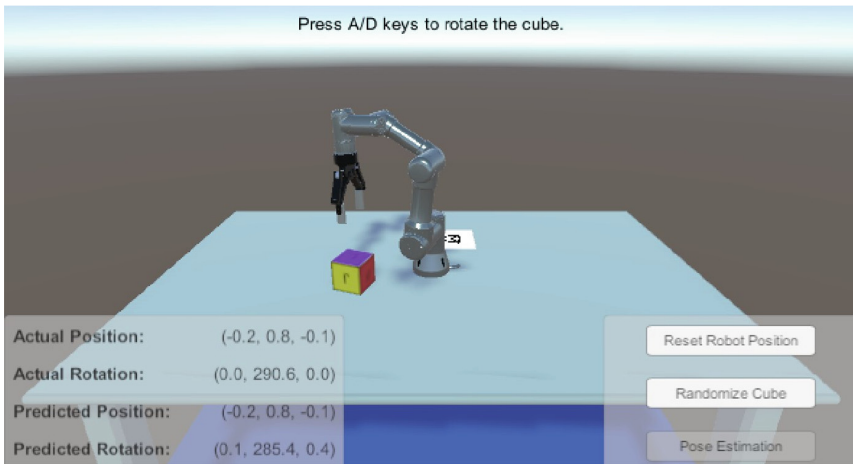


Fig. 4. A shot of the robot after finishing giving feedback and waiting for the user to interact with the cube.

3.3 Tests and Validation

Since the implementation of the starting point for this project has been previously validated, it was decided that the main focus of validation for this work should be the social interpretation of the robot's gestures. With this in mind, an experiment involving 12 participants with higher education on the field (from ages 20–40; 10 male, 2 female) was designed. The experiment aims to provide some feedback on how users perceive and feel about this solution.

The participants were placed in front of the simulation and asked to interact with the robot. It was explained that the objective of the robot is to pick up the cube and place it inside the goal, however, the robot needs the user's help to do so. After interacting with the simulation, the participants were asked to anonymously answer a survey regarding what they just experienced. The survey consists of the nine following questions:

1. **How old are you?;**
2. **What is your gender?;**
3. **What is your level of education?;**
4. **How satisfied are you with the look and feel of the robot's movements? (weight of 1);**
5. **How intuitive is it to understand the robot's feedback? (weight of 3);**
6. **How satisfied are you with the reliability of the solution? (weight of 2);**
7. **How useful do you think this solution would be in a manufacturing setting? (weight of 2);**
8. **Would you recommend this solution to a colleague/friend?;**
9. **How many corrections were made to the cube?**

Additionally, some weight was added to the questions that require a score between one and five so that an overall score can be attributed to this solution and to provide a

point of reference for future work. The weights were given from a range of one to three according to the perceived importance of each question.

4 Results Analysis

Questions 4 to 7, inclusive, are the most significant ones in the survey presented to the participants, and as such, special attention was given to them. Looking at the graphs in Fig. 5, the results obtained from these four questions can be seen. In these pictures, the vertical axis corresponds to the number of responses and the horizontal axis represents to the score given in that question.

While the results may not appear as good as expected, this is aligned with the early stage of development of the solution. The usefulness of such validation stems from the possibility to collect valuable feedback for future iterations of the HRI. According to the participants, the look and feel of the robot’s movements are mostly pleasant. The robot proves to be reliable for the most part, obtaining an average rating of 3.5 on the reliability scale. While sufficient, perhaps this is an aspect for improvement in future work.



Fig. 5. Graphical representation of some of the obtained results. The vertical axis corresponds to the number of responses and the horizontal axis corresponds to the score given in that question.

Another aspect for improvement is how intuitive the gestures produced by the robot are. Despite getting an average rating overall, there are instances where the solution was rated as not intuitive at all. In addition, some participants reported having difficulty understanding the direction of the circular gesture and how much they had to rotate the cube. This is an important point and will be considered in future implementations.

Lastly, based on what was presented to them, the participants consider this to be a useful solution in a manufacturing setting. Question 7 obtained a significant score with an average of approximately four on a scale of usefulness from 1 to 5. This is an important result because it is a big step in the validation of our solution. Additionally,

10 out of the 12 participants said that they would recommend this solution to a friend or colleague, showing an overall appreciation of this solution.

In addition to these results, an evaluation was also made regarding how many attempts it would take the user to get the cube to the correct position, i.e., how many times the robot had to signal the user to rotate the cube. Figure 6a shows a box chart of the corrections needed to complete the simulation by the participants. Although there was one outlier case where it took the user six attempts to reach the correct orientation of the cube, it took participants on average less than three attempts to correct the orientation of the cube. Ideally, this number would be lower, however, it will serve as an evaluation metric for future solutions.

Finally, the overall score was calculated according to each participant's answers. This result is calculated by multiplying the rating of each of the four questions ranked from 1 to 5 by its respective assigned weight and then adding it all up. Attending to the box plot in Fig. 6b we can see that an average result of $\mu = 28.583$ and a standard deviation $\sigma = 3.523$ was obtained on a maximum of 40 points. This result, although still not very representative of the quality of the developed solution, will be used as a reference point for future implementations, always aiming to overcome it.

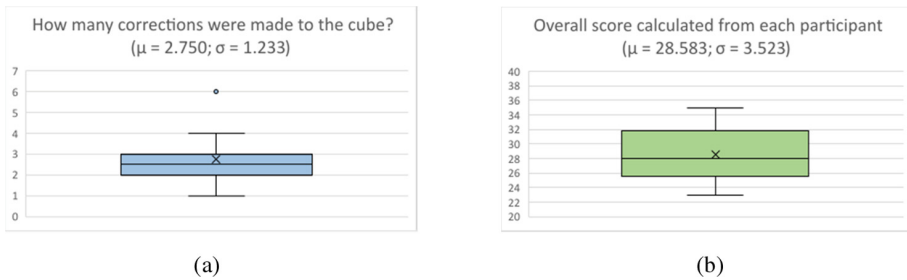


Fig. 6. Box charts of: (a) The number of corrections made to the cube by the participants; (b) The overall score calculated from the participants' answers.

5 Limitations

One of the limitations of this work is the fact that there is no real-world validation. Despite achieving promising results in a simulated and controlled setting, these are not directly transferable to the relevant operational environment, since possible real-world obstacles and drawbacks may not have been taken into consideration. Future implementations will take this into account when performing validation.

Additionally, the conducted experiment to validate our solution has a considerably small population due to time and cost constraints. For a more robust evaluation the experiment should consider a bigger and more diverse population for general use, or a more specific population for a manufacturing setting validation.

Finally, given the innovative nature of the solution proposed here, there are no alternatives yet in the literature that offer a term of comparison. As such, any argument that

claims to compare our solution with other alternatives is merely an assumption that would be difficult to support. That said, using a global metric to evaluate the solution will allow us to validate the assumptions made here in future work by directly comparing it to existing approaches.

6 Conclusion and Future Work

This paper proposes a framework for HRI that focuses on the use of robotic gestures to enable a robot of communicating with a user in a manufacturing environment. The main contributions of this work can be summarized as follows. A base project from Unity was modified in order to implement the proposed framework which was validated through an experiment involving 12 participants. Furthermore, a global scoring methodology was created to enable the direct comparison of this solution with different future approaches. The results obtained from the conducted experiment prove that although not as intuitive as initially thought, the presence of robotic gestures in HRI scenarios proves to be a useful addition.

With this in mind, and answering the research question raised in Sect. 1, it is possible to verify that with the integration of robotic gestures as social cues to a robot's movement, there is an improvement to the interaction between a human and a robot. In addition, it is expected that the shortcomings and limitations of this solution will serve to drive any future work in the HRI topic with a focus on robot feedback. For better visualization of the implemented solution, animations containing examples of the robot's behavior can be found here: <https://bit.ly/3oRtLoV>.

As future work, many aspects of this implementation can be improved upon. The circular path above the object cannot always be feasible due to reachability constraints, for that it is suggested to modify the gesture to accommodate such restraints. As stated in Sect. 4, one aspect to improve is the translation of movement in the robot's gestures where some participants reported difficulties in perceiving how much the robot wanted them to rotate the cube. By generating the path according to how much the user has to rotate the cube this can be avoided, although, another issue can be raised since for minimal corrections the path would most likely be incomprehensible. Finally, as a step forward to this implementation, different gestures for different use-cases will be implemented to improve the robustness of the solution.

References

1. Goel, R., Gupta, P.: Robotics and Industry 4.0. In: Nayyar, A., Kumar, A. (eds.) *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*, pp. 157–169. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-14544-6_9
2. Matheson, E., Minto, R., Zampieri, E.G.G., Faccio, M., Rosati, G.: Human–robot collaboration in manufacturing applications: a review. *Robotics* **8**, 100 (2019)
3. Berg, J., Lu, S.: Review of interfaces for industrial human-robot interaction. *Curr. Robot. Rep.* **1**(2), 27–34 (2020). <https://doi.org/10.1007/s43154-020-00005-6>
4. Erel, H., Tov, T.S., Kessler, Y., Zuckerman, O.: Robots are always social. *ACM* **5**, 1–6 (2019). <https://doi.org/10.1145/3290607.3312758>

5. Fechter, M., Foith-Förster, P., Pfeiffer, M.S., Bauernhansl, T.: Axiomatic design approach for human-robot collaboration in flexibly linked assembly layouts. *Proc. CIRP* **50**, 629–634 (2016). <https://doi.org/10.1016/j.procir.2016.04.186>
6. Maksymova, S., Matarneh, R., Lyashenko, V.V., Belova, N.V.: Voice control for an industrial robot as a combination of various robotic assembly process models. *J. Comput. Commun.* **5**(11), 1–15 (2017). <https://doi.org/10.4236/jcc.2017.511001>
7. Neto, P., Simão, M., Mendes, N., Safeea, M.: Gesture-based human-robot interaction for human assistance in manufacturing. *Int. J. Adv. Manuf. Technol.* **101**(1–4), 119–135 (2018). <https://doi.org/10.1007/s00170-018-2788-x>
8. Fang, H.C., Ong, S.K., Nee, A.Y.C.: Novel AR-based interface for human-robot interaction and visualization. *Adv. Manuf.* **2**(4), 275–288 (2014). <https://doi.org/10.1007/s40436-014-0087-9>
9. Andronas, D., Apostolopoulos, G., Fourtakas, N., Makris, S.: Multi-modal interfaces for natural human-robot interaction. *Proc. Manuf.* **54**, 197–202 (2021). <https://doi.org/10.1016/j.promfg.2021.07.030>
10. Berg, J., Lottemoser, A., Richter, C., Reinhart, G.: Human-robot-interaction for mobile industrial robot teams. *Procedia CIRP* **79**, 614–619 (2019). <https://doi.org/10.1016/j.procir.2019.02.080>
11. Clair, A.S., Mataric, M.: How robot verbal feedback can improve team performance in human-robot task collaborations. *IEEE Comput. Soc.* **3**, 213–220 (2015)
12. Lohse, M., Rothuis, R., Gallego-Perez, J., Karreman, D.E., Evers, V.: Robotgestures make difficult tasks easier: the impact of gestures on perceived workload and task performance. *ACM* **4**, 1459–1466 (2014). <https://doi.org/10.1145/2556288.2557274>
13. Unity-technologies/robotics-object-pose-estimation. <https://github.com/Unity-Technologies/Robotics-Object-Pose-Estimation/>
14. Coleman, D., Sucas, I., Chitta, S., Correll, N.: Reducing the barrier to entry of complex robotic software: a move it! case study. arXiv preprint arXiv:1404.3785 (2014)
15. Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., Birchfield, S.: Deepobject pose estimation for semantic robotic grasping of household objects. arXiv preprint arXiv:1809.10790v1 (2018)
16. Doosti, B., Naha, S., Mirbagheri, M., Crandall, D.J.: Hope-net: A Graph-Based Model for Hand-Object Pose Estimation, pp. 6608–6617 (2020). <http://vision.sice.indiana.edu/projects/hopenet>
17. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199v3 (2017)
18. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domainrandomization for transferring deep neural networks from simulation to the real world arXiv preprint arXiv:1703.06907 (2017)
19. Kuffner, J., LaValle, S.: Rrt-connect: An Efficient Approach to Single-Query Path Planning, vol. 2, pp. 995–1001. IEEE (2000). <http://ieeexplore.ieee.org/document/844730/>
20. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. arXiv preprint arXiv:1105.1186 (2011)