Safe robot affordance-based grasping and handover for Human-Robot assistive applications

Safe robot affordance-based grasping and handover for Human-Robot assistive applications

Cesare Luigi Blengini, Pangcheng David Cen Cheng, Marina Indri
Dipartimento di Elettronica e Telecomunicazioni - Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129, Torino, Italy
cesareluigi.blengini@studenti.polito.it, {pangcheng.cencheng, marina.indri}@polito.it

***Abstract*—A crucial aspect of human-robot collaboration involves the robot's ability to safely perform handovers of objects to be used by the operator. In this study, we introduce two complementary frameworks designed to execute every stage of a robot-to-human handover process, using only RGB-D input data. The first framework employs a machine learning model, trained on a custom real-world dataset, to detect objects and their parts' affordances. Affordance is encoded using a novel representation based on keypoints, which are utilized to model hazardous sections of objects and plan appropriate grasps. The second framework tracks hand movements to dynamically determine handover locations, while enforcing safety protocols to prevent exposure of dangerous object parts during the movement. Additionally, it ensures correct object orientation upon delivery, presenting the object handle to the human. The effectiveness of the proposed solution has been successfully validated through testing on a real mobile manipulator.**

***Index Terms*—Mobile manipulation, task-oriented object grasping and handover, affordance, robot-to-human handover, human-robot collaboration.**

I. Introduction

Many human-robot assistive applications rely on the robot's capabilities to hand over objects to humans. Some real-world applications include assisting people with motor disabilities or handing the right tools to a worker.

In most human-robot assistive applications, the goal of robots handing over the items to humans is to facilitate the latter to perform specific tasks. This is referred to as task-oriented handover. A comprehensive review of robot handover [1] and an in-depth study [2] revealed gaps in implementing task-oriented grasping and handovers. Currently, this remains an actively researched area, as evidenced in [3].

In human interactions for exchanging items, it is a common practice for the one that delivers the object to offer the most convenient part to grip, like the handle [4], to the receiver, so the latter can comfortably hold it for later use. Additionally, some objects have hazardous components, such as sharp edges or blades, so it is crucial to prevent any harm to the receiver. To replicate this behavior, the robot needs to plan suitable ways to grasp objects and deliver them with the correct handover orientation, allowing comfortable and efficient use of the object after the handover.

The object handover orientation can be learned from real-life handovers [5]. However, a recent study [6] has shown that leveraging information about the affordances of object parts can yield similar results.

In a general-purpose application, the robot needs to autonomously and safely grasp novel objects from a class of manipulable objects. Grasps should be performed with minimal prior information, without relying on object 3D models or complex sensors. Moreover, the grasp of a robot has an influence on the one performed by the human [7]. Affordance estimation can be used to determine which object's parts should be left unoccupied for human interaction.

Another crucial aspect of handover planning takes into account natural interactions while operating efficiently. During human interactions, there exists a sequence of implicit and explicit communication signals, both verbal and non-verbal. These signals enable individuals to identify the different phases of the exchange from beginning to end, contributing to the natural flow of interaction.

Recent advancements in machine learning (ML) for computer vision and robotics have enabled the development of highly effective tools for tracking human movements, modeling objects, and planning grasps [3].

This paper introduces two main contributions compared to the state of the art, overviewed in Section II. Firstly, it proposes a novel approach for the use of a single ML network capable of conducting object detection, affordance and grasp pose estimation for objects from a single RGB-D image of a cluttered scene. This is facilitated by the proposed representation of affordance, utilizing keypoints to easily model object handles and hazardous parts. Secondly, it presents an end-to-end solution, designed to execute all stages of a task-oriented handover process with a focus on safety. This solution leverages hand tracking and affordance to plan grasp and handover poses. Also, it introduces a gesture-based system to initiate the handover.

In particular, the proposed solution employs two neural networks. The first one, as already stated, is used to identify objects, model their hazardous components and plan grasps. Meanwhile the second network manages hand tracking and recognizes gestures. This setup allows the robot to autonomously and safely pick up objects and hand them over to humans with the proper orientation. The developed solution is adaptable and can function as a module within more complex applications.

The proposed architecture, which is developed for low-resource mobile agents using a distributed setting, is introduced in Section III, after the analysis of the state of the art in Section II. The two complementary frameworks that constitute the proposed solution are described in Sections IV and V, which are devoted to affordance detection and pose estimation, and to safe handover, respectively. Section VI illustrates the

experimental setup and reports the results of the carried-out tests. Finally, Section VII draws some conclusions and open issues for possible future works.

II. Related works

In early work on task-oriented robot-to-human handovers, robots were not fully autonomous and could work on a limited number of previously seen objects. For instance, a 3D model of the object was obtained from a laser scan, and the handover orientation was computed based on the choice of a human operator [8], [9].

More recent studies have proposed different solutions to automate the computation of object orientations. A possibility is to learn from human-to-human handover [5] or demonstrations from tool usability [10]. In other approaches, appropriate information is extracted from the manipulable object. In [11], the authors use the information on human grasps [12], together with a 3D model of the object and real-time hand tracking [13] to plan the object grasp and subsequent delivery. In [6], the authors use a neural network to extract information on an object's part affordance and plan handovers accordingly. They show that affordance-based methods can have similar performance as learning from real demonstrations.

The affordance-based approach is ideal to develop a non-specific solution that enables the robot to manipulate a wide variety of objects. In this application, affordance refers to the intuitive use of object parts. In that way, the robot can recognize the object's dangerous parts and those that are useful for human grasping. Several ML-based techniques exist for visual affordance estimation, mainly based on RGB inputs [14] or point cloud data [15].

Planning how to grasp objects is essential for handing them over during tasks. Thanks to advancements in ML, it is now simpler to grasp new and unseen objects using RGB-D data. There are networks dedicated to estimate object poses from RGB data [16], as well as complete end-to-end frameworks for object-agnostic grasping, such as the ones presented in [17] and [18]. These methods can work alongside networks that estimate object affordances. Additionally, there are grasping solutions that already consider object affordances during the planning process [19].

The main limitation of ML for grasping and affordance estimation is the need for complex datasets. These are challenging to construct and annotate, leading to the training of many models using synthetic data, such as in [20].

The robot needs also to identify the target object and then to carry out the delivery at a convenient location for the human. One of the most used algorithms for detecting objects is YOLO [21], while the handover location can be determined by tracking the position of the human hand, e.g., using Mediapipe [13]. In particular, Mediapipe allows real-time hand tracking from RGB images, as well as gesture recognition, providing a way to command the robot.

III. The proposed architecture

The aim of the paper is to enable a mobile manipulator to grasp an object from a cluttered scene and deliver it safely to a human, implementing some functionalities envisioned in [22]. Preliminary results are presented in [23], whose code is available in [24]. The robot can sense the environment with a single RGB-D camera. The grasps are performed by a 6-DOF robotic arm with a parallel gripper.

The general working process is summarized in Figure 1. In particular, two complementary frameworks are proposed:

- The *affordance and grasp pose estimation framework* leverages ML and a novel affordance representation based on keypoints, to extract information about the dangerous parts of an object and plan its grasp accordingly. YOLOv8 Pose model is employed to detect an object and, at the same time, estimate both its position and its affordance from a single-view RGB-D image. The grasp pose is implicitly estimated together with affordance.
- The *safe handover framework* uses MediaPipe. It enables the robot to dynamically track the human hand to plan handovers that (i) prioritize safety, (ii) are predictable, and (iii) feel natural. Also, hand gesture classification is used to give commands to the robot.

IV. Affordance detection and pose estimation

The affordance information is used to model the orientation of the object's hazardous part and its handle. In this way, the robot can perform the handover while keeping the object in a

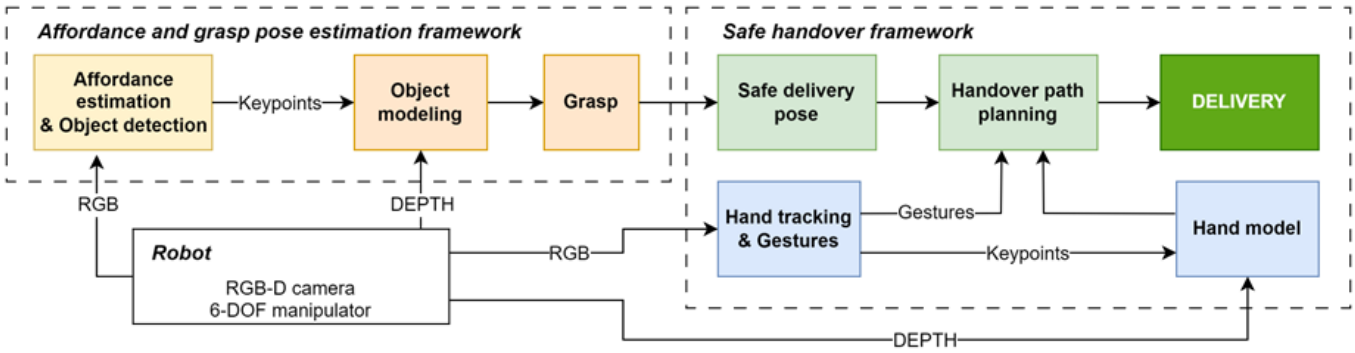

Fig. 1: The overall workflow of the two proposed frameworks.

suitable position, facilitating the subsequent use of the object by the human.

Three affordance labels have been defined: *danger*, *handle*, *grasp*. *Danger* refers to the part of the object used to perform some kind of work, which can represent a threat to the human. *Handle* refers to the object part used by humans to hold it. *Grasp* indicates the part best suited for robot grasping. Figure 2 (a) shows the resulting affordance mask for a knife.

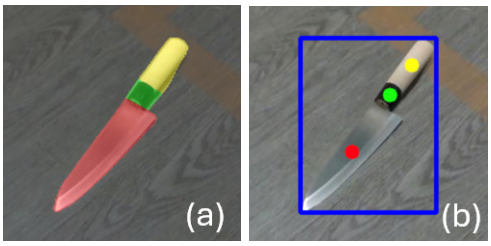

Fig. 2: (a) Affordance mask and (b) affordance keypoints of a knife. Red, green, and yellow marks refer to *danger*, *grasp* and *handle* affordances, respectively.

A. Affordance keypoints

A novel approach for affordance encoding is proposed. The object affordance is represented in a compact way with a keypoint for each affordance label. These keypoints can be obtained from the affordance mask as the average of the coordinates of all pixels that have the same affordance label. Given the coordinates (u, v) of a pixel in the image frame, an affordance function $P(u, v)$ is introduced that associates to the pixel its affordance label among the n predefined ones:

$$P(u, v) = i, \quad i = 1, \ldots, n \tag{1}$$

In our case, three affordance labels have been defined, with $i = 1, 2, 3$, corresponding to *danger*, *handle*, *grasp*, respectively. The pixels are regrouped in n sets A_i, $i = 1, \ldots, n$, of N_i elements each, according to their affordance label:

$$A_i = \{(u, v) : P(u, v) = i\} \tag{2}$$

Then, the coordinates of the keypoint $(\bar{u}_i, \bar{v}_i)$ of each set are computed as:

$$\bar{u}_i = \frac{1}{N_i} \sum_{(u,v)\in A_i} u, \quad \bar{v}_i = \frac{1}{N_i} \sum_{(u,v)\in A_i} v, \tag{3}$$

The resulting keypoints are shown in Figure 2 (b) in the case of a knife.

B. Affordance keypoint estimation network

The estimation of the affordance keypoints is performed using ML over an RGB image of the scene. The network in use is YOLOv8 Pose model, adapted to suit the application needs. The network performs bounding box detection and keypoints estimation at the same time. Together with the bounding box parameters, the output encodes the keypoints as a matrix $K \in \mathbb{R}^{k\times d}$, where k is the number of keypoints $(\bar{u}_i, \bar{v}_i)$ and d is the dimension of each keypoint. In our case $k = 3$ (number of affordance labels) and $d = 2$, since we assume each keypoint to be always visible.

C. Dataset

YOLOv8 Pose model was trained on a custom dataset created from scratch. The dataset contains 7 object classes, featuring common household objects: *razors, paintbrushes, hairbrushes, pens, lighters, screwdrivers* and *knives*. The dataset was built by taking RGB pictures of real objects with an Intel RealSense D435 camera. Each image was manually annotated with the 3 affordance labels using a custom-made tool, and the keypoints were computed as in (3). Since the process of manual image acquisition and annotation was very time-consuming, the dataset was augmented with synthetic images. These images were generated by capturing a photo and employing an ML algorithm to replace the background. Additionally, the visual quality of the resulting images was randomly altered.

D. Affordance-based object modeling and grasp pose estimation

The objects, which the robot is designed to interact with, can be modeled as boxes, as shown in Figure 3 (a). The object's dangerous and handle parts are located on opposite sides along the box's major axis. The boxes' dimensions are fixed and are based on the average dimensions of manipulable objects. The length of the major axis can be optionally obtained as the length of the bounding box's diagonal, projected into three-dimensional space. These approximations will be accounted for during the handover planning by introducing safety offsets in the planned path. The orientation of the box's major axis is obtained from the three-dimensional projection of the grasp keypoints. Therefore, the affordance keypoints implicitly describe the grasp pose. The projection is performed by transforming the keypoints coordinates from the image frame to the camera frame, using the pinhole camera model. The camera frame has origin in the center of the camera focal plane, and axes X_c, Y_c, Z_c. The X_c and Y_c axes lays on the focal plane. The image frame has the origin in the top left corner of the image, and axes u, v normal to the Z_c axis. Since an RGB-D camera is used, the color information $C(u_k, v_k) \in \mathbb{R}^3$ associated to the k-th pixel can be coupled with the corresponding value of the depth map $D(u_k, v_k) \in \mathbb{R}$. Each pixel value of the depth map represents the distance from the camera to the corresponding point in a scene.

Using the keypoints predicted on the color image and the corresponding depth map, we describe the transformation between the two frames as follows:

$$Z_c = D(u, v) \tag{4}$$

$$X_c = (u - cx)\frac{Z_c}{fx} \tag{5}$$

$$Y_c = (v - cy)\frac{Z_c}{fy} \tag{6}$$

where cx, cy, fx and fy are the depth camera intrinsic parameters. Parameters fx and fy are the focal lengths expressed in pixels, while cx and cy indicate the offset in pixels between the origins of the camera and image frames.

In order to plan the grasp, the world frame is defined with axes X_w, Y_w, Z_w. The $X_w - Y_w$ plane is aligned with

the ground. The graspable object is supposed to be laying flat on the ground and a parallel grasp must be performed. Therefore, the gripper grasp pose is defined by a grasp point in world coordinates, an orientation angle α corresponding to the rotation about the Z_w axis and the gripper aperture β. The grasp point is obtained by transforming the grasp keypoint into the world frame. To determine the gripper orientation, the vector δ, which connects the grasp keypoint to the danger keypoint, is calculated; next, the orientation α is computed as the angle between the projection of δ onto the $X_w - Y_w$ plane and the Y_w axis. The gripper aperture β is set as the maximum width of any graspable object. The gripper grasp pose is described in Figure 3 (b).

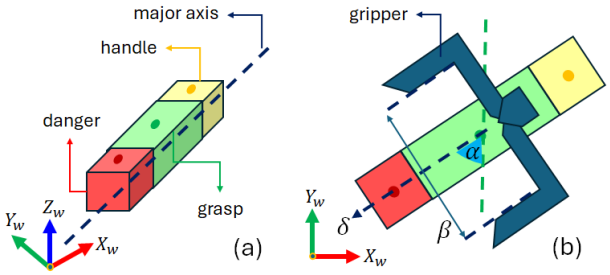

Fig. 3: (a) The object modeled as a box. (b) The gripper grasp pose.

V. Robot-to-Human Handover

Handover involves the robot and human interacting directly with each other, so safe and pleasant interaction must be guaranteed. The robot is able to plan the handover according to the current position of the human hand. Additionally, the object model derived from affordance estimation ensures that the object's handle faces the human and that any hazardous part remains concealed during robot movements. Various handover poses for different applications can be designed accordingly.

A. Hand pose and gesture recognition

The hand pose estimation, inspired by [11], is performed using the MediaPipe hand tracking solution. The solution relies on ML to track 21 hand keypoints from an RGB video stream. Using the camera intrinsic parameters together with the depth information, it is possible to project the hand keypoints into the 3D space (see Section IV-D), thus obtaining the position vector of each keypoint $\boldsymbol{h}_m$, for $m = 1, \ldots, 21$, in the world reference frame. These keypoints, collected in $H = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \mathbf{h}_m\}$, can be used to model the human hand as a sphere, as shown in Figure 4, whose center, defining the hand location, is computed as the geometric center of the subset $H_p \subset H$, containing only the 6 palm keypoints. The sphere's radius is set to approximate the size of an adult human hand. The orientation of the human hand is determined by the computation of the plane defined by 3 out of the 6 palm keypoints. In particular, such a plane is defined by the two vectors connecting the wrist to the base of the index and pinkie fingers; the orientation of the normal to the plane indicates whether the hand palm is facing the camera or not.

MediaPipe also offers the possibility to classify the predicted hand keypoints into 8 common hand gestures. These hand gestures can be assigned to various commands, providing a means of communication with the robot.

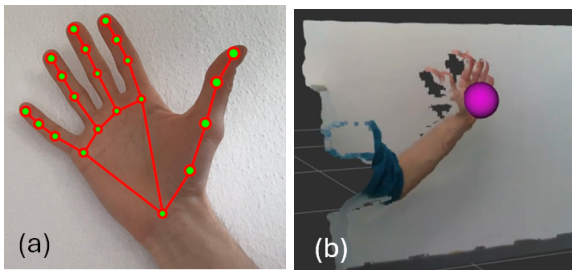

Fig. 4: (a) The predicted hand keypoints over the 2D color image and (b) the hand model in 3D.

B. Handover path and pose

The handover is performed by bringing the object towards the current position of the human hand, whose motion is shown in Figure 5. The robot stops at the *handover point* located at a safety distance from the hand. To maximize the robot reachability, the *handover point* (red) lays on the line (cyan) connecting the center of the manipulator's base to the center of the sphere representing the hand (green). The handover starts after the grasp, with the object already brought in the correct pose for delivery. This pose can be set arbitrarily by using the object's box model. A viable delivery's pose (purple), which prioritizes safety and human grasp convenience, involves positioning the object perpendicular to the ground, with the handle pointing upward and the hazardous part facing downward. This pose should be maintained throughout the entire duration of the arm movement, so that the dangerous part is not exposed. Some path planning constraints are imposed to comply with the safety requirements: (i) a *position constraint* is used to limit the path (yellow) to the line connecting the gripper at the starting position to the *handover point*, and (ii) an *orientation constraint* is used to allow the gripper only to rotate about the Z_w axis of the world frame. In this way, the object is kept in the correct pose, while being aligned with the receiver's hand.

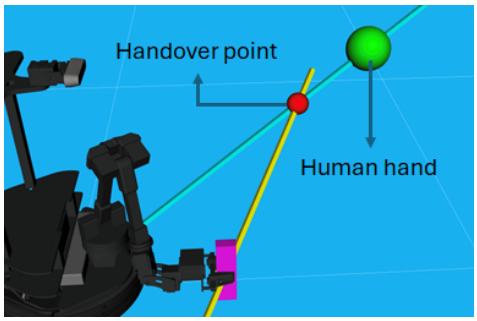

Fig. 5: The handover path and the handover point.

VI. Experimental Validation

Experiments were performed on a LoCoBot WX250 [25], a mobile manipulator with a 6-DOF robotic arm, a parallel

gripper, an RGB-D camera, and a LIDAR. The robot was programmed using ROS1 Noetic with the libraries made available by the robot manufacturer. Motion planning was performed using the MoveIt! framework, including the motion constraints. The overall experimental demonstration is available in a video in [26].

A. YOLOv8 Pose model training and results

YOLOv8 was developed by Ultralytics, which released a simple Python library for model training and inference. Training parameters are very customizable, and multiple feature augmentation options are also available. The model was trained on the custom dataset, containing 3000 pictures. The *mosaicing* feature augmentation option was used to merge multiple object images into a single one. This approach ensures that the model can identify multiple objects in the same picture, even if the original dataset does not contain images of cluttered scenes. The training was performed with a batch size of 16 for 210 epochs, with images resized to 640×640 px. In the last 50 epochs, *mosaicing* was turned off. The learning rate and loss weights were left at their default settings.

Because of the limited dataset, there is some overfitting, making it challenging to accurately evaluate the model's performance. So, to prove the effectiveness of the model, an experiment was designed to showcase performances in a real-world setting. Some objects were laid on the floor, then the robot was maneuvered around them to assess how well it could detect objects and estimate their affordances in real-time. Figure 6 shows the predicted bounding boxes with object classes and affordance keypoints.

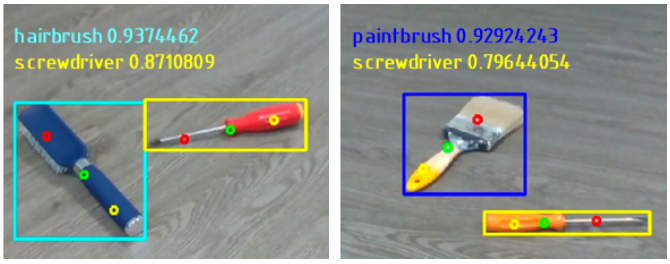

Fig. 6: Real time prediction of YOLOv8 Pose model. Predictions contain object classes, bounding boxes and affordance keypoints.

B. Smart grasping

An experiment was designed to prove the effectiveness of the *affordance and grasp pose estimation framework*. Different objects are positioned in front of the robot to be picked up, and the grasp pose is obtained from a single RGB-D picture. Figure 7 displays the case of a hairbrush: the picture on the left shows the 3D projection over the scene's point cloud, obtained from the image on the left, in which the three 2D affordance keypoints have been marked. Figure 8 illustrates how the robot can leverage this information to adjust its grasp according to various orientations of the target objects.

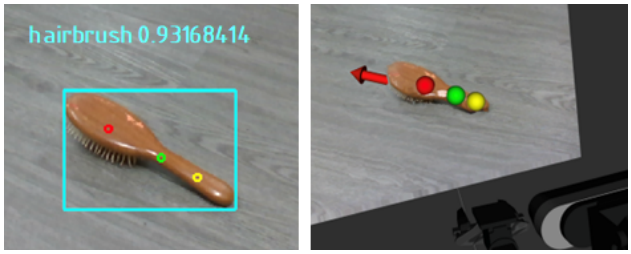

Fig. 7: The 2D affordance keypoints and their 3D projection over the scene's point cloud.

Fig. 8: Two examples of grasp poses for a screwdriver (left) and a hairbrush (right).

C. Object delivery

Both frameworks' capabilities were tested in a comprehensive handover experiment: an object is initially placed in front of the robot, which then grasps it and delivers it to a human. The implementation involves multiple ROS nodes that execute the actions detailed in Figure 9. The process begins with (a) taking a picture of the scene, detecting the object, and estimating its affordances. Subsequently, (b) the robot performs the grasp, and (c) moves the object to a safe position, orienting it with the handle towards the human. Next, (d) the camera tilts upwards to face the human, and the robot initiates real-time tracking of the human hand and classification of gestures. Upon detecting the *open palm* gesture, (e) the robot fixes the last known hand position and computes the *handover point*. The handover process starts, with the object being transported to the *handover point*. Throughout the handover motion, the robot adheres to safety constraints, following the path described in Section V-B. Finally, (f) the object is presented to the human with the correct orientation.

VII. Conclusions and Future Works

This paper presents a comprehensive system designed for safe task-oriented handovers. Developed within ROS1, it has been tested on a low-cost mobile manipulator with limited computational resources and sensor capabilities. Using an RGB image, the robot employs the YOLOv8 Pose model to identify previously unseen objects and estimate their affordance. The affordance is encoded using keypoints, enabling the creation of a 3D model that considers object handle placement and hazardous areas. This model guides grasp

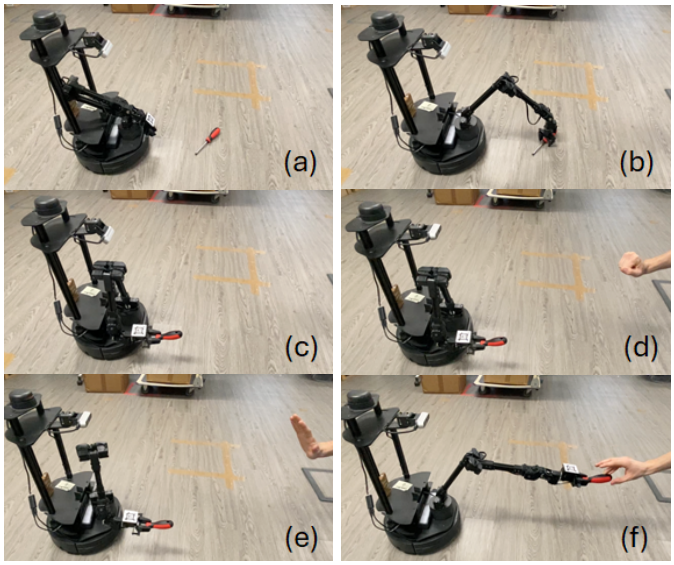

Fig. 9: A comprehensive robot-to-human object delivery, with grasp and handover.

planning and ensures appropriate object orientation during handover to humans. MediaPipe is used to perform hand tracking, allowing the robot to adjust delivery based on human needs. Gesture signals indicate human readiness to receive the object, and the robot adjusts the handover location, as well as its movements to prevent contact with hazardous parts. Future developments may focus on enhancing interactivity, by dynamically modifying the handover paths to account for hand movements. Moreover, expanding the dataset utilized to train YOLO within the *affordance pose and grasp estimation framework* would enhance the robot's capabilities, mitigate overfitting and enable accurate performance measurement.

ACKNOWLEDGEMENTS

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

REFERENCES

[1] V. Ortenzi, A. Cosgun, T. Pardi, W. P. Chan, E. Croft, and D. Kulić, "Object handovers: a review for robotics," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1855–1873, 2021.

[2] V. Ortenzi, M. Controzzi, F. Cini, J. Leitner, M. Bianchi, M. A. Roa, and P. Corke, "Robotic manipulation and the role of the task in the metric of success," *Nature Machine Intelligence*, vol. 1, no. 8, pp. 340–346, 2019.

[3] H. Duan, Y. Yang, D. Li, and P. Wang, "Human–robot object handover: Recent progress and future direction," *Biomimetic Intelligence and Robotics*, vol. 4, no. 1, p. 100145, 2024.

[4] V. Ortenzi, M. Filipovica, D. Abdlkarim, T. Pardi, C. Takahashi, A. M. Wing, M. Di Luca, and K. J. Kuchenbecker, "Robot, Pass me the tool: Handle visibility facilitates task-oriented handovers," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2022, pp. 256–264.

[5] W. P. Chan, M. K. Pan, E. A. Croft, and M. Inaba, "Characterization of handover orientations used by humans for efficient robot to human handovers," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1–6.

[6] D. Lehotsky, A. Christensen, and D. Chrysostomou, "Optimizing robot-to-human object handovers using vision-based affordance information," in *2023 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2023, pp. 1–6.

[7] V. Ortenzi, F. Cini, T. Pardi, N. Marturi, R. Stolkin, P. Corke, and M. Controzzi, "The grasp strategy of a robot passer influences performance and quality of the robot-human object handover," *Frontiers in robotics and AI.*, vol. 7, 2020-10-19.

[8] J. Aleotti, V. Micelli, and S. Caselli, "Comfortable robot to human object hand-over," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 771–776.

[9] ——, "An affordance sensitive system for robot to human object handover," *International journal of social robotics.*, vol. 6, no. 4, pp. 653–666, 2014-11.

[10] M. Qin, J. Brawer, and B. Scassellati, "Task-oriented robot-to-human handovers in collaborative tool-use tasks," in *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2022.

[11] C. Meng, T. Zhang, and T. l. Lam, "Fast and comfortable interactive robot-to-human object handover," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3701–3706.

[12] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5031–5041.

[13] A. Vakunov, C.-L. Chang, F. Zhang, G. Sung, M. Grundmann, and V. Bazarevsky, "Mediapipe hands: On-device real-time hand tracking," in *CVPR Workshop*, 2020.

[14] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[15] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3D affordancenet: A benchmark for visual object affordance understanding," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1778–1787.

[16] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6D object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[17] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[18] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics (T-RO)*, 2023.

[19] W. Chen, H. Liang, Z. Chen, F. Sun, and J. Zhang, "Learning 6-DOF task-oriented grasp detection via implicit estimation and visual affordance," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 762–769.

[20] A. D. Christensen, D. Lehotskỳ, M. W. Jørgensen, and D. Chrysostomou, "Learning to segment object affordances on synthetic data for task-oriented robotic handovers," in *The 33rd British Machine Vision Conference*. British Machine Vision Association, 2022.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[22] P. D. Cen Cheng, F. Sibona, and M. Indri, "A framework for safe and intuitive human-robot interaction for assistant robotics," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2022, pp. 1–4.

[23] C. L. Blengini, "Smart mobile manipulation for human-robot assistive applications," Master's thesis, Politecnico di Torino, 2024.

[24] "Github repository," https://github.com/celubi/affordance_based_handover_grasping.git, [Online; accessed May 2024].

[25] T. Robotics, "LoCoBot WX250 with 6 DOF Arm (with Lidar)," [Accessed February 2024]. [Online]. Available: https://docs.trossenrobotics.com/interbotix_xslocobots_docs/specifications/locobot_wx250s.html

[26] "Experimental test video demo," https://youtu.be/xZjKYFkRZZg, [Online; accessed May 2024].