Fast Perception for Human-Robot Handovers with Legged Manipulators

Author(s):
Tulbure, Andreea; Abi-Farraj, Firas; Hutter, Marco

Publication date:
2024-03

Permanent link:
https://doi.org/10.3929/ethz-b-000652195

Originally published in:
https://doi.org/10.1145/3610977.3634958

Funding acknowledgement:
188596 - Perceptive Dynamic Locomotion on Rough Terrain (SNF)

Fast Perception for Human-Robot Handovers with Legged Manipulators

Andreea Tulbure
Robotic Systems Lab, ETH Zurich
Zurich, Switzerland
tulbure@mavt.ethz.ch

Firas Abi-Farraj
Robotic Systems Lab, ETH Zurich
Zurich, Switzerland
firas@enchanted.tools

Marco Hutter
Robotic Systems Lab, ETH Zurich
Zurich, Switzerland
mahutter@ethz.ch

ABSTRACT

Deploying perception modules for human-robot handovers is challenging because they require a high degree of reactivity, generalizability, and robustness to work reliably for a diversity of cases. Further complications arise as each object can be handed over in a variety of ways, causing occlusions and viewpoint changes. On legged robots, deployment is particularly challenging because of the limited computational resources and the image-space noise resulting from locomotion.

In this paper, we introduce an efficient and object-agnostic real-time tracking framework, specifically designed for human-to-robot handover tasks with a legged manipulator. The proposed method combines optical flow with Siamese-network-based tracking and depth segmentation in an adaptive Kalman Filter framework. We show that we outperform the state-of-the-art for tracking during human-to-robot handovers with our legged manipulator. We demonstrate the generalizability, reactivity, and robustness of our system through experiments in different scenarios and by carrying out a user study. Additionally, as timing is proven to be more important than spatial accuracy for human-robot handovers, we show that we reach close to human timing performance during the approaching phase, both in terms of objective metrics and subjective feedback from the participants of our user study.

CCS CONCEPTS

• **Computer systems organization → Robotics**.

KEYWORDS

legged robotics, physical human-robot interaction, human-robot handover

ACM Reference Format:
Andreea Tulbure, Firas Abi-Farraj, and Marco Hutter. 2024. Fast Perception for Human-Robot Handovers with Legged Manipulators. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3610977.3634958

HRI '24, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0322-5/24/03…$15.00
https://doi.org/10.1145/3610977.3634958

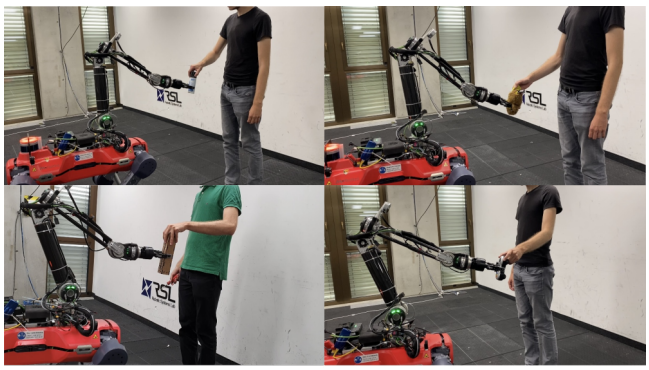

Figure 1: *ANYmal with the 6-DoF Dynaarm* performing human-to-robot handovers with four different objects.

1 INTRODUCTION

Human-robot handover is a challenging collaborative manipulation task with three different phases: approach, object transfer, and post-handover [25]. Each phase requires motion coordination, robust and real-time perception, and quick reaction to changes, regardless of the environment and the object that is being passed. This work focuses on the scenario where a human (giver) hands over an object to a robot (receiver). Simple approaches for human-to-robot handovers require the human to place the object in a stationary robot gripper [3, 18]. While such pragmatic approaches work in simple cases, they do not reflect the 'natural' motion coordination when handing over an object. Moreover, in assistive scenarios, where the human range of motion is constrained, it is essential that the robot actively contributes to the handover process.

Even though significant progress has been made in enabling seamless autonomous human-to-robot handovers [25], one unsolved challenge remains the visual perception during the approach phase. Existing works focus on fixed manipulators, yet several real-life handover scenarios (e.g. fetching an object from another room) require a mobile platform. While several mobile robots exist, legged robots have become increasingly popular as progress in quadrupedal locomotion [22] shows that such systems are ready to be deployed among humans and perform collaborative tasks. Engaging in physical human-robot interaction with a legged platform comes with advantages, such as navigating on rough terrain and extending the reach of the manipulator via whole-body motion. It comes, however, with some perception challenges: 1) having to use on-board cameras only, which leads to significant viewpoint changes and image-space noise caused by locomotion; 2) limited battery capacity and payload (and hence, limited computational power). Therefore, existing algorithms for the approach phase are not directly transferable to legged manipulators.

2 RELATED WORK

Some works assume that geometric [21] or visual [10] a priori knowledge of the object is available. Demonstrating some degree of generalizability to novel objects, Chang et al. [8] introduce an end-to-end RL-based grasping approach. Using pixel-wise affordance predictions to infer possible gripper positions and angles, they show that their method outperforms the state-of-the-art in terms of grasp success for the objects they trained on, especially for high occlusions. However, the performance decreases for novel objects.

Other works restrict the a-priori knowledge to the class information of the object. Sanchez-Matilla et al. [29] extract the centroid and dimensions of cups in real-time using an object segmentation network specifically trained on the object class. The proposed setup uses two external cameras to monitor the workspace. The 3D centroid of the object is obtained by triangulating the 2D centroids generated in each image. In a more practical setup, Rosenberger et al. [28] use a generic object detector on images from a wrist-mounted camera to detect the object in the hand. A body and hand segmentation network is used to remove the points belonging to the human from the point cloud of the bounding box. The occlusions of the object by the human hand are limited and the human is constrained to not move after the robot starts moving. This impacts negatively the fluency of the handover. Although class information is less constraining for handovers, the objects have to be within the classes known by the detector, and occlusion handling is limited.

To tackle human-to-robot handovers of unknown objects, Yang et al. [35] use a body skeleton tracker on an image from a conveniently mounted external camera to detect the human hand. A 3D bounding box of predefined size is cropped around it and the resulting point cloud is used to classify the human grasp types into one of seven predefined categories. The robot trajectory is adjusted depending on the label. Building up upon this framework, the same authors introduce a pixel-wise hand segmentation method to assign the points in the cropped 3D bounding box to the human hand or the object [36]. This way they obtain an object point cloud which is used to generate grasps. Their vision framework runs at 9 Hz and the grasps are generated with 5 Hz. This perception framework is used in several works to develop better-performing control strategies [9, 37]. Taking advantage of recent developments in joint hand and object detection, unknown-object handovers with anthropomorphic hands have been considered by Duan et al. [11]. Their system can handle object occlusions and generalize to unknown objects, but significant processing time is required for image processing and grasp generation, which impacts negatively the handover time.

Overall, related work on perception of unknown objects during the approach phase of human-to-robot handovers concentrates on fixed manipulators and computationally intensive perception frameworks, with conveniently mounted external cameras monitoring the workspace. Even though the results are promising [9, 36, 37], there is still a gap to human-human handover times [15]. In fact, the perception pipeline in those works runs relatively slowly (processing time 100 ms [36]) compared to the processing time for visual stimuli in the human brain ($20 - 40$ ms [16]). We believe that reducing the visual perception processing times to values closer to those reported for humans can close the current performance gap between human-human and human-to-robot handover timing.

In this paper, we present a fast visual perception algorithm for legged systems which accounts for the high image-space noise, view-point changes, and limited computational power. We aim to enable a legged manipulator to successfully carry out human-to-robot handover tasks while achieving close-to-human timing performance during the approach phase. Our contributions are three-fold: i) A computationally cheap and object-agnostic tracking framework, which deals with partial object occlusions and viewpoint changes and runs at speeds comparable to which visual stimuli reach the human brain [16]; ii) A detailed user study on human-to-robot handovers with a legged manipulator and a thorough comparison to human handover and reaction times from independent studies; iii) The first open-source human-to-robot handover dataset from a legged manipulator with manually labeled 2D bounding boxes [1].

3 METHODOLOGY

A schematic of the proposed tracking and grasping framework is depicted in Fig. 2. For tracking, we use an adaptive Kalman Filter framework to fuse 2D bounding box measurements from different sources running at different frequencies. To achieve object-agnostic tracking, we rely on SiamRPN [20] as a main component. It considers tracking as a matching problem between an initial bounding box, called a template, of the object and the search region in the current image. Hence, solely a bounding box for initialization is required without any knowledge about the object being tracked. This bounding box is obtained from the object detection step, which runs once at the beginning and can be re-triggered if the object is lost. The disadvantage of such trackers is that they are computationally intensive and can only run at a low frequency on commonly available compute on legged robots. To overcome these limitations, we integrate measurements from faster optical flow to correct for the displacements that appear between capturing the image and finishing the network inference. Additionally, we exploit depth segmentation both to improve the 2D tracking, especially when the object is close, and to get its point cloud, needed for grasp planning.

It is important to note that the presented framework is modular and its components can be replaced depending on the advances of the state-of-the-art algorithms and available computing power.

3.1 Object detection

We employ a two-stage object detection strategy. In the first step, a pre-trained Yolov7 network [33] is used. If nothing is detected we assume occlusions by the hand or unknown objects. Therefore, in the second step, we use the Openpose skeleton tracker [7] to draw a bounding box around the human wrist marker and use it for initialization. Note that the object to be handed over has to be at least partially visible in the first frame.

3.2 Object-agnostic tracking framework

After initializing each component with the bounding box from object detection the tracking is started. In this section, we detail the more accurate but delayed Siamese-network-based tracker, the faster optical flow, and depth segmentation algorithms and explain how we fuse them in our adaptive Kalman Filter framework.

[1] available at https://u.ethz.ch/Uuknf

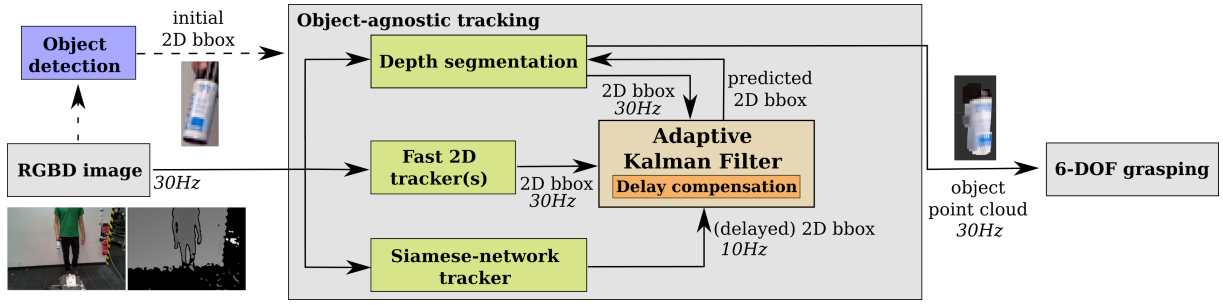

Figure 2: Overview of the proposed framework. The gray boxes depict the different components of the framework: input (RGBD image), tracker, and grasping algorithm. The dotted arrows show that the object detection is triggered and the bounding box is passed to the tracking framework (at initialization or when the object is lost).

3.2.1 Lukas-Kanade tracker. We calculate sparse optical flow using an iterative Lucas-Kanade method with pyramids [6]. This is a fast 2D feature-based tracker, which we use to bridge the gap between the more accurate, but slower measurements of the Siamese-network-based tracker. During initialization, we detect good features in the initial bounding box of the object. We make sure that the detected 2D features are efficiently distributed over the object using [4]. These features are tracked and scored by their quality. Features with low scores are re-initialized, while the ones with good scores are used to compute the homography between the new and old features. The resulting homography transformation is applied to the previous bounding box to get a new measurement. If no good features can be detected or other anomalies occur, the bounding box is reinitialized with the current filter estimate.

3.2.2 Depth segmentation. To enhance image-based tracking with depth information, we use the geometric depth segmentation introduced in [12], from which we get the segment corresponding to the object. We back-project this segment to 2D and compute the minimum oriented bounding box on the pixels corresponding to the object and feed this as a measurement to the Kalman Filter.

To reduce computational time, we perform this segmentation in the region of interest (ROI) only, which is computed from the previous bounding box of the depth segmentation, padded with a specific number of pixels. The amount of padding is computed based on the maximum displacement in image space for a maximum 3D velocity of 1 m/s between the robot and the human [15]. If the ROI diverges from the bounding box predicted by the filter, we reset it with the current filter estimate. We use the intersection over union (IoU) of the ROI and the filter estimate to detect divergence.

3.2.3 Siamese-network-based tracker. In a handover scenario, the appearance of the tracked object might change for various reasons, e.g. the human re-grasping or re-orienting the object. Hence, the initial bounding box or template is not enough for accurate tracking. This problem is addressed in [30], where the authors introduce a framework for tracking holistic object representations (THOR), which includes long-term (LTM) and short-term memory (STM) buffers consisting of RGB templates of the object. The goal of the LTM is to memorize the object in diverse conditions (e.g. lighting) to improve re-detection and long-term tracking, while STM handles short-term variations (e.g. partial occlusions).

The STM is updated every k-th iteration, using a first-in-first-out strategy. The updated template corresponds to the search area at that timestep. For all templates in this buffer a diversity measure γ is computed. This diversity measure is used to decide which of the templates in the STM are added to the LTM, which contains the M most diverse templates of the same object. To get the final predicted bounding box, bounding boxes with the highest score in STM and LTM are chosen as candidates and an intersection over union (IoU) operation between these two is applied to choose the best fit. If the value is above a given threshold the STM prediction is used. Otherwise, the LTM prediction is used and the STM is initialized.

3.2.4 Kalman Filter. The Kalman Filter state at time step k is defined as $\mathbf{x_k} = \{\mathbf{p_k}, \mathbf{v_k}, w_k, h_k\} \in \mathbb{R}^6$, where $\mathbf{p_k} \in \mathbb{R}^2$ is the center of the bounding box, $\mathbf{v_k} \in \mathbb{R}^2$ its velocity in image space, w_k and h_k its width and height. Each measurement $\mathbf{y_k} \in \mathbb{R}^4$ contains the center position, width, and height of the bounding box. The state space model is:

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{A}\mathbf{x_k} + \boldsymbol{q_k} \\ \mathbf{y'_k} &= \mathbf{H'_k}\mathbf{x_k} + \boldsymbol{r'_k}, \end{aligned} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{6x6}$ is the the state transition matrix, $\mathbf{H_k}'$ is the measurement matrix, $\boldsymbol{q_k} \sim N(0, \mathbf{Q_k})$ and $\boldsymbol{r'_k} \sim N(0, \mathbf{R'_k})$ are prediction and measurement noise, respectively. We consider a constant velocity model, assuming w_k and h_k are constant over the horizon. For the measurement update, we use the augmented observation model from [13] and concatenate all available measurements at time k :

$$\begin{aligned} \mathbf{y'_k} &= [\mathbf{y_{1_k}}, \mathbf{y_{2_k}}, \mathbf{y_{3_k}}]^T \\ \mathbf{H'_k} &= [\mathbf{H_1}, \mathbf{H_2}, \mathbf{H_3}]^T \\ \mathbf{R'_k} &= \mathrm{diag}[\mathbf{R_{1_k}}, \mathbf{R_{2_k}}, \mathbf{R_{3_k}}], \end{aligned} \tag{2}$$

with $\mathbf{H_1} = \mathbf{H_2} = \mathbf{H_3}$. Note that $\mathbf{H'_k}$ has different dimensions $\in \mathbb{R}^{12x6}$ or $\in \mathbb{R}^{8x6}$, depending on the availability of $\mathbf{y_{3_k}}$, the measurement of the Siamese-network-based tracker.

Time-delay compensation: In our framework, there is a significant time delay when the measurement from THOR arrives, i.e. the measurement at time k is actually from time $s = k - n$:

$$\mathbf{y^d_{3_k}} = \mathbf{H_3}\mathbf{x_s} + \boldsymbol{r_{3_k}}. \tag{3}$$

One straightforward strategy to deal with this issue is to recalculate the filter each time a delayed measurement arrives. However, this is computationally intensive and we aim to exploit cheaper methods introduced in literature [5, 19]. The idea is based on the extrapolation of the delayed measurement to the present time using the difference between current and past estimates of the filter and

computing an optimal gain for the extrapolated measurement. If the state from time s is stored and we extrapolate the measurement by computing the difference between the estimates at time k and s, the extrapolated measurement at time k becomes:

$$\mathbf{y}_{3_k}^{\mathbf{exp}} = \mathbf{y}_{3_k}^{\mathbf{d}} + \mathbf{H_3}\hat{\mathbf{x}}_{\mathbf{k}}^{+} - \mathbf{H_3}\hat{\mathbf{x}}_{\mathbf{s}}^{+} \tag{4}$$

where $\hat{\mathbf{x}}^{+}$ denotes the estimates after the measurement update at the given times. Replacing 3 in 4 we get: $\mathbf{y}_{3_k}^{\mathbf{exp}} = \mathbf{H_3 x_s} + \boldsymbol{r}_{3_k} + \mathbf{H_3}\hat{\mathbf{x}}_{\mathbf{k}}^{+} - \mathbf{H_3}\hat{\mathbf{x}}_{\mathbf{s}}^{+}$. This can be reformulated into:

$$\mathbf{y}_{3_k}^{\mathbf{exp}} = \mathbf{H_3 x_k} + \boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{exp}}, \tag{5}$$

where $\boldsymbol{r}_{\boldsymbol{k}}^{\boldsymbol{exp}} = \boldsymbol{r}_{3_k} + \mathbf{H_3}\boldsymbol{\delta x_k} - \mathbf{H_3}\boldsymbol{\delta x_s}$ is the measurement noise for the corrected measurement and $\boldsymbol{\delta x_i} = \hat{\mathbf{x}}_{\mathbf{i}}^{+} - \mathbf{x_i}$ denotes the error between the estimated and true state at time i. It can be seen that the measurement noise depends on the state $\mathbf{x_s}$. Hence, to ensure optimality, a new filter gain $\mathbf{K_k}$ for fusing $\mathbf{y}_{3_k}$ is derived by minimizing the covariance of the estimation errors:

$$\mathbf{K_k} = \mathbf{M}^T \mathbf{H_3}^T [\mathbf{H_3 P_s H_3}^T + \mathbf{R}_{3_k}]^{-1}, \tag{6}$$

where $\mathbf{P_s}$ is the measurement covariance matrix of the delayed measurement, $\mathbf{M} = \prod_{i=0}^{n-1} (\mathbf{I} - \mathbf{K_{k-i}} \mathbf{H'_{k-i}}) \mathbf{A_{k-i-1}}$ and n is the number of steps by which the measurement is delayed. This gain is used in the measurement update step to fuse the delayed measurement. The updated estimation covariance after the measurement update is $\mathbf{P_k^+} = \mathbf{P_k} - \mathbf{K_k H_3 M}$, with $\mathbf{P_k}$ being the previous estimation covariance of the filter. The full derivation is presented in [19].

Measurement noise adaptation: Depending on the situation, we want to adapt our confidence for the different measurement sources, e.g. for fast movements optical flow is more accurate than the THOR prediction. For THOR, the confidence depends on the estimated velocity, for the depth segmentation on the distance to the object, and for optical flow on the blur measure (computed using the variance of the Laplacian [27]). To ensure smoothness we use a fading factor α, as in [2]:

$$\mathbf{R'_k} = \alpha \mathbf{R'_{k-1}} + (1 - \alpha) \mathbf{R}^{*\prime}_{\mathbf{k}}, \tag{7}$$

with $\mathbf{R}^{*\prime}_{\mathbf{k}}$ the new measurement covariance matrix determined experimentally.

Note that as the depth segmentation uses the previous bounding box from the algorithm itself to compute the ROI for segmentation, we assume that the covariances of measurement and process noise are uncorrelated.

Process noise adaptation: Since our Kalman Filter model does not describe the process well at all times, e.g. when the human suddenly changes the approaching direction, we employ an adaptation method for estimating the process noise online [2]. By reformulating the process equation in Eq. 1 we get $\boldsymbol{q_k} = \mathbf{x_{k+1}} - \mathbf{A x_k}$, and hence:

$$\hat{\boldsymbol{q}}_{\boldsymbol{k}} = \hat{\mathbf{x}}_{\mathbf{k+1}}^{+} - \mathbf{A}\hat{\mathbf{x}}_{\mathbf{k}}^{+} = \hat{\mathbf{x}}_{\mathbf{k+1}}^{+} - \hat{\mathbf{x}}_{\mathbf{k+1}}^{-}, \tag{8}$$

where $\hat{\mathbf{x}}_{\mathbf{k+1}}^{-}$ is the estimate after the process update of the filter. Knowing that the posterior estimate $\hat{\mathbf{x}}_{\mathbf{k}}^{+} = \hat{\mathbf{x}}_{\mathbf{k}}^{-} + \mathbf{K_k}\boldsymbol{\epsilon_k}$, we get $\hat{\boldsymbol{q}}_{\boldsymbol{k}} = \mathbf{K_k}\boldsymbol{\epsilon_k}$ with its covariance

$$E\{q_k q_k^T\} = \mathbf{K_k} E\{\boldsymbol{\epsilon_k} \boldsymbol{\epsilon_k}^T\} \mathbf{K_k}^T, \tag{9}$$

where $E\{\boldsymbol{\epsilon_k} \boldsymbol{\epsilon_k}^T\} = \mathbf{R'_k} + \mathbf{H'_k P_k^- H'_k}^T = \mathbf{S_k}$ is the covariance of the innovation term. The adapted process covariance is:

$$\mathbf{Q}^{*}_{\mathbf{k+1}} = \mathbf{K_k S_k K_k}^T. \tag{10}$$

Similar to the measurement covariance adaptation, a forgetting factor β is introduced: $\mathbf{Q_{k+1}} = \beta \mathbf{Q_k} + (1 - \beta)\mathbf{Q}^{*}_{\mathbf{k+1}}$.

3.3 6-DOF grasping

The grasping pipeline is based on primitive shape fitting on the object point cloud similar to [23]. The point cloud is obtained from the object segment output by the depth segmentation introduced in 3.2.2 and post-processed to find the best-fitting cuboid. Other primitive shapes, e.g. spheres or cylinders, can be used as well.

We use a particle filter with a constant velocity model in the 3D space for estimating the pose, size, and score of the best-fitting box. Its state is $\mathbf{x_k} = [\mathbf{p_k}, \mathbf{q_k}, \mathbf{d_k}, \mathbf{v_k}, \boldsymbol{\omega_k}, s_k] \in \mathbb{R}^{17}$, with $\mathbf{p_k} \in \mathbb{R}^3$ and $\mathbf{q_k} \in \mathbb{R}^4$ being the 3D position and orientation of the box in the reference frame defined as a quaternion, $\mathbf{d_k} \in \mathbb{R}^3$ is the 3D dimension, $\mathbf{v_k}$ and $\boldsymbol{\omega_k}$ are the linear and angular velocities of the center of the box and s_k is a similarity score between the point cloud and the computed best-fitting box. The state space is:

$$\begin{aligned} \mathbf{x_k} &= f(\mathbf{x_{k-1}}, \boldsymbol{\nu_{k-1}}) \\ \mathbf{y_k} &= h(\mathbf{x_k}, \boldsymbol{r_k}), \end{aligned} \tag{11}$$

with $\boldsymbol{\nu_{k-1}}$ and $\boldsymbol{r_k}$ Gaussian process/measurement noise and $\mathbf{y_k} = [\mathbf{p_k}, \mathbf{q_k}, \mathbf{d_k}, s_k] \in \mathbb{R}^{11}$ the measurements. The velocities, size, and score are considered constant during the horizon. The position update is $\mathbf{p_{k+1}} = \mathbf{p_k} + \mathbf{v_k}\Delta T$, while the updated quaternions $\mathbf{q_{k+1}} = \mathbf{q_k} + \frac{1}{2}[\boldsymbol{\omega_k}]_x \mathbf{q_k} \Delta T$, with:

$$[\boldsymbol{\omega_k}]_x = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0x & -\omega_z & \omega_y \\ \omega_y & \omega_z & 0 & -\omega_x \\ \omega_z & -\omega_y & \omega_x & 0 \end{bmatrix} \tag{12}$$

where $\omega_x, \omega_y, \omega_z$ are the angular velocities along the axes.

We use the most likely predicted box to generate a grasp candidate by matching the end-effector orientation with the box orientation. The position is constrained to the lower object half to avoid full object occlusions by the gripper and grasping human fingers.

4 EXPERIMENTAL VALIDATION

We analyze the performance of the proposed tracker and the handover performance of our system separately, as the latter requires significant integration of hardware, control, motion planning, and perception. For the experiments, we use our ANYmal equipped with a custom-made torque-controlled 6-DoF robotic arm (Fig. 3), but it can be deployed on any mobile manipulator. The tracker runs on the onboard NVIDIA Jetson Xavier AGX [1] at 30 Hz. For high-level control, we use a state machine with four states:

(1) *Walk base to object*: the base moves towards the object
(2) *Move end-effector to object*: the object is within reach and end-effector moves towards it
(3) *Grasp object*: the object is close, and the gripper grasps it
(4) *Wait*: standing still, waiting for detection

The state depends on the end-effector to object distance. As we use a fixed camera mount on the arm, the end-effector orientation is fixed at a convenient location while approaching (until *Move end-effector to object*-state) to avoid significant object occlusions by the gripper. The desired end-effector poses are the grasp candidates

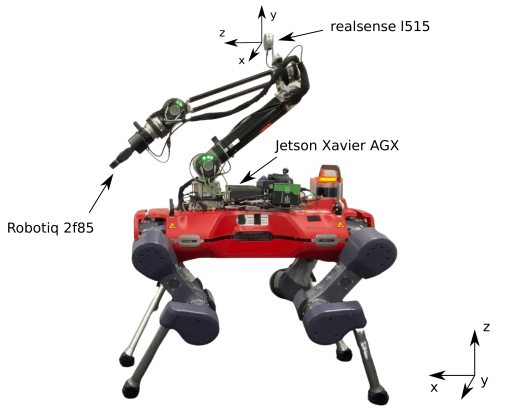

Figure 3: *ANYmal with the 6-DoF Dynaarm* equipped with a Robotiq 2F-85 gripper and a realsense l515 lidar on the arm.

generated from Sec. 3.3. They are used as targets for the whole-body MPC motion planner from [31]. The weights are tuned and the velocity is capped to satisfy the human comfort criteria in [26], e.g. peak speed of 1.5 m/s.

4.1 Tracking performance analysis

To validate the tracking framework the following scenarios are considered:

- *simple* handover: human hands over non-adversarially and with minimum occlusions different objects: bottle, block, joystick, toy. The object orientation can vary.
- *fast* handover: human moves the object fast into the end-effector of the robot with minimum occlusion.
- *occlusion* handover: human partially occludes the object during non-adversarial handover.
- *adversarial* handover: human changes the object pose when the object to end-effector distance is < 30 cm, forcing the robot to adjust quickly. Occlusions are minimal.

We consider simple handovers to evaluate the ability to handle different objects. The other scenarios are chosen to evaluate the impact of velocity, occlusion, and adversarial motion, which we consider object-agnostic. We open-source the dataset containing manually labeled 2D ground truth bounding boxes of sixteen handovers from all scenarios recorded on the robot. As the whole handover sequence is recorded, the dataset contains images from trotting, when the base moves towards the human, and during stance with only the arm moving, after the object is within its reach. We use this dataset for state-of-the-art comparison and to perform an ablation study.

4.1.1 Comparison to state-of-the-art. We compare latency and 2D/3D accuracy to baseline state-of-the-art algorithms capable of running in real-time on our platform, namely, Yolov7 [33] with a Kalman Filter (Yolov7-f), THOR [30] and STARK [34]. Yolov7 and STARK run at 30 Hz, while THOR [30] runs at 10 Hz, a frequency comparable to state-of-the-art perception frameworks for human-to-robot handovers on fixed manipulators [36]. Such frameworks are based on skeleton trackers and even though they run in real-time on our system, the tracking fails when the human shoulders are not visible in the image, making it impractical for our setup.

For state-of-the-art comparison, we consider the sets containing the handover of the bottle to avoid retraining Yolov7 on custom classes. The detection threshold is set to 0.2 which we empirically observed leads to the best results. It is important to note that Yolov7-f is limited to a predefined set of classes, while the other methods can track any object in the hand. Nevertheless, we use Yolov7-f as a baseline because of its widespread usage in mobile manipulation.

Table 1: State of the art comparison using the bottle only

Dataset	Method	IoU	min IoU	P	*n*CLE
Simple	Yolov7-f	71.17	24.70	91.13	0.75
	THOR	63.69	17.80	82.15	0.78
	STARK	71.22	31.04	90.41	0.84
	ours	**71.93**	**34.15**	**92.37**	**0.67**
Fast	Yolov7-f	65.50	25.21	85.89	0.78
	THOR	59.28	8.40	69.32	0.82
	STARK	65.78	0	86.46	0.98
	ours	**66.79**	**25.82**	**86.79**	**0.69**
Occlusion	Yolov7-f	52.76	0.0	53.60	0.88
	THOR	47.92	0.0	31.82	0.83
	STARK	55.51	0.0	60.21	0.80
	ours	**58.22**	**10.97**	**65.17**	**0.78**
Adversarial	Yolov7-f	65.24	14.96	70.23	0.83
	THOR	55.20	15.91	63.22	0.82
	STARK	54.98	2.52	62.75	0.96
	ours	**66.54**	**31.19**	**73.60**	**0.68**

2D Comparison: For comparison in the image space, we use the following metrics: intersection over union (IoU), precision (P) with true positives (TP) when $IoU > 50$ % and RMSE of the normalized center location error (*n*CLE) [38]. We additionally consider minimum IoU (min IoU), which we define as the minimum value of the IoU among all the experiments in the specific scenario.

The results are presented in Table 1. We notice that our tracker outperforms THOR, while Yolov7-f has a close performance in terms of IoU, except for the occlusion scenario. There the detections are rare and the Kalman Filter can not properly capture the motion in the image. As in the adversarial scenario, the object is grasped on the top, the occlusions by the human hand are small and the performance of Yolov7-f is comparable to ours. STARK has a similar performance to our method, except for the adversarial case. It handles well object occlusions but can not deal with sudden changes in the direction of motion. Overall, the min IoU is higher in all the scenarios for our method, underlining its robustness. Furthermore, the impact of the object velocity with respect to the robot base can be noticed by observing the performance difference between the fast and simple scenarios. In the simple scenario, the object speed reaches 0.62 m/s, while in the fast scenario, it goes up to 1.06 m/s.

3D Comparison: Using the same dataset, we analyze the 3D object position error and show the results in Fig. 4. The position error is computed in the camera frame using $e = ||\hat{x} - x_r||$, where $\hat{x}$ is the 3D object center from depth segmentation and x_r is the reference, defined as the 3D projection of the center of the ground truth bounding box. The errors in the simple and fast scenarios are similar. There is no substantial difference between the baselines and our method, except when the object-gripper distance is < 5 cm. This is expected, as the smaller this distance, the bigger the occlusion by the robot. For the other scenarios, the differences are more

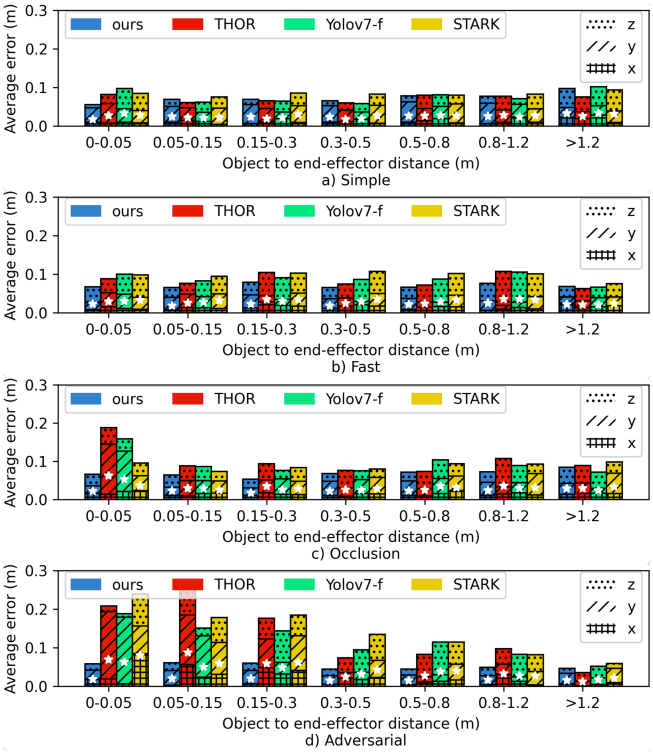

Figure 4: Object position error with respect to the reference object-end-effector distance in the camera frame, in a) simple, b) fast, c) occlusion, and d) adversarial scenarios. The star represents the position error average over all axes.

meaningful. With occlusions, the errors of the baselines increase when very close (< 5 *cm*). For THOR and Yolov7-f the error goes up to $15 - 20$ *cm*, while for STARK it reaches 10 *cm*. This can be explained by the high degree of occlusion, leading to detections that are off and just part of the object being visible in the image. For the adversarial scenario, the baseline errors increase significantly when the object-gripper distance is < 30 *cm*. The main error comes again from the y-axis, which can be explained by the occlusion. Furthermore, the sudden movement direction changes cause issues for the slower THOR baseline, but also for STARK. For THOR the lateral error goes up to 5 *cm* but decreases when the end-effector to object distance is < 5 *cm* because there is no bigger movement. For STARK the errors keep increasing for all axes.

Looking at the performance of our method across the scenarios we observe that it has similar performance in all of them. The errors along the z-axis in the occlusion scenario are higher than in the other cases and occur because the human hand holding and occluding the object is sometimes considered part of the object. In such situations, the fingers are aligned with the object and because of the noise in the depth image small angles and discontinuities are not properly detected by the depth segmentation. Moreover, the error along the y-axis of the camera is dominant for our method, especially when the distance is small. This is explained by the partial object occlusion by the gripper, which leads to the object center being estimated higher than it is. We also notice that in the adversarial scenario, the x-axis error is small, which means that the lateral object movement is tracked well by our tracker.

Latency: Lastly, we compare the latency of the three methods, which we define as the total processing time needed for one frame. For THOR the average latency is 0.1036*s*, for Yolov7-f 0.0352*s*, for STARK 0.0333*s*, and for our tracking framework 0.0183*s*.

4.1.2 Ablation study. We carry out ablation studies for the measurement sources (M) and the Kalman Filter (KF) using the same metrics as for the 2D comparison to the state-of-the-art. For the measurement sources, we consider THOR (THOR-f), THOR and optical flow (THOR-fo), and our method (THOR-fod). For the Kalman Filter, we compare a simple filter (simple) and a filter with delay-compensation without the adaptive part (compensated) to our implementation. The results are shown in Table 2. We observe that optical flow and depth segmentation improve the overall performance in all the scenarios, leading to results comparable to the ones reported on visual object tracking benchmarks [14, 24]. We also notice that delay compensation is crucial, especially in adversarial and fast scenarios. Adapting process and measurement noise leads to improvements in all cases, especially for min IoU, demonstrating that it increases robustness and corrects inaccuracies.

Table 2: Ablation study results

	Type	Method	IoU	min IoU	P	*n*CLE
Simple	**M**	THOR-f	63.42	18.13	76.25	0.74
		THOR-fo	65.69	23.98	85.85	0.74
		THOR-fod (ours)	**70.51**	**30.48**	**92.89**	**0.66**
	KF	compensated	64.70	16.79	77.76	0.77
		simple	60.27	15.20	68.33	0.79
Fast	**M**	THOR-f	59.39	10.62	84.02	0.67
		THOR-fo	61.88	17.96	86.70	**0.66**
		THOR-fod (ours)	**66.79**	**25.82**	**86.79**	0.69
	KF	compensated	60.30	14.79	73.80	0.73
		simple	51.04	9.97	52.83	0.79
Occlusion	**M**	THOR-f	48.26	0.00	42.39	0.83
		THOR-fo	52.68	9.12	52.79	0.79
		THOR-fod (ours)	**58.22**	**10.97**	**65.17**	**0.78**
	KF	compensated	54.92	0.0	55.90	0.80
		simple	48.31	0.0	48.30	0.86
Adversarial	**M**	THOR-f	55.33	18.56	57.42	0.76
		THOR-fo	60.55	27.31	67.61	0.72
		THOR-fod (ours)	**66.54**	**31.19**	**73.60**	**0.68**
	KF	compensated	59.61	10.64	69.40	0.76
		simple	48.53	2.75	45.86	0.97

4.2 Integrated system performance

To thoroughly evaluate our system performance during the approach phase of the handovers we carry out two sets of experiments: validation by ourselves (Sec. 4.2.1) and a user study (Sec. 4.2.2). We use success rate and timing metrics such as handover, and reaction time for this analysis. We consider the handover time the time elapsed between the moment the robot starts moving the end-effector (the object is within reach), and when it grasps the object. We define reaction time as the duration between the moment the object can be safely grasped and when the robot reacts by closing the gripper. The time the object can be safely grasped is determined by monitoring the distance between the object point cloud from the raw point cloud of the camera and the end-effector from state

estimation. Carrying out a timing performance evaluation is particularly important, as temporal precision has greater significance than spatial precision in human-robot handover interactions [17].

4.2.1 Validation experiments. For this analysis, we carry out 40 handovers: 20 simple, 4 fast, 4 occlusion, and 12 adversarial scenarios, as defined in Sec. 4.1. The different numbers of experiments in each scenario come from the fact that in the simple scenario, we use four objects, while for the others one. In the adversarial scenario, we have 12 experiments to cover different motions.

Our overall success rate is 89.75 %, which outperforms the state-of-the-art for human-to-robot handovers on fixed manipulators (81.8 % [36]). We consider a handover successful if the robot takes the object from the human. We report one failure in the simple and three in the adversarial scenario. In the simple scenario, joint limits were hit, while in the adversarial scenario, the object went once out of the field of view and the tracker drifted away twice. The initial distance between the human and the robot ranged between 1.6 and 3.1 m and had no meaningful impact on the success rate.

Reaction time analysis: We compute the reaction time of our system for the simple, fast, and occlusion scenarios. We group the simple and fast scenarios for this analysis, as without prediction capabilities, the approaching speed of the object does not affect the reaction time. Adversarial is not considered as the times depend on the moment the human stops moving.

Table 3 shows our average reaction time and the reaction times for the different scenarios compared to the average human reaction times to complex visual stimuli from an independent study [32]. We believe the study to be a fair comparison as both, our robot and the participants have to react on a frame-by-frame basis after processing complex visual stimuli, given as images. Similar to our definition, in [32] the time between displaying the image and the moment a button is pushed is defined as reaction time. We notice that our reaction times for the simple and fast scenarios are close to the ones reported in [32]. The occlusion dataset exhibits higher reaction times due to increased tracking errors.

Table 3: Average reaction times for validation experiments

Scenario	Human	Simple+Fast	Occlusion	Average
$T_{reaction}(s)$	0.46	0.58	0.83	0.68

Handover time analysis: The approach phase for a mobile manipulator can be separated into two sub-phases: walk to object and move end-effector to object when it is within reach. For this analysis, we consider just the second sub-phase, because independent human-human handover studies in a similar setup as our simple scenario report values for that sub-phase [15]. Table 4 compares the handover times for different scenarios, and for every object in the simple scenario to human-human handover times from [15]. It can be observed that we are close to the human-human handover times.

Table 4: Average handover times for validation experiments

Object	Av. Time (s)	Scenario	Av. Time (s)
Toy	2.36	Simple	2.38
Joystick	2.41	Fast	0.87
Bottle	2.26	Occlusion	3.24
Block	2.81	Human [15]	1.76

The average handover times for the objects are similar, except for the wooden block. This is explained by the increased orientation error, which we think comes from the holes in its structure. As expected, the occlusion scenario has the highest average handover time. The fast scenario has the shortest handover time, as the human is trying to accomplish the handover fast and contributes to minimizing this time. The best handover times for human-to-robot handovers are reported in [37] and [9]. Even though our handover times are better, the setup has significant differences as they use a fixed manipulator, and a direct comparison would not be fair.

4.2.2 User study. To get subjective feedback about the interaction quality and the timing performance of our system we carried out a user study with sixteen participants who were not familiar with the robot. Each participant was asked to hand over three different objects: a toy, a cup, and a bottle. We stopped the experiments after one successful handover for each object, leading to three to five handovers per participant. The order of the objects was randomly chosen to avoid any bias. The participants were not instructed on how to hand over the objects, just not to behave adversarially, and to grasp the object on top or on the handle to avoid their fingers being grasped. The last instruction led to small object occlusions, making the setup similar to our simple scenario. The initial distance between the participants and the robot is randomly varied between 1.6 m to 2.5 m. A handover sequence is shown in Fig. 5.

To ensure maximum safety during the user study, manual validation of the tracker's initialization was required. This took up to 5 s and was excluded from the experimental procedure to not influence the participant's perception of the interaction quality and the handover times. Hence, we put the object on a table to initialize the tracker before the actual handover. After initialization, the robot started moving toward the participants, who automatically engaged in the interaction. After the experiments, participants were asked to fill out a questionnaire with Likert scale and open-ended questions.

We carried out 67 handovers during the user study with a success rate of 85.48 %. The success rate per object is 91.3 % for the toy, 94.44 % for the bottle, and 70.83 % for the cup. The participants tended to hand over the cup at a lower position with respect to the gripper probably because they grasped the handle and not the top. This lead often to almost full occlusions of the object by the gripper, impacting negatively the tracking accuracy.

In terms of objective timing metrics, we notice that the average reaction time during the user study is close to the average reaction time of "simple+fast" from Sec. 4.2.1. On the other hand, the handover times decreased compared to the simple scenario, showing that on average the participants were naturally helping the robot to get the object. We notice that the reaction time for the cup is smaller, but the handover time is bigger than for the other objects. The explanation is that it takes longer to get into a graspable position for the cup, as the tolerance between the gripper and cup width is small compared to the other objects. The results of the subjective timing performance analysis are presented in Fig. 6. Over 90 % of the participants agreed or strongly agreed that the system's timing during handover is appropriate and most participants had the same opinion about not having significant idle time during the interaction. These results underline our findings from Sec. 4.2.1 that the reaction and handover times are close to human performance.

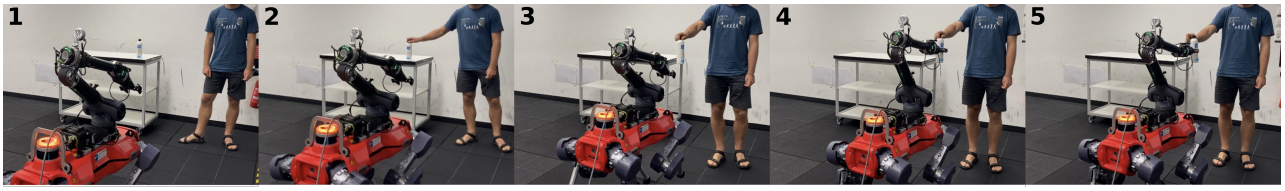

Figure 5: User study setup: (1) starting positions, (2) the robot starts moving and the participant grasps the object, (3) the robot starts moving its end-effector to the object, (4) the robot grasps the object, and (5) the participant releases the object.

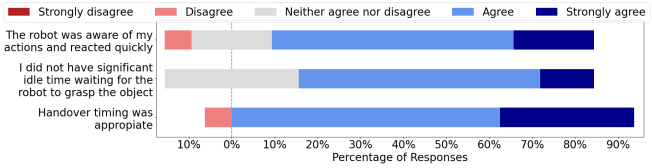

Figure 6: Answers related to the timing performance.

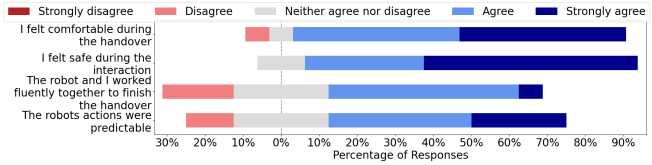

Figure 7: Answers related to the interaction quality.

Moreover, we observed that participants have different preferences on handover location and motion. Additionally, several participants occluded the objects partially. Some participants moved more toward the robot, while others waited for the robot to get closer. Some rotated the objects during the handover, while others kept the orientation constant. Each person moved the different objects with different velocities according to their preference. This is reflected also in the handover time variance, which is half of the average time, as shown in Table 5. The perception of comfort was also slightly different. Some thought the robot came too close for the handover, while others thought the distance was close to what they experienced in human-human handovers. A similar difference was noticeable in the approaching speed with some participants preferring a higher speed, while others considered the speed comfortable. Nevertheless, even with such differences over 90 % of the participants felt safe and comfortable during the handover, as shown in Fig. 7. Furthermore, over half of the participants agreed or strongly agreed that the robot's actions were predictable and that they worked together fluently to finish the handover.

Table 5: Handover and reaction times from the user study

	Toy	**Bottle**	**Cup**	**Average**
$T_{handover}$	1.96 ±1.01s	1.91 ±1.03s	2.03 ±0.94s	1.97 ±0.99s
$T_{reaction}$	0.74 ±0.25s	0.67 ±0.09s	0.52 ±0.08s	0.62 ±0.12s

All participants thought the system could be improved in a few ways mainly integrating intention recognition, such that the participants decide when the handover starts and not the robot as in the current state, and using a different gripper: a soft and preferably anthropomorphic hand. We also noticed that the position of the camera is unfavorable, leading to significant object occlusions when the object is not handed over from the top, e.g. the case of the cup.

4.2.3 Limitations. One disadvantage of the proposed method is the increased number of tuning parameters, which depend highly on the camera being used. Additionally, the object has to be in the field of view at all times. The rigid attachment of the camera to the arm in addition to its mounting position limited the feasible grasp locations which guarantee that the object remains in the field of view or does not get fully occluded by the arm and gripper. This can be alleviated by using a pan-tilt head for the camera and redesigning the mounting position. Even though the tracker is object-agnostic, some categories, such as small, thin, or non-convex objects can not be handled properly. The latter is a limitation of depth segmentation and can be relaxed by using a different algorithm.

To avoid grasping their fingers, the participants in the user study were instructed to hold the objects from the top, while the robot was constrained to grasp the lower half. However, because of the occlusions caused by the gripper, the estimated object height might be smaller than the actual one and therefore, the robot fails to grasp the object in its lower half. In our experiments, it was enough to avoid the fingers, but a more reliable and sophisticated avoidance module is necessary.

Finally, as the user study shows, every person has different handover preferences, e.g. in terms of approaching speed and stopping distance of the robot. The perception algorithm adjusts to these preferences, tracking the object reliably and generating feasible grasps. The simple control strategy, however, lacks adaptiveness to such preferences, as the robot approaches everyone with the same velocity and stops at a specific distance without online adaptation.

5 CONCLUSIONS AND FUTURE WORK

In this work, we introduce an efficient object-agnostic perception framework designed for human-to-robot handovers with legged manipulators. It handles partial object occlusions and viewpoint changes and runs at speeds comparable to which visual stimuli reach the human brain [16]. We compare the proposed tracking algorithm to the state-of-the-art on our open-sourced handover dataset and show that we outperform them in 2D and 3D for all four handover scenarios. Furthermore, we analyze the performance of the integrated legged robot system in terms of handover success rate and timing-related metrics. We reach close to human timing performance for the approach phase of the handovers not only in terms of handover and reaction time but also by considering subjective metrics gathered from the user study.

Future work focuses on tackling the most important limitations, such as human hand collision avoidance and integrating a pan-tilt unit for the camera. Furthermore, we will focus on adaptive control strategies to adjust to human preferences online during handovers.

ACKNOWLEDGMENTS

This research was supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Digital Fabrication (NCCR dfab) and as part of project No.188596.

REFERENCES

[1] [n. d.]. Jetson Xavier AGX. https://developer.nvidia.com/embedded/jetson-agx-xavier. Accessed: 2023-06-26.

[2] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. [n. d.]. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In *2017 IEEE Power & Energy Society General Meeting*. 1–5.

[3] Jacopo Aleotti, Vincenzo Micelli, and Stefano Caselli. [n. d.]. Comfortable robot to human object hand-over. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. 771–776.

[4] Oleksandr Bailo, François Rameau, Kyungdon Joo, Jinsun Park, Oleksandr Bogdan, and Inso Kweon. 2018. Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. *Pattern Recognition Letters* (2018).

[5] Heike Benninghoff, Florian Rems, and Toralf Boge. 2014. Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. *Acta Astronautica* 102 (2014), 67–80.

[6] J.-Y. Bouguet. 1999. Pyramidal implementation of the lucas kanade feature tracker.

[7] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[8] Po-Kai Chang, Jui-Te Huang, Yu-Yen Huang, and Hsueh-Cheng Wang. [n. d.]. Learning End-to-End 6DoF Grasp Choice of Human-to-Robot Handover using Affordance Prediction and Deep Reinforcement Learning. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*.

[9] Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. 2023. Learning Human-to-Robot Handovers from Point Clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9654–9664.

[10] Marco Costanzo, Giuseppe De Maria, and Ciro Natale. 2021. Handover Control for Human-Robot and Robot-Robot Collaboration. *Frontiers in Robotics and AI* 8 (2021), 132.

[11] Haonan Duan, Peng Wang, Yiming Li, Daheng Li, and Wei Wei. 2022. Learning Human-to-Robot Dexterous Handovers for Anthropomorphic Hand. *IEEE Transactions on Cognitive and Developmental Systems* (2022), 1–1. https://doi.org/10.1109/TCDS.2022.3203025

[12] F. Furrer, T. Novkovic, M. Fehr, A. Gawel, M. Grinvald, T. Sattler, R. Siegwart, and J. Nieto. [n. d.]. Incremental Object Database: Building 3D Models from Multiple Partial Observations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 6835–6842.

[13] Q. Gan and C.J. Harris. 2001. Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. *IEEE Trans. Aerospace Electron. Systems* 37, 1 (2001), 273–279. https://doi.org/10.1109/7.913685

[14] Lianghua Huang, Xin Zhao, and Kaiqi Huang. 2021. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 5 (2021), 1562–1577. https://doi.org/10.1109/TPAMI.2019.2957464

[15] Markus Huber, Markus Rickert, Alois Knoll, Thomas Brandt, and Stefan Glasauer. [n. d.]. Human-robot interaction in handing-over tasks. In *RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication*. 107–112.

[16] Bryan J Kemp. 1973. Reaction time of young and elderly subjects in relation to perceptual deprivation and signal-on versus signal-off conditions. *Developmental Psychology* 8, 2 (1973), 268.

[17] Ansgar Koene, Anthony Remazeilles, Miguel Prada, Ainara Garzo, Mildred Puerto, Satoshi Endo, and Alan M Wing. 2014. Relative importance of spatial and temporal precision for user satisfaction in human-robot object handover interactions. In *Third International Symposium on New Frontiers in Human-Robot Interaction*.

[18] Jelizaveta Konstantinova, Senka Krivic, Agostino Stilli, Justus Piater, and Kaspar Althoefer. [n. d.]. Autonomous object handover using wrist tactile information. In *Towards Autonomous Robotic Systems: 18th Annual Conference, TAROS 2017*. Springer, 450–463.

[19] T.D. Larsen, N.A. Andersen, O. Ravn, and N.K. Poulsen. 1998. Incorporation of time delayed measurements in a discrete-time Kalman filter. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Vol. 4. 3972–3977 vol.4.

[20] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. 2018. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. *CoRR* abs/1812.11703 (2018).

[21] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, and Yasemin Bekiroglu. 2019. Dynamic grasp and trajectory planning for moving objects. *Autonomous Robots* 43, 5 (2019), 1241–1256.

[22] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. 2022. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics* 7, 62 (2022), eabk2822. https://doi.org/10.1126/scirobotics.abk2822

[23] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. [n. d.]. Automatic grasp planning using shape primitives. In *2003 IEEE International Conference on Robotics and Automation*, Vol. 2. 1824–1829.

[24] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. 2018. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. *CoRR* abs/1803.10794 (2018). arXiv:1803.10794

[25] Valerio Ortenzi, Akansel Cosgun, Tommaso Pardi, Wesley P Chan, Elizabeth Croft, and Dana Kulić. 2021. Object handovers: a review for robotics. *IEEE Transactions on Robotics* (2021).

[26] Matthew K.X.J. Pan, Espen Knoop, Moritz Bächer, and Günter Niemeyer. [n. d.]. Fast Handovers with a Robot Character: Small Sensorimotor Delays Improve Perceived Qualities. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 6735–6741.

[27] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martinez, and Joaquín Fernández-Valdivia. [n. d.]. Diatom autofocusing in brightfield microscopy: a comparative study. In *2000 IEEE International Conference on Pattern Recognition. ICPR*, Vol. 3. 314–317.

[28] Patrick Rosenberger, Akansel Cosgun, Rhys Newbury, Jun Kwan, Valerio Ortenzi, Peter Corke, and Manfred Grafinger. 2020. Object-Independent Human-to-Robot Handovers using Real Time Robotic Vision. *CoRR* abs/2006.01797 (2020).

[29] Ricardo Sanchez-Matilla, Konstantinos Chatzilygeroudis, Apostolos Modas, Nuno Ferreira Duarte, Alessio Xompero, Pascal Frossard, Aude Billard, and Andrea Cavallaro. 2020. Benchmark for human-to-robot handovers of unseen containers with unknown filling. *IEEE Robotics and Automation Letters* 5 (2020), 1642–1649.

[30] Axel Sauer, Elie Aljalbout, and Sami Haddadin. 2019. Tracking Holistic Object Representations. In *British Machine Vision Conference (BMVC)*.

[31] Jean-Pierre Sleiman, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter. 2021. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters* 6, 3 (2021), 4688–4695.

[32] Simon Thorpe, Denis Fize, and Catherine Marlot. 1996. Speed of processing in the human visual system. *nature* 381, 6582 (1996), 520–522.

[33] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* (2022).

[34] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. 2021. Learning Spatio-Temporal Transformer for Visual Tracking. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 10428–10437.

[35] Wei Yang, Chris Paxton, Maya Cakmak, and Dieter Fox. [n. d.]. Human grasp classification for reactive human-to-robot handovers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 11123–11130.

[36] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. 2020. Reactive Human-to-Robot Handovers of Arbitrary Objects. *CoRR* abs/2011.08961 (2020).

[37] Wei Yang, Balakumar Sundaralingam, Chris Paxton, Iretiayo Akinola, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. [n. d.]. Model Predictive Control for Fluid Human-to-Robot Handovers. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*. 6956–6962.

[38] Luka Čehovin Zajc, Ales Leonardis, and Matej Kristan. 2015. Visual Object Tracking Performance Measures Revisited. *IEEE Transactions on Image Processing* 25 (02 2015). https://doi.org/10.1109/TIP.2016.2520370