Towards safe human-to-robot handovers of unknown containers

Yik Lung Pang, Alessio Xompero, Changjae Oh, Andrea Cavallaro

Abstract—Safe human-to-robot handovers of unknown objects require accurate estimation of hand poses and object properties, such as shape, trajectory, and weight. Accurately estimating these properties requires the use of scanned 3D object models or expensive equipment, such as motion capture systems and markers, or both. However, testing handover algorithms with robots may be dangerous for the human and, when the object is an open container with liquids, for the robot. In this paper, we propose a real-to-simulation framework to develop safe human-to-robot handovers with estimations of the physical properties of unknown cups or drinking glasses and estimations of the human hands from videos of a human manipulating the container. We complete the handover in simulation, and we estimate a region that is not occluded by the hand of the human holding the container. We also quantify the safeness of the human and object in simulation. We validate the framework using public recordings of containers manipulated before a handover and show the safeness of the handover when using noisy estimates from a range of perceptual algorithms.

I. INTRODUCTION

Human-to-robot handovers are important for daily household activities. These interactions involve a human, a robot, and an object, and can be split into three phases, namely human manoeuvring, handover, and robot manoeuvring [1], [2]. For example, a human can give to the robot household containers, such as drinking glasses, cups, or food boxes, whose properties vary when filled with content. Therefore, an additional phase where the human prepares the object for the handover (object manipulation), e.g. pouring content into a container, can be included prior to human manoeuvring. During human manoeuvring, the human holds the object and approaches the robot. The robot should understand the intention of the human to pass the object and approach the human hand. The robot should reach for a safe region on the object without harming the human, and close the gripper with sufficient forces to hold the object before the human releases their hold. During the handover, both the human and the robot are in contact with the object. During robot manoeuvring, the robot delivers the object to a target location without damaging the container or spilling the content.

Accurately estimating the locations of the human hands and the properties of the object is critical for the robot to safely execute handovers. Achieving this estimation with a real setup is challenging because no information about the object is available a priori to the robot, or additional and expensive equipment would be needed [2]. Referring specifically to household container-like objects, recent works classify the type and level of the content in a container

or estimate the capacity and mass of the container (empty or filled), using audio or visual data [3]–[7]. Reasoning about the human dynamics from visual data can also provide information about the physical properties of a container [8].

Existing methods for human-to-robot handovers focus on the control algorithm of the robot to reach an object (e.g., water bottle) at a predicted location [1]. The safety of the human is partially addressed by avoiding to select grasp points near the human hand in an image [9] or by classifying the human grasp type [10]. Moreover, most of the existing works rely on accurate predictions from expensive motion capture systems [1], [11], or objects that are easily recognisable in the scene because of their shape or colour [12], or equipped with markers [13]. Simulation environments provide a safe alternative to test robot algorithms for handovers. However, existing works focus on the planning of the handover trajectory and location [14]–[16], disregard object shape and safety [17], and are limited to static setups where the robot waits for the object to be placed in the gripper [18]–[20].

In this paper, we propose a framework that combines a perceptual algorithm using recordings of real scenes with a simulator to complete and achieve safe human-to-robot handovers. The framework addresses potential safety issues for the human and object before deploying the robot to a real environment, while requiring minimal hardware setup. Our perceptual algorithm uses videos as input and estimates the poses of the human hands and the properties of a container, such as shape, weight, and location. Using these estimations, we propose to quantify the safeness of the handover as the probability for the robot gripper to touch the human hand (human safety) and the probability to drop, break or squeeze the container (object safety). Moreover, we estimate a safe grasp region that accounts for the available and unoccluded region on the held container to decide where to grasp without harming the human hand. We validate our framework on selected video recordings from the CORSMAL Containers Manipulation (CCM) dataset, where humans interact with different containers [21]. Experiments show the safeness of the handovers when the physical properties are estimated with the vision algorithm or with different perceptual and multi-modal algorithms[1].

II. RELATED WORK

In this section, we review existing works using simulations for developing handover algorithms, addressing the safeness of the handover, and estimating the mass of containers.

Yik Lung Pang, Alessio Xompero, Changjae Oh, and Andrea Cavallaro are with Centre for Intelligent Sensing, Queen Mary University of London, UK {y.l.pang, a.xompero, c.oh, a.cavallaro}@qmul.ac.uk.

[1]Code, data, and videos of the experiments are available at: http://corsmal.eecs.qmul.ac.uk/safe_handover.html

Simulation platforms are often used by existing approaches for developing path planning algorithms focusing on the trajectory of the human hand and robot gripper, while providing a safe virtual environment before the deployment to a real setup [14]–[16]. However, the object and its properties are neglected [16], precise 3D object models are required [18]–[20], and grasping forces are not considered in the simulation [14], [15]. Moreover, the handover is simulated in a static [19] or a limited dynamic [18], [20] setup. In the former, the robot stays stationary while waiting for the object to be placed in the gripper. In the latter, the robot reaches a handover location, waits for the human to apply a force on the object, and then releases the gripper. Alternatively, virtual and mixed reality combine user input from the real world and allow the users to interact with a virtual robot, but requiring additional hardware [22]–[24]. A real-sim-real framework produces a task-relevant simulation environment based on visual estimations from RGB video, using simulated objects with primitive geometric shapes (spheres, cubes) [17]. However, these approaches do not consider the human or object safety during a human-to-robot handover, which is crucial in a real-world scenario.

Safeness of the handover is important for the human and also for the object when it is fragile or deformable. Existing works have addressed the human safety by quantifying the risk of harming the human [25], [26] or by designing strategies for the robot control to prevent collision between the human and robot [9], [10], [27]–[29]. A danger index quantifies the safeness of the handover as the distance between the human and the robot, as well as the operating velocity and inertia of the robot [25]. An exponentially decaying function is then modelled to reduce the danger index when the distance between the human and the robot is greater than a minimum threshold. This leads to a sharp decrease in the danger index when the robot is still close to the human. The operating speed and force of the robot can be limited to reduce the risk of injuring humans in accidental collisions [26], [28], as also outlined in ISO/TS 15066:2016 [30], or a safety barrier function can be defined around the robot [29]. A neural network is trained to classify the human hand pose into a discrete set of grasp types and then executes a corresponding predefined canonical robot grasp that is safe for the human [10]. Potentially dangerous robot grasps that are too close to the human can be removed by segmenting the body and hand as well as applying a threshold on the pixel distance between the human and the grasp point [9]. While the safety of the object has not yet been considered during the handover, there are existing works that assess the deformability of some objects. Deformability of the object can be minimised by defining an upper bound on the applied normal force using tactile data for each specific object [31]. A grasp metric that minimises the work done by the gripper jaws can be defined for deformable objects, but calibration against physical objects is still needed [32].

Object mass estimation requires the reasoning on different physical properties, especially when the object is a container. Existing perception algorithms use uni-modal or multi-modal data, such as audio, images, and videos, to classify the content type and level as well as the container capacity [4]–[7]. Convolutional neural networks can be trained to classify the content level within a range of containers from a single image when hand occlusions are present [7]. While the performance is limited by the uni-modal input, the choice of training strategy, e.g. combining adversarial training and transfer learning, can improve the classification accuracy [7]. Independent classification of content type and level can be achieved by using convolutional and recurrent neural networks with only audio as input data [4] or through late fusion of the predictions from both audio and visual features [5]. Alternatively, multiple multi-layer perceptrons can be trained with audio data and conditioned on the container category estimated from a majority voting of the object detection across the frames of multi-view sequences [6]. Container capacity can be estimated as an approximation of a reconstructed shape [4], [5], [33]. An iterative approach minimises a 3D primitive to the real object shape by constraining to the object segmentation mask from two views of a wide-baseline stereo camera, using both RGB, depth, and infrared images [5]. The shape is then approximated to a cylinder to compute the capacity [5]. The shape of a container can be instead reconstructed as a 3D point cloud by detecting the object in the RGB frame where the container is the most visible and using the corresponding pixel values in the depth map [4]. Then, the shape can approximated to a cuboid and the capacity is computed by considering only the voxels that project within the object boundaries in the RGB image [4]. Alternatively, the prior knowledge of a detected object category can guide the sampling from a shape distribution using Gaussian process [6]. Despite achieving high accuracy in individual properties [4], [5], the estimation of the object mass is still challenging and a robot deployed in a real environment may apply forces that can be unsafe for the human or the object.

III. PROBLEM DEFINITION

Let $\mathcal{H}_L = \{\mathbf{h}_{L,i}\}_{i=1}^K$ and $\mathcal{H}_R = \{\mathbf{h}_{R,i}\}_{i=1}^K$, where $\mathbf{h}_{L,i}, \mathbf{h}_{R,i} \in \mathbb{R}^3$, be the set of K keypoints in 3D representing the left and right hand, respectively, of a human manipulating a container. The container is characterised by its dimensions given as maximum width w, height h, and depth d; its volume V; and its mass when empty m_C. During manipulation, the container is also represented by its time-varying location $\mathbf{t}_k \in \mathbb{R}^3$ (i.e., the centroid in 3D) and orientation $\theta_k \in \mathbb{R}^3$ at time k. The container can be either *empty* or filled with a content $\tau \in \{pasta, rice, water\}$ at different levels $\lambda \in \{half\text{-}full, full\}$, expressed as a percentage of the container capacity (i.e., 50%, 90%). This results in seven feasible combinations of content type and level. Moreover, each content type has an associated density $\rho(\tau)$. The final object mass is defined as $m = m_C + m_f$, where m_f is the filling mass.

After manipulating the container, the human approaches the robot while holding the container to hand it over. The robot should approach the human and receive the container,
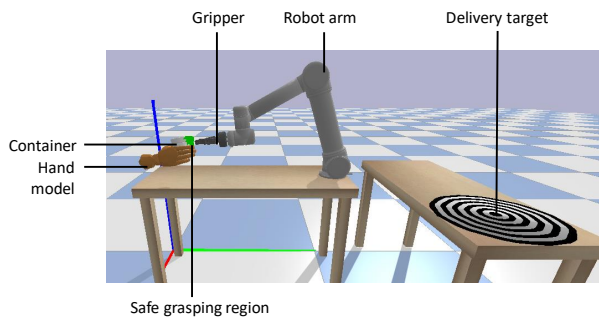
Fig. 1. The handover environment in PyBullet [34] and its coordinate system. The estimated safe grasp region on the container, highlighted in green, is not occluded by the hand and thus safe for the robot to receive the container without harming the human hand.

Fig. 2. Our manually defined — *directional vector* and — *up vector* (left). The — *right vector* is obtained as the vector orthogonal to both the directional and up vector. The reconstructed hand model (right).

while avoiding harming the human and stably holding the container. Once the handover is completed, the robot places the container at a pre-defined target area centred at $\mathbf{d} \in \mathbb{R}^3$ with radius η on a table.

The goal is to i) estimate the physical properties of a container, i.e. width ($\tilde{w}$) and mass ($\tilde{m}$), ii) estimate the location of the container ($\tilde{\mathbf{t}}_k$) and human hands ($\tilde{\mathcal{H}}_{L,k}, \tilde{\mathcal{H}}_{R,k}$) over time, iii) perform a safe handover, and iv) quantify the safeness of the human and container.

IV. SAFE HUMAN-TO-ROBOT HANDOVER

To achieve a safe human-to-robot handover, we present a simulated handover framework recreated from the real scene based on the physical properties estimated by a perception algorithm[2]. We introduce an algorithm that identifies a target region on the container for the robot to grasp, while avoiding touching the human hand. We also quantify the safeness of the human and container during the handover within the simulation environment.

A. Handover setup

We recreate in PyBullet [34] the handover setup that consists of the 6 DoF robotic arm (UR5) equipped with a 2-finger parallel gripper (Robotiq 2F-85); a table where object manipulation and handover is happening, as well as where the robot is placed; an additional table behind the robot for delivering the object in a targeted area; a 3D container model; and left and right human hands based on the Modular Prosthetic Limb model [35]. Fig. 1 illustrates the handover setup in the simulator. The simulator renders the motion of the object and hands based on the predictions provided by a perception algorithm during object manipulation and human manoeuvring phases. The simulator also renders the motion of the robot arm based on a control algorithm, and then provides the applied forces while receiving the container in the moment of the simulated handover.

For the container, the simulator renders a 3D object model as reconstructed by the perception algorithm, assuming that scanned 3D models of the objects are not available. To better replicate the real physical property, we use a high-precision

[2]Note that we design the framework to be modular and efficient and hence different hardware and algorithms can replace the choices of our setup.

electronic scale to annotate the object weight, with or without the content. We then simulate the container movement by defining a physical constraint that applies a fixed force to move the container to the estimated position and orientation. Similarly for the hand motions, the simulator adjusts the hand and finger poses over time using a physical constraint based on the hand predictions, $\mathcal{H}_L$ and $\mathcal{H}_R$, and their estimated joint angles. The container movement is simulated until the robot reaches the container and closes the gripper to grasp the container. If the robot is not able to reach the container before the recording ends, we keep the last estimated location of both the container and the human hands for 2 s.

B. Perception

We estimate the pose of the hands over time, the location of a container over time, and the physical properties of the container, such as dimensions and mass, using only video inputs. Videos are acquired by fixed, calibrated stereo cameras, that capture object manipulation and human manoeuvring from the real scene.

To obtain the pose of the human hands, we first use OpenPose [36] to detect the $K = 21$ keypoints from each hand and for each camera view independently. For each keypoint, we triangulate its position in 3D [37] and apply a Kalman filter [38] to reduce the impact of inaccurate estimations. We also approximate the hand model to a rigid object by defining directional vectors based on the estimated keypoints (see Fig. 2).

To obtain the trajectory of the container, we build upon an existing vision algorithm [2] that localises and tracks a container from two camera views. We first localise the object of interest and initialise a 2D tracker in each view using Mask R-CNN [39]. Note that Mask R-CNN is pre-trained on COCO [40] considering only two classes, *wine glass* and *cup*. We then track the selected initial mask with SiamMask [41] in each view independently. For each frame k, we recover the 3D location ($\tilde{\mathbf{t}}_k$) of the container based on the triangulation method with the centroid of the two object masks (LoDE) [33]. We then apply a Kalman filter [38] to $\tilde{\mathbf{t}}_k$ to reduce the impact of inaccurate estimations by LoDE, especially in the presence of occlusions. We assume that the hand and held container can be approximated to a single rigid object. Therefore, the orientation of the held container θ_k is set to the orientation of the human hand.

From the estimated 3D location, LoDE reconstructs a 3D shape of the container as a sparse point cloud, $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^J, \mathbf{q}_j \in \mathbb{R}^3$, where J is the total number of points. Assuming that containers have a rotationally symmetrical shape, LoDE approximates the shape to a set of 3D points lying on circumferences at different heights. The

radius of each circumference is obtained by minimising an initial hypothetical radius until all the 3D points of the circumference lie within the object mask of both views. From the estimated shape, the container dimensions $(\tilde{w}, \tilde{h}, \tilde{d})$ and volume $\tilde{V}$ can be obtained as a by-product, i.e. the maximum width is given by the largest diameter among the circumferences, and the volume is approximated to the Riemman sum of the partial volumes (slicing method).

To render in simulation a container as close as possible to the reality, we also perform offline shape estimation of the container with LoDE by manually selecting the frame where the container is occlusion-free. We then process the sparse point cloud by defining faces of neighbouring points and forming a triangular 3D mesh. For some containers with concave shapes (e.g., wine and cocktail glasses), we use Volumetric Hierarchical Approximate Convex Decomposition [42] to approximate the concave shape to multiple convex components.

To estimate the object mass, we first determine the filling mass $\tilde{m}_{f,k}$ for the k-th frame as the product of estimated content level $\tilde{\lambda}_k$, volume $\tilde{V}$, and selected content density $\rho(\tilde{\tau}_k)$ based on the estimated type $\tilde{\tau}_k$:

$$\tilde{m}_{f,k} = \tilde{\lambda}_k \tilde{V} \rho(\tilde{\tau}_k). \tag{1}$$

Similar to [7], we devise a convolutional neural network-based classifier that predicts the content type and level for each frame and each view. In addition to the feasible combinations as classes, we also include *unknown* as an extra class to handle opaque or translucent containers for which the content type and level cannot be estimated. Note that, prior to the hard decision on the predicted class, we multiply the predicted probabilities of all classes between the two views at the current frame k and the predicted status at frame k from the status at the previous frame $k-1$. For the latter, we design a fixed transition matrix that avoids the classification to change between unrelated statuses. Because of the object manipulation phase, we initialise equal probabilities for *empty* and *unknown* at $k = 0$. Note that we only consider the estimation of the filling status at the handover moment.

We complement the estimations that cannot be obtained by our vision algorithm using the annotated values, such as the filling densities, $\rho(\tilde{\tau}_k)$, and container mass, m_C. When computing the filling mass, we use pre-defined filling densities as the average of the computed densities from the annotated masses, volumes, and filling levels: 0.41 g/mL for pasta, 0.85 g/mL for rice, and 1 g/mL for water. We then compute the estimated object mass as $\tilde{m}_k = m_C + \tilde{m}_{f,k}$.

C. Robot control with safe grasp region

During the handover, the robot approaches the human to receive the object using the predicted time-varying object location $\tilde{\mathbf{t}}_k$ as input to the robot controller [2]. However, $\tilde{\mathbf{t}}_k$ can be inaccurate and occluded by the human hand. We thus seek the region on the container surface that is not affected by the hand occlusion to execute the robot grasp. If a safe region is not available (see Fig. 3, bottom row), the control
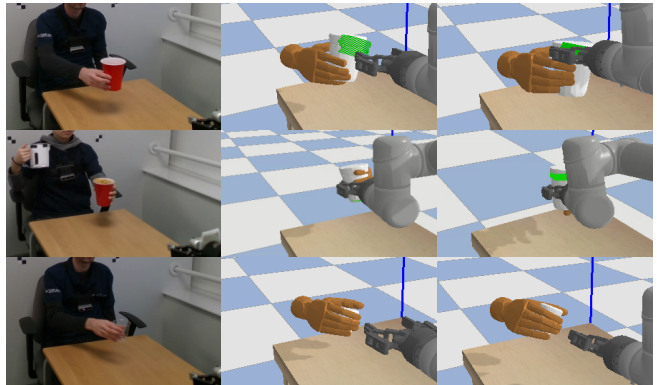

Fig. 3. Samples of safe regions and grasping. Top row: The estimated safe grasp region with successful grasping. Middle row: Target switch between disjoint safe regions, causing a collision with the hand as there is not enough time for the robot to react. Bottom row: The robot avoids approaching the container as there is no safe grasp region.

algorithm will keep the robot at a fixed distance from the container to guarantee the safety of the human.

To determine the safe region, we find a subset of the estimated object shape that is not occluded by the human hands and is in front of the robot. Moreover, we take into consideration a margin r that accounts for both the width of the gripper w_g and an enlarged thickness of the human finger w_h, i.e. $r = (w_g + w_h)/2$. To this end, we first select a set of vertical components in $\mathcal{Q}$ that are within the margin of the gripper to grasp the object:

$$\mathcal{Z} = \{z : \min \mathcal{Q}_z + \frac{w_g}{2} \leq z \leq \max \mathcal{Q}_z - \frac{w_g}{2}\}. \tag{2}$$

We then determine the set of unsafe vertical ranges based on the estimated keypoints of both human hands, $\mathcal{H} \leftarrow \mathcal{H}_L \cup \mathcal{H}_R$, by selecting all the keypoints within the space occupied by the container:

$$\bar{\mathcal{Z}} = \{h_z : (h_y > t_y - \frac{\tilde{w}}{2}) \wedge (|h_x - t_x| < \tilde{w})\}, \tag{3}$$

and we thus compute the safe range as

$$\tilde{\mathcal{Z}} = \{z \in \mathcal{Z} : (z \leq \min \bar{\mathcal{Z}} - r) \vee (z \geq \max \bar{\mathcal{Z}} + r)\}. \tag{4}$$

The safe region is therefore given by all the 3D shape points in the safe range and in front of the robot, $\{\mathbf{q}_s \in \mathcal{Q} : q_{s,z} \in \tilde{\mathcal{Z}} \wedge q_{s,y} > t_y\}$. Note that there might be cases where the hand occludes the middle of the container resulting in two disjoint safe regions. In this situation, we select the largest subset of safe range because a small region can lead to higher chance of harming the human, e.g. due to inaccuracies of the controller.

We integrate the safe region estimation into the control algorithm [2] that continuously estimates and adjusts the joint angles of the robot arm to reach the container based on the prediction from the perceptual algorithm. Specifically, the control algorithm uses the Quadratic Programming solver with inverse kinematics formulations to reach the container at the predicted target location [43]. To consider the estimated safe region, we modify the target location from the estimated 3D object centroid $\tilde{t}_k$ to the location $\mathbf{g} = [t_x, t_y, g_z]$ within the centre of the container and in the middle of the safe

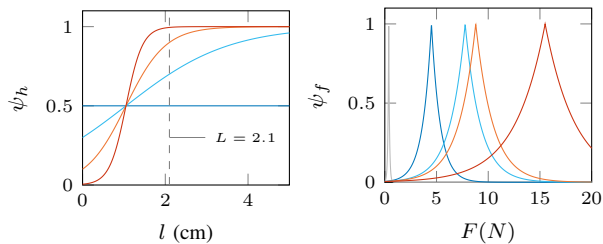Fig. 4. Human and object safety functions. Human safety, ψ_h, as a function of the distance between the closest points of the robot and the human subject's hand, l, when varying c: ▬ $c = 0.5$; ▬ $c = 0.7$; ▬ $c = 0.9$; ▬ $c = 0.995$. Object safety, ψ_f, as a function of the total normal force applied on the object, $\tilde{F}$, when $c = 0.995$ for plastic red cup (C1) with different content: ▬ *Empty*; ▬ 50% *pasta*; ▬ 90% *pasta*; ▬ 50% *rice*; ▬ 90% *pasta*.
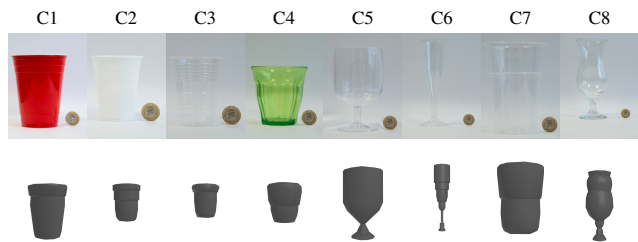

Fig. 5. Containers from the CCM dataset (top row) and their corresponding 3D mesh reconstruction (bottom row) used in our framework. Note the different container sizes compared to the £1 coin used as reference size.

region, i.e. $g_z = (\max \tilde{\mathcal{Z}} + \min \tilde{\mathcal{Z}})/2$. Once the robot is within 1 cm from the target location, the robot closes the gripper with the estimated width from $\mathcal{Q}$ at g_z. The control algorithm will also target the orientation of the hand model as estimated by the vision algorithm to grasp the container directly from the front to avoid spilling of the content due to tilting of the gripper.

To hold and deliver the container, the gripper must apply enough force to balance the weight of the object and the force introduced by the acceleration of the robot arm a_{max} when holding and delivering the object. Therefore, we approximate the applied force to

$$\tilde{F} \approx \frac{\tilde{m}(g + a_{max})}{\mu}, \quad (5)$$

where $\tilde{m}$ is the predicted mass of the container (and content, if any); $g = 9.81$ m/s^{-2} is the gravitational acceleration; and μ is the coefficient of friction between the container and the gripper. To apply the required normal force $\tilde{F}$ given the predicted mass $\tilde{m}$, the robot needs to issue a joint effort $\tilde{\varepsilon}$. We design a linear model between an issued joint effort ε and the normal force F as applied by the robot in the simulator,

$$F = a\varepsilon + b. \quad (6)$$

We fit this model with a set of trials where a synthetic cylindrical object (length 10 cm, radius 3 cm, weight 0.01 kg) placed in mid-air above a table was grasped and held by the robot arm to measure the applied total normal force, while varying the issued joint effort ε ([0.01, 1.25] N). The robot control issues $\tilde{\varepsilon}$ based on the predicted force $\tilde{F}$ and using the inverse model of Eq. 6, where $\tilde{m}$ is estimated at the frame when the grasp is executed.

D. Safeness

We assess the safeness of a handover within the simulation environment by separately quantifying two measures, human safety and object safety.

For *human safety*, we quantify the probability of grasping a container while not touching the human hand. Specifically, we define a minimum safety distance L around the human hand and a modified sigmoid function of l, *i.e.* the distance

between the closest points of the robot and the human hand,

$$\psi_h = \frac{1}{1 + e^{(\frac{2l}{L} - 1)\ln\frac{1-c}{c}}}, \quad (7)$$

where c controls the sensitivity of ψ_h (Fig. 4). We set the value of c such that ψ_h is converging at its maximum when the robot is further than the safety distance, i.e. $l \geq L$, and decays when too close to the human fingers, i.e. $l < L$.

For *object safety*, we quantify the probability that the predicted force applied by the robot will be able to hold the container so that it does not drop, break or deform. Specifically, we define an exponential function that accounts for the difference between the predicted normal force $\tilde{F}$ and the required normal force at the real object mass as available to the simulator, $\hat{F}$:

$$\psi_f = e^{\frac{\lfloor \tilde{F} - \hat{F} \rfloor}{\hat{F}} \ln(1-c)}, \quad (8)$$

where c controls the sensitivity of ψ_f (Fig. 4). A negative difference represents an increase in the probability of dropping the container and a positive difference represents an increase in the probability of breaking or deforming the container. Note that the function penalises the safeness for rigid objects, as larger forces would be required to reach the breaking point compared to a non-rigid object that could start deforming with an applied force that is larger than the optimal one used to just hold the object [31], [32].

V. VALIDATION

A. Experimental setup

We evaluate the proposed framework on the CORSMAL Containers Manipulation dataset [21]. We consider RGB videos, acquired at 30 Hz with 1280×720 pixels resolution, from the two fully calibrated (intrinsic and extrinsic parameters) camera views fixed to the side of the robot arm. The videos contain a human sitting in front of the robot and pouring a content into a container placed on the table (scenario 1) or into a container while held by the human (scenario 2) before handing the container towards the robot. We select in total 112 recordings of eight cups and drinking glasses (see Fig. 5, top row) captured under the same background and lighting conditions.

The recordings in the dataset contain only the object manipulation and human manoeuvring phases. The simulation environment of our framework complements the real recordings to complete the handover and robot manoeuvring
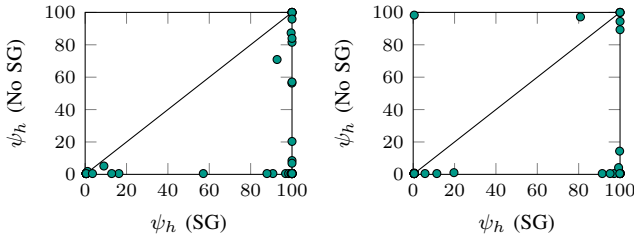
Fig. 6. Comparison of human safety (ψ_h) between our safe grasping region estimation (SG) and the baseline (No SG) [2] on scenario 1 (left) and scenario 2 (right) from the CCM dataset. The more the points are near the bottom right, the more effective SG is.

Fig. 7. Safety measures for human (ψ_h) and object (ψ_f, Δ) across the 112 CCM recordings. Note that failures in detecting or estimating the hand pose are shown with a red cross. KEY – 0: empty; FX: filling type (F) and level (X), where F represents pasta (P), rice (R), or water (W); and X is half-full (5) or full (9). CY: container (C) id (Y) in the same order as Fig. 5.

phases, and assesses the safeness of the handovers. For further assessment of the handover safeness, we compared the physical properties estimated by our vision algorithm against those estimated by three alternative perceptual pipelines, namely BIT [5], HVRL [4], and VA2M [6] (see Sec. II for details). Note that these alternative algorithms use both audio and visual data as input for the different tasks. Moreover, we use the estimations from our vision algorithm for the properties not addressed by these pipelines.

B. Parameters setting

We approximate the width of the human finger to $w_h = 2.0$ cm [44] and set the width of the gripper finger to $w_g = 2.2$ cm ($r = 2.1$ cm). We set $\mu = 1.0$ for both the hand and the container. We set $L = r$ to discourage the gripper from colliding with the hand when reaching the container [25]. We set $c = 0.995$ to ensure that the safety measures decay quickly as the probability of harm for the human and object increases. For Δ, we set $\eta = 500$ mm [2]. We set $\phi = \frac{\pi}{4}$ as any container placed at this angle would fall after a few seconds before evaluating Δ. For estimating the force $\tilde{F}$, we set $a_{max} = 27.9$ ms^{-2} (mean maximum acceleration). For the maximum joint effort, we obtain $a = 25.35$ and $b = 0.045$ N by fitting the model via linear regression.

C. Performance measure

We quantify the accuracy of delivering the container upright and within the target area as

$$\Delta = \begin{cases} 1 - \frac{\alpha}{\eta}, & \text{if } (\alpha < \eta) \text{ and } (\beta < \phi), \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where α is the distance from the centre of the container base to the target location $\mathbf{d}$; η is the maximum distance allowed from the pre-defined delivery location; β is the angle between the vertical axis of the container and the vertical axis of the world coordinate system; and ϕ is the value of β at which the container would tip over.

D. Results and discussion

In this section, we discuss the impact of selecting a safe grasp region on the safeness of the handover and compare different perception algorithms. The safety and performance measures were defined in the range [0,1] and will be discussed in this section as percentages. Fig. 6 shows that estimating the safe region makes the robot control achieve
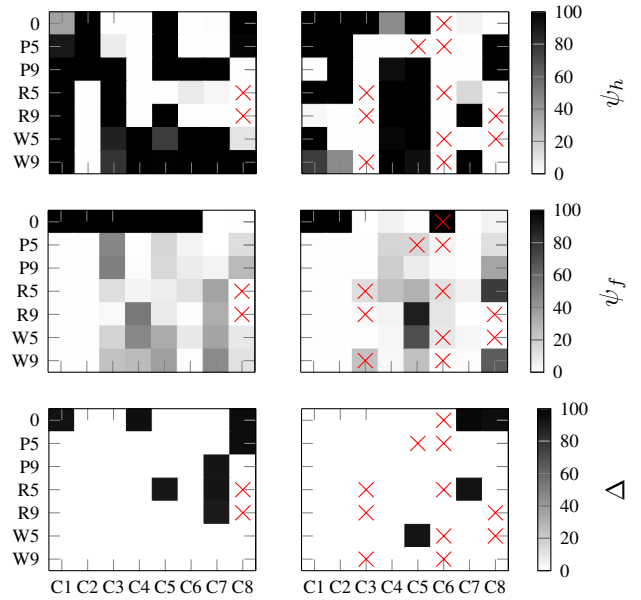
safer handovers across the recordings and scenarios (points near the bottom-right corner). When the object centroid is not occluded by the hand, ψ_h is close to 100 regardless of the use of the safe region (points near the top-right corner). When the visible area is too small ($\approx w_g$), grasping the container is challenging even when considering the safe region (points near the bottom-left corner). Note that there are a few cases where ψ_h is low when selecting the safe region for grasping due to errors in the robot movements when reaching the target. Fig. 7 shows that the algorithm achieves high human safety ($\psi_h > 80$) across filling type and level configurations for wine glass (C5), champagne flute glass (C6), beer cup (C7), cocktail glass (C8). This is due to limited hand-occlusions, accurate object segmentation, or the control algorithm that keeps the robot at a safe distance when there are severe occlusions. Note that scenario 2 is more challenging than scenario 1 for both human and object safety measures due to hand occlusions at the start of the video leading to wrong segmentation of the object (failures shown by red crosses). Object safety is maximum for empty opaque containers, $\psi_f = 100$, due to our assumption that the container is empty when we cannot estimate the content. Moreover, safeness increases for non-empty transparent containers, as the vision algorithm is more accurate when the filling is visible. Overall, our framework successfully grasps (high ψ_f) and accurately delivers ($\Delta > 90$) the containers for 12 out of 112 recordings, while avoiding spilling (as we would expect in reality). Moreover, the robot does not perform approaching and container grasping for 35 out of 112 recordings due to low safeness and to ensure the safety of the human.

In Tab. I, we compare the object safety based on the mass

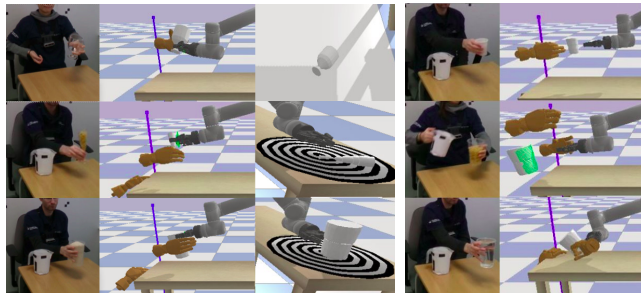|		ES	BIT	HVRL	VA2M	ES	BIT	HVRL	VA2M
E	$\delta_m(g)$	253.28	0.00	0.00	0.00	273.14	0.00	0.00	0.00
	ψ_f	0.00	100.00	100.00	100.00	0.57	100.00	100.00	100.00
	$\bar{\psi}_f$	0.00	42.37	42.37	42.37	0.19	45.44	45.44	45.44
	Δ	80.38	92.36	92.36	92.36	93.70	94.13	94.79	93.48
P5	$\delta_m(g)$					140.85	-29.95	-45.60	3.40
	ψ_f					13.16	64.97	51.87	95.22
	$\bar{\psi}_f$					27.10	1.79	5.55	0.25
	Δ					0.00	92.79	92.17	0.00
P9	$\delta_m(g)$	118.96	-36.53	-63.10	-99.53	109.07	-51.17	-91.35	-0.31
	ψ_f	4.48	38.54	19.26	7.44	26.73	53.85	33.12	99.63
	$\bar{\psi}_f$	5.88	41.85	21.71	8.75	NE	NE	NE	NE
	Δ	91.54	91.41	91.40	0.00	0.00	0.00	0.00	0.00
R5	$\delta_m(g)$	52.94	97.66	-200.78	-209.91				
	ψ_f	35.65	14.92	2.00	1.68				
	$\bar{\psi}_f$	15.56	9.71	1.99	1.38				
	Δ	93.60	93.62	0.00	0.00				
R9	$\delta_m(g)$	103.68	-154.11	-99.85	-403.24				
	ψ_f	32.07	18.44	33.44	1.20				
	$\bar{\psi}_f$	29.81	20.97	29.32	0.50				
	Δ	89.12	0.00	82.58	0.00				
W5	$\delta_m(g)$	-64.41	-76.37	-120.44	-266.18	-235.00	-81.74	304.16	-178.00
	ψ_f	35.44	29.23	14.38	1.38	8.91	43.13	4.38	16.02
	$\bar{\psi}_f$	NE	NE	NE	NE	NE	NE	NE	NE
	Δ	0.00	0.00	0.00	0.00	0.00	90.59	97.68	94.90
W9	$\delta_m(g)$					-289.49	-210.37	-118.32	-324.61
	ψ_f					11.25	20.44	40.94	8.63
	$\bar{\psi}_f$					12.14	20.46	2.69	9.19
	Δ					0.00	0.00	0.00	95.76

Fig. 8. Samples of human-to-robot handover results. Successful handovers (left) may end with unsuccessful delivery (dropped or tipped over), whereas unsuccessful handovers (right) are caused by failures of the vision algorithm (mismatch in hand and container position, incorrectly tracking of the jug instead of the container, inaccurate hand pose).

predicted by our vision algorithm against HVRL, BIT, and VA2M on the two CCM test sets (beer cup and cocktail glass) for scenario 1. Note that the object is occlusion-free in the first frames of the recordings of this scenario. Correctly classifying filling type and level is challenging for our vision algorithm (ES) due to the varying container shapes and ambiguities, such as a container filled with water can be confused with an empty container. This results in a ψ_f of around 30 and an overestimation of $\tilde{m}$, indicating chances of deformation or breaking of the container in reality. By exploiting the audio modality, BIT, HVRL, and VA2M are often more accurate in estimating $\tilde{m}$, resulting in higher ψ_f. This is due to their ability to identify filling type and level by the characteristics and duration of the audio, even when the container is occluded by the hand. While object safety is maximum for empty containers ($\psi_f = 100$), the probability decreases for cases with the presence of content. This is caused by the underestimation of the filling mass and hence we would expect the container to slip and fall,

as the robot might not apply enough force. Moreover, we compute a complementary object safety $\bar{\psi}_f$ that replaces in Eq. 5 the predicted force with the applied total normal force measured in the simulator after grasp execution. This force includes additional factors that we did not model or control, e.g. slipping between the contact surfaces and the shape of the container. Despite $\psi_f > 0$ before grasping execution, we can observe that there are cases where the robot fails to complete the execution of the grasp with the real mass (NE), e.g. container filled with water up to 50% due to the hand occluding the container. Moreover, there are many cases where object safety substantially differs before and after the grasping execution, e.g. VA2M decreases from 95.22 to 0.25 or our vision algorithm increases from 13.16 to 27.10 for P5. This confirms that the object safety should consider the additional factors that may also occur in a real human-to-robot handover. Finally, we observe that the delivery accuracy is high ($\Delta \geq 80$) for all algorithms and all cases where the robot successfully held the container. This shows that the robot manoeuvring would have been stable enough to avoid spilling the content.

In Fig. 8, we show samples of human-to-robot handovers with our real-to-simulation frameworks, visualising the estimated safe region (in green) and the outcomes, such as successful receiving and holding, delivery, and failures.

VI. CONCLUSION

We proposed a real-to-simulation framework that estimates the physical properties of unknown containers and their content from real videos of people manipulating the objects. The simulation part allows the framework to develop and assess algorithms for safe human-to-robot handovers of these objects. For human safety, we proposed to estimate the safe grasp region for the robot gripper to avoid the human hand when receiving the object. We also quantified the safeness of the human and object during the handover. We analysed the safety measures and tested our vision baseline compared to various approaches and sensing modalities (vision, sound).

Future work will include the validation of the algorithms discussed in this paper with a real setup as well as the simulation of deformable objects and dynamics of the content.

References

[1] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2016.

[2] R. Sanchez-Matilla, K. Chatzilygeroudis, A. Modas, N. Ferreira Duarte, A. Xompero, P. Frossard, A. Billard, and A. Cavallaro, "Benchmark for human-to-robot handovers of unseen containers with unknown filling," *IEEE Robotics Autom. Lett.*, vol. 5, no. 2, 2020.

[3] S. Donaher, A. Xompero, and A. Cavallaro, "Audio classification of the content of food containers and drinking glasses," in *Europ. Signal Proc. Conf.*, 2021.

[4] R. Ishikawa, Y. Nagao, R. Hachiuma, and H. Saito, "Audio-visual hybrid approach for filling mass estimation," in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.

[5] V. Iashin, F. Palermo, G. Solak, and C. Coppola, "Top-1 CORSMAL challenge 2020 submission: Filling mass estimation using multi-modal observations of human-robot handovers," in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.

[6] Q. Liu, F. Feng, C. Lan, and R. H. M. Chan, "VA2Mass: Towards the fluid filling mass estimation via integration of vision & audio learning," in *IEEE Conf. Pattern Recognit. Workshops and Challenges*, 2021.

[7] A. Modas, A. Xompero, R. Sanchez-Matilla, P. Frossard, and A. Cavallaro, "Improving filling level classification with adversarial training," in *IEEE Int. Conf. Image Process.*, 2021.

[8] N. F. Duarte, K. Chatzilygeroudis, J. Santos-Victor, and A. Billard, "From human action understanding to robot action execution: how the physical properties of handled objects modulate non-verbal cues," in *IEEE Int. Conf. Development Learning Epigenetic Robotics*, 2020.

[9] P. Rosenberger, A. Cosgun, R. Newbury, J. Kwan, V. Ortenzi, P. Corke, and M. Grafinger, "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robotics Autom. Lett.*, vol. 6, no. 1, pp. 17–23, 2020.

[10] W. Yang, C. Paxton, M. Cakmak, and D. Fox, "Human grasp classification for reactive human-to-robot handovers," in *IEEE Int. Conf. Intell. Robot Syst.*, Virtual, 2020.

[11] H. Nemlekar, D. Dutia, and Z. Li, "Object transfer point estimation for fluent human-robot handovers," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2627–2633.

[12] D. Vogt, S. Stepputtis, B. Jung, and H. B. Amor, "One-shot learning of human–robot handovers with triadic interaction meshes," *Autonomous Robots*, vol. 42, no. 5, pp. 1053–1065, 2018.

[13] M. K. Pan, E. A. Croft, and G. Niemeyer, "Evaluating social perception of human-to-robot handovers using the robot social attributes scale (rosas)," in *ACM/IEEE Int. Conf. Human-Robot Inter.*, 2018.

[14] A. Fishman, C. Paxton, W. Yang, D. Fox, B. Boots, and N. Ratliff, "Collaborative behavior models for optimized human-robot teamwork," arXiv:1910.04339 [cs.RO], 2020.

[15] F. Riccio, R. Capobianco, and D. Nardi, "Learning human-robot handovers through π-stam: Policy improvement with spatio-temporal affordance maps," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2016.

[16] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *IEEE Int. Symposium Robot Human Inter. Comm.*, 2012.

[17] N. Liu, Y. Cai, T. Lu, R. Wang, and S. Wang, "Real–sim–real transfer for real-world robot control policy learning with deep reinforcement learning," *Applied Sciences*, vol. 10, no. 5, 2020.

[18] A. Bestick, R. Bajcsy, and A. D. Dragan, "Implicitly assisting humans to choose good grasps in robot to human handovers," in *Int. Symposium Exper. Robotics*, 2016.

[19] M. Webster, D. Western, D. Araiza-Illan, C. Dixon, K. Eder, M. Fisher, and A. G. Pipe, "A corroborative approach to verification and validation of human–robot teams," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 73–99, 2020.

[20] N. Vahrenkamp, H. Arnst, M. Wächter, D. Schiebener, P. Sotiropoulos, M. Kowalik, and T. Asfour, "Workspace analysis for planning human-robot interaction tasks," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2016.

[21] A. Xompero, R. Sanchez-Matilla, R. Mazzon, and A. Cavallaro, "CORSMAL Containers Manipulation." [Online]. Available: http://corsmal.eecs.qmul.ac.uk/containers_manip.html

[22] A. Fernandes and R. Ventura, "Human-robot collaboration in micro-gravity: The object handover problem," in *Int. Symposium Artificial Intelligence, Robotics and Automation in Space*, 2020.

[23] U. Dombrowski, T. Stefanak, and J. Perret, "Interactive simulation of human-robot collaboration using a force feedback device," *Procedia Manufacturing*, vol. 11, pp. 124–131, 2017.

[24] S. Meyer zu Borgsen, P. Renner, F. Lier, T. Pfeiffer, and S. Wachsmuth, "Improving human-robot handover research by mixed reality techniques," in *Inaugural Int. Workshop Virtual, Augmented and Mixed Reality for Human-Robot Interaction*, 2018.

[25] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, vol. 22, no. 2, pp. 149–164, 2007.

[26] M. J. Rosenstrauch and J. Krüger, "Safe human-robot-collaboration-introduction and experiment using ISO/TS 15066," in *Int. Conf. Control, Autom. Robotics*, 2017.

[27] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[28] C. T. Landi, F. Ferraguti, S. Costi, M. Bonfè, and C. Secchi, "Safety barrier functions for human-robot interaction with industrial manipulators," in *European Control Conf.*, 2019.

[29] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Tran. Automation Science and Eng.*, vol. 13, no. 2, pp. 882–893, 2015.

[30] ISO/TS 15066:2016, "Robots and robotic devices – collaborative robots," International Organization for Standardization, Standard ISO/TS 15066:2016, 2016.

[31] M. Kaboli, K. Yao, and G. Cheng, "Tactile-based manipulation of deformable objects with dynamic center of mass," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2016.

[32] J. Xu, M. Danielczuk, J. Ichnowski, J. Mahler, E. Steinbach, and K. Goldberg, "Minimal work: A grasp quality metric for deformable hollow objects," in *IEEE Int. Conf. Robotics Autom.*, 2020.

[33] A. Xompero, R. Sanchez-Matilla, A. Modas, P. Frossard, and A. Cavallaro, "Multi-view shape estimation of transparent containers," in *IEEE Int. Conf. Acoustics, Speech Signal Process.*, 2020.

[34] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[35] M. Johannes, J. Bigelow, J. Burck, S. Harshbarger, M. Kozlowski, and T. Doren, "An overview of the developmental process for the modular prosthetic limb," *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, vol. 30, pp. 207–216, 2011.

[36] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[37] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[38] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[39] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis.*, 2017.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Eur. Conf. Comput. Vis.*, 2018.

[41] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. Torr, "Fast online object tracking and segmentation: A unifying approach," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.

[42] K. Mamou, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, E. Lengyel, Ed. A K Peters, 2016.

[43] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization-based full body control for the darpa robotics challenge," *J. Field Robotics*, vol. 32, no. 2, pp. 293–312, 2015.

[44] K. Dandekar, B. I. Raju, and M. A. Srinivasan, "3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense," *J. Biomech. Eng.*, vol. 125, no. 5, pp. 682–691, 2003.