ARnnotate: An Augmented Reality Interface for Collecting Custom Dataset of 3D Hand-Object Interaction Pose Estimation

Xun Qian*
qian85@purdue.edu
School of Mechanical Engineering, Purdue University
West Lafayette, IN, USA

Fengming He*
he418@purdue.edu
School of Electrical & Computer Engineering, Purdue University
West Lafayette, IN, USA

Xiyun Hu
hu690@purdue.edu
School of Mechanical Engineering, Purdue University
West Lafayette, IN, USA

Tianyi Wang
wang3259@purdue.edu
School of Mechanical Engineering, Purdue University
West Lafayette, IN, USA

Karthik Ramani
ramani@purdue.edu
School of Mechanical Engineering, Purdue University
West Lafayette, IN, USA

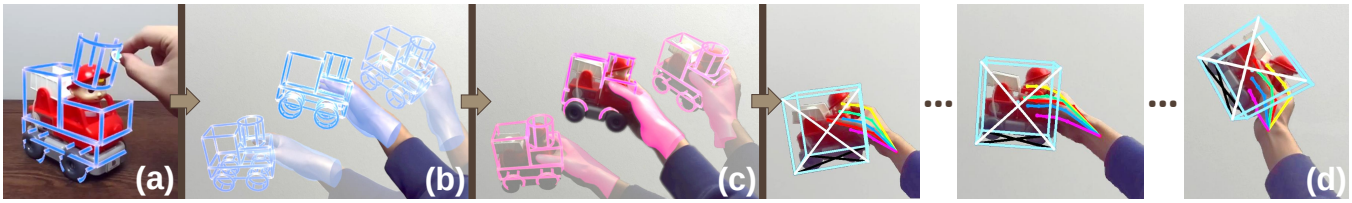


Figure 1: Overview of the ARnnotate workflow. (a) A user creates a *bounding contour* as the virtual representation of a target physical object in AR. (b) The user manipulates the *bounding contour* with a preferred gesture, while ARnnotate keeps recording the 3D poses of both the *bounding contour* and hand obtained by a hand-tracking-capable AR-HMD as the dataset labels. (c) ARnnotate replays the record as an *interaction clip* in AR. The user grabs the physical object using the same gesture and manipulates it while ensuring both the hand and object are accurately aligned with the counterparts of the *interaction clip*. ARnnotate captures the user's first-person view as the dataset images and temporally pairs them with the corresponding labels via back-end processing. (d) The custom dataset created by ARnnotate.

ABSTRACT

Vision-based 3D pose estimation has substantial potential in hand-object interaction applications and requires user-specified datasets to achieve robust performance. We propose ARnnotate, an Augmented Reality (AR) interface enabling end-users to create custom data using a hand-tracking-capable AR device. Unlike other dataset collection strategies, ARnnotate first guides a user to manipulate a virtual bounding box and records its poses and the user's hand joint positions as the labels. By leveraging the spatial awareness of AR, the user manipulates the corresponding physical object while following the in-situ AR animation of the bounding box and hand model, while ARnnotate captures the user's first-person view as the images of the dataset. A 12-participant user study was conducted, and the results proved the system's usability in terms of the spatial accuracy of the labels, the satisfactory performance of the deep neural networks trained with the data collected by ARnnotate, and the users' subjective feedback.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**; **Interactive systems and tools**.

KEYWORDS

Augmented Reality, Hand-Object Interaction, 3D Pose Estimation, Dataset Collection

ACM Reference Format:
Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, and Karthik Ramani. 2022. ARnnotate: An Augmented Reality Interface for Collecting Custom Dataset of 3D Hand-Object Interaction Pose Estimation. In *The 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22), October 29-November 2, 2022, Bend, OR, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3526113.3545663

*Both authors contributed equally to this research.

UIST '22, October 29-November 2, 2022, Bend, OR, USA

ACM ISBN 978-1-4503-9320-1/22/10.
https://doi.org/10.1145/3526113.3545663

1 INTRODUCTION

Humans use hands to interact with physical objects and tools in everyday life and work. With the advents of hardware devices and computational algorithms, Human-Computer Interaction (HCI) researchers have exploited the information behind the hand-object

interaction in practical applications such as daily activity monitoring [45, 78], interaction-triggered context-aware applications [72], engineering task tutoring [17, 32, 48], and tangible Mixed Reality (MR) interfaces [37, 80]. Among these works, the researchers have gradually embraced vision-based 3D pose estimation deep neural networks [61, 76] as the hand-object interaction perception approach owing to their high reliability and scalability. Typically, these works prove the HCI-oriented system usability through laboratory experiments while adopting the networks trained with bench-marking datasets [6, 19, 24, 82].

However, for 3D hand-object interaction detection, a network trained with a pre-designated dataset may not cover the diversified real-world scenarios, which significantly limits the in-the-field deployment of the above-mentioned applications. In specific, while, an object pose estimation network easily fails when a user interacts with an object that is visually distinct from its training data counterparts (e.g., a plain-color 'cup' in a bench-marking dataset versus the user's 'cup' in a different shape and with decorations); on the other hand, a pre-trained network has limited performance when a specific application context is not considered during training. For example, a hand pose estimation network trained using a daily-object dataset may malfunction in industrial scenarios (e.g., machine repair and assembly tasks) because the objects involved, their background scenes, as well as the object manipulation ways can be significantly different. In light of this, we aim to assist end-users to collect object-specified and task-specified datasets and train the networks that can achieve satisfactory performance when the same users consume the applications.

With regards to the 3D hand-object interaction dataset collection, some works [1, 62] allow users to first capture images, then label the 3D poses of the involved objects using a post-hoc 2D user interface following the idea of the 2D labeling tools [59]. Nevertheless, they become infeasible when the 3D hand poses are taken into consideration. The inevitable hand-object occlusions hamper users from labeling the hand joints hidden behind the object on an image. Further, the cognitive load for the annotators and the number of operations to convert the 3D-domain hand-object interaction as the labels on a 2D image are high. Typically, an annotator has to first understand the 3D spatial relationship between a hand skeleton and an object, then manipulate the 3D labels using the projected 2D labels as visual feedback. In addition, it is tedious to mark over thousands of images where each image contains more than 20 hand joints. On the other hand, Computer Vision (CV) researchers place multiple cameras and sensors in laboratory environments to obtain the 3D poses of either hands or objects [6, 19, 40, 82], while other works adopt optimization algorithms to synthesize or estimate the 3D poses as labels [24, 27]. Compared with the post-hoc interface ideas, these works not only solve the occlusion issue, but can also generate both the images and labels concurrently through continuous recordings, which significantly improve the efficiency. However, since they mainly serve research purposes, most require additional hardware setups, or the target objects are limited to the ones included in other bench-marking datasets. Consequently, it is impracticable for ordinary users to appropriate these dataset collection systems ad-hoc. Thus, we are highly motivated to explore a dataset collection approach that addresses both the hand-object occlusion problem and the feasibility of out-of-lab usage.

The emerging Augmented Reality (AR) technology shows a strong potential to fulfill the needs. First, the spatial awareness of AR allows for pervasive perception of the digital elements' 3D poses with respect to the AR device. Meanwhile, bare-hand tracking has been embedded in the recent off-the-shelf AR Head-Mounted Device (AR-HMD) [51], which supports accurate 3D hand skeleton detection when no occlusion happens. Together with the image capture capability, such a human-centered device has the potential to support users in continuously and fluently collecting 3D object and hand labels and the corresponding images in any local environment. More importantly, with the spatial awareness of AR, pre-recorded digital contents can be fixed in mid-air as the spatial reference of embodied tasks [8, 25, 26, 73]. These works typically consist of two sequential steps where users first record specific 3D hand and body movements, then, the system replays them in AR for the users to align their hands and bodies with the digital counterparts to complete the tasks. In our work, we adopt a two-step dataset collection process as analogous to these prior workflows to solve the hand-object occlusion issue. In specific, a user first generates the labels of the object and hand by manipulating the virtual bounding box, while the bare-hand gesture is being detected by the AR-HMD, then records the images when spatially following the replaying AR animation of the hand-object interaction. In this way, the occlusion is completely decoupled from the labeling stage, while the two separately collected parts of the dataset can be paired to form the final dataset without additional effort.

We propose ARnnotate, an AR-based system that supports user-specified dataset collection for 3D hand and object pose estimation. With ARnnotate, a user first creates a virtual 3D bounding box that preserves a physical object's geometric features by spatially referring to the object in AR. Next, the user grabs the bounding box in a preferred gesture of holding the physical object, and starts to manipulate it in mid-air. At the same time, ARnnotate records the 3D poses of the user's hand and bounding box from the hand-tracking-capable AR-HMD as the dataset labels. Then, ARnnotate displays the record as an AR animation with the moving bounding box and hand mesh model, and the user manipulates the physical object in the same hand gesture while accurately aligning both the hand and object with the AR counterparts. Meanwhile, ARnnotate captures the dataset images that are automatically paired with the corresponding labels, and completes the dataset collection.

We highlight our contributions as: (1) An AR-based sequential workflow for pervasive and continuous collection of custom hand-object pose estimation datasets while addressing the hand-occlusion issue. (2) An AR interface with front-end visual assistance and back-end computational processes that supports end-users in creating high-quality datasets. (3) A systematic evaluation in terms of the spatial accuracy of the collected labels, the performance of the networks trained with the datasets, and the qualitative feedback from the users.

2 RELATED WORK

2.1 Vision-Based Hand-Object Interaction Applications in the HCI Area

Using hands to manipulate objects and tools is one of the most dominant ways in human-object interaction in daily living and work

[60]. With the advances in hardware and software, HCI researchers have been able to digitalize the hand-object interaction into the computational domain for an expanding range of applications such as monitoring and analyzing daily activities in smart environments [42, 45, 78, 79], education [67], and industrial tutoring [2, 14].

Among the techniques for hand-object interaction perception, using ego-centric-view-based deep learning approaches to extract the 3D poses of hands and objects has drawn HCI researchers' attention because of the always-on perception of human activities and the simple setup for a higher deployment scalability. Ego-Topo [54] builds topological maps that represent human interactions using first-person videos. Nagarajan et al. [53] develops a system that infers the hotspots of the available object interactions based on egocentric videos. Wang et al. [74] and Lee et al. [43] enable robots to assist workers by understanding the assembly interactions through first-person-perspective live streams and videos. Vizlens [23] and Lee et al. [44] support vision-impaired users to interact with daily objects and interfaces. Recently, with the flourishing development of AR-HMDs, researchers have developed AR/MR interfaces that dynamically respond to interactions with the surrounding objects. Kosch and Schmidt [37] augment everyday objects with tangible digital interfaces that can react to hand-object interactions. Gripmarks [80] generates MR interfaces on the handheld object based on the detected gesture and object identity. CAPturAR [72] supports users to build personalized context-aware applications which trigger various AR functions by hand-object interactions. AdapTutAR [32] adaptively shows different AR tutoring elements to learners by inferring the learning progress from the detected interactions with the machine interfaces. Recently, ScalAR [58] assists domain designers to author semantically adaptive AR contents that can dynamically change the spatial placements with respect to the physical objects detected in different environments.

However, most works focus on the novelties in terms of the HCI-oriented design, and conduct preliminary evaluations of the system usability by implementing the 3D pose estimation networks trained with limited bench-marking objects and gestures or alternate algorithms that imitate the perception of the 3D hand-object interaction. Yet, many promising use cases are still being explored at a conceptual level due to the limited performance of the hand and object pose estimation networks when being deployed in real-world environments. Because of the characteristics of deep learning, training a network using the dataset collected from the target application scenarios instead of using research-oriented datasets with pre-designated objects and interaction gestures plays an essential role in improving the network performance. Hence, we strive to explore a system that supports collecting site-specific and task-specific 3D hand and object pose estimation datasets, and therefore, to enable HCI researchers to realize the diversified use cases using the improved hand-object interaction detection networks.

2.2 3D Hand and Object Pose Estimation Dataset Collection

Multiple approaches have been explored to collect 3D hand and object pose datasets. Here, we investigate the practicability of adopting them to collect user-specified datasets.

Some works appropriate the broadly adopted idea of post-hoc labeling in 2D dataset creation [59] to the 3D object pose labeling domain. Typically, a user first collects the images of the target objects in the local environment, then creates the 3D bounding boxes with the assistance of the system, which allows the user to freely collect datasets at need. Objectron [1] provides an interface that allows users to draw 3D bounding boxes in key frames and manually adjusts the bounding boxes' poses calculated by the system for other frames. SUN RGB-D [62] focuses on indoor objects standing on the floor/table without elevation change and designs a web-based tool for extruding 3D bounding boxes from the 2D rectangles drawn by users. However, they become infeasible when being implemented in the hand-object interaction dataset collection. It is troublesome and even impossible for end-users to label the 3D hand and object poses based on the 2D images when hand-object mutual occlusions happen. Specifically, the 3D hand joints hidden by the object cannot be labeled, while the separated annotated labels cannot preserve the 3D hand-object relationships across all the data samples. In addition, they require tedious workloads to label over 20 3D hand joints for over thousands of images.

On the other hand, researchers set multiple external cameras [6, 40, 82] and wearable sensors [19] to concurrently collect the images and infer the hand and object poses through computational algorithms. These approaches resolve the hand-object occlusion issue and achieve high efficiency of labeling with the support of the additional setups. However, the ubiquity of the hand-object interaction in real scenarios requires the datasets to cover different contexts, and sometimes, within a large environment. Thus, it is cumbersome for end-users to set up the equipment. More importantly, for the users without domain expertise, the quality of the collected datasets cannot be guaranteed due to the complicated hardware system calibration and configuration. In contrast, HOnnotate [24] adopts optimization algorithms to estimate the data labels using much simpler hardware setups, while ObMan [27] synthesizes hand-object interaction labels based on manipulation constraints. Yet, the objects considered by these research-oriented processes are constrained to the ones included in other existing bench-marking datasets [76, 82].

Recently, leveraging the emerging AR technology, LabelAR [41] develops a spatial interface that supports users to rapidly label object poses using an AR-capable device by projecting in-situ placed 3D bounding boxes onto the corresponding 2D images. While this approach addresses both the efficiency and scalability with the help of AR, it only focuses on the static objects placed on the table, and does not touch the hand-object interaction when users need to dynamically manipulate objects with hand occlusions involved.

In light of the pros and cons of these approaches, we aim to develop an interface that supports users to complete the custom dataset creation in an intuitive and effortless way, while resolving the troublesome hand-object occlusion issue.

2.3 Leveraging Spatial Awareness of AR

Augmented Reality (AR) shows a strong potential to resolve the issues in hand and object pose dataset collection. Thanks to the 3D spatial awareness capability provided by the emerging AR technology, the relative transform of the virtual contents with respect to

the AR device is constantly accessible. Meanwhile, the recent off-the-shelf AR-HMD [51] has embedded the hand tracking capability that keeps providing the 3D hand skeleton for freehand interaction with virtual contents [13, 71] where no occlusion exists. With this integrated device, a user's first-person view, and the 3D spatial information of both the virtual contents and the hand skeleton can be simultaneously recorded. Thus, we are motivated to enable users to create custom 3D hand and object labels and images in a pervasive and continuous manner by only wearing a portable hand-tracking-capable AR-HMD.

Furthermore, enabled by the spatial awareness, virtual contents can be fixed in mid-air as spatial reference. Typically, researchers propose various systems that first record users' body and hand movements, then replay them in AR for users to utilize as spatial reference. LightPaintAR [73] facilitates photographers to precisely move a light source in 3D while following a pre-recorded AR trace for light-painting photography. In body movement tutoring systems, a user can move with an animated virtual body that is overlaid on the user's first-person view in AR [25, 26, 29]. By following a moving full-body AR avatar that is pre-recorded by an expert, a learner can rapidly master embodied machine operations [8]. In remote collaboration, a local user is able to complete hand-object interaction tasks by aligning both hands with the point cloud of a remote helper's hands [18]. GhostAR [9] enables designers to author human-robot-collaboration tasks by referring to the AR avatar externalized from the pre-recorded human action.

By using AR as spatial reference, users' dynamic activities in the virtual and physical domain can be temporally split but spatially aligned, which is highly similar to the hand-object interaction dataest collection process where the images are captured when users physically interact with the object and the labels are spatially annotated onto the images in the virtual domain. Meanwhile, these works have proved that users are able to perform accurate 3D movements following the in-situ animated digital contents with the help of the depth perception of AR. Therefore, by imitating the workflows mentioned above, we propose to allow users to first record the labels of hands and objects by manipulating virtual bounding boxes using the hand-tracking-capable AR-HMD and save them as AR animations. Then, the users perform the physical-domain hand-object interaction while following the in-situ AR animations so that the users' first-person view can be captured as the dataset images. In this way, not only the occlusion issue is intrinsically resolved since the labeling process purely happens in the virtual domain, but the pairing of the images and labels is also fluently accomplished when the users spatio-temporally align the physical objects with the in-situ AR animations.

3 ARNNOTATE SYSTEM DESIGN

In this section, we demonstrate the system design of ARnnotate. First, we provide an overview of the system workflow using a specific task as an example. Next, we elaborate on the system features that assist end-users to succeed in the collection of hand-object interaction datasets while addressing the considerations discussed in the previous sections.

3.1 System Walk-through

The overall workflow of ARnnotate is illustrated in Figure 2. As addressed in the previous sections, instead of collecting the images and labels of a dataset concurrently [19, 24] or in a post-hoc labeling manner [1, 22, 62], ARnnotate adopts a sequential process of first creating the digital hand and object pose labels, then capturing the images of the physical-domain hand-object interaction. The two parts are temporally paired when users spatially align the object and hand with the animated labels in AR. In the following subsections, we will sequentially cover in detail every step illustrated in the workflow.

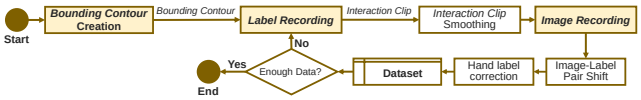


Figure 2: ARnnotate system workflow.

Here, we walk-through the workflow of ARnnotate using the scenario shown in Figure 1 where an end-user wants to collect data of interacting with a toy car. The user first creates a ***bounding contour*** that holds the same geometric features of the physical toy car by spatially aligning multiple virtual primitives with proper sizes with the corresponding elements of the toy car in AR (Figure 1a). Next, the user starts to collect the dataset. ARnnotate separates the dataset collection into two sequential steps: ***label recording*** and ***image recording***. To start the *label recording* step, the user grabs the *bounding contour* in the same way as grabbing the physical toy car, and starts to manipulate the *bounding contour* in AR (Figure 1b). With the spatial awareness and the hand-tracking capability of the AR-HMD, ARnnotate keeps recording the 3D positions and orientations of the *bounding contour* together with the 3D positions of the hand joints with respect to the AR-HMD, and saves them as an ***interaction clip***. Then, the user enters the *image recording* step. ARnnotate replays the *interaction clip* as an AR animation using the same *bounding contour* and a virtual hand model. The user now grabs the physical car in the same way as in the *label recording* by overlaying both the *bounding contour* and the virtual hand model with the corresponding physical elements with the help of the depth occlusion provided by the AR-HMD (Figure 1c). During the *image recording* step, the user ensures the toy car is accurately aligned with the animated *bounding contour* throughout the entire *interaction clip* with the help of visual reference and depth occlusion enabled by the AR-HMD. ARnnotate captures the user's first-person view from the AR-HMD as the dataset's images. Guided by ARnnotate, the user repeats the two steps for multiple trials until enough data have been collected (Figure 1d). In the following sections, we describe the system design of ARnnotate.

3.2 *Bounding Contour* Creation

In 3D object pose estimation, two major considerations affect the quality of the labeled data, and therefore, affect the performance of the network: (1) The orientation of the 3D bounding box should precisely represent the orientation of the physical object, and this orientation alignment should be consistent for all the potential poses of the physical object. For instance, from the perspective shown in Figure 1a, the 'upward' direction of the bounding box

should point upwards, the 'forward' direction of the bounding box faces towards the head of the car, and the 'right' direction is perpendicular to both the other two. (2) The physical object should be enclosed by the bounding box with the smallest volume given the orientation mentioned before. Besides these two concerns, specifically for ARnnotate, during the *image recording* (Figure 1c), users need to manipulate the physical object to align with the animated virtual bounding box, which requires instant recognition of the bounding box's spatial orientation. Thus, instead of using a simple cuboid as the bounding box, ARnnotate supports users to create a ***bounding contour*** that extensively preserves the geometric shape and characteristic features of the physical objects. Note that the main goal is not to replicate every detail of an object. Instead, users are expected to create the outline and some characteristic features of the object so that during the *image recording*, users are able to rapidly align the object's 6DOF with the replaying *interaction clip*.

Researchers have proposed to accurately create AR contents using physical objects and surfaces as spatial reference [5, 20, 33, 39, 41]. Following these works, we provide two methods to create a *bounding contour*: ***primitive creation*** and ***free-sketch creation***. For objects with non-symmetric regular geometric features (a milk box or a cooking pan), it is feasible to segment such an object into a combination of different standard primitives. ARnnotate provides these primitives and allows users to move/rotate/scale them to encase the corresponding parts of the physical objects (Figure 3a). For instance, a soft drink bottle can be represented by a cuboid as the main body and a small cylinder to indicate the cap (Figure 3a-1). For objects such as cups and spray bottles, they usually have complicated curves that are difficult to be represented by the primitives. We allow users to create 3D sketches in AR by directly referring to the shape of the physical objects as a complementary method such as the handle of a cup in Figure 3b-1. Meanwhile, users can leverage this method to create identifiable markers on regular-shaped objects such as sketching the logo of a wipe bottle and a paper box to indicate its orientation (Figure 3b-3 and b-4). Users can freely utilize the combination of these two methods to customize their own *bounding contour*s. Eventually, ARnnotate automatically converts them into the general bounding box broadly accepted by the CV area. The user operation details of the *bounding contour* creation will be explained in Section 3.5.

Figure 3: Example *bounding contour*s created by ARnnotate. (a-1 to a-4) *Bounding contour*s created by the *primitive creation* method. (b-1 to b-4) *Bounding contour*s created by both the *primitive creation* and *free-sketch creation* methods.

3.3 *Label Recording*

ARnnotate manages to allow a user to create the labels before collecting the images in order to eliminate the hand-object occlusion issue. During the *label recording*, our system saves the labels as an ***interaction clip***, an AR animation that includes the spatial movements of the *bounding contour* and the specific hand gesture to grab the object. Technically, an *interaction clip* is a time series with the 3D positions/rotations of the *bounding contour*, and the 3D positions of the 21 hand joints broadly adopted in hand datasets [81, 82]. Since the user needs to refer to these records as spatial reference in the *image recording* step, ARnnotate provides several textual and visual hints to guide the user to create *interaction clip*s that can be easily and precisely followed later.

3.3.1 Grabbing the bounding contour. In most single-hand-object interaction scenarios, a user grabs a physical object using an unchanged gesture throughout the entire manipulation (e.g., holding the handle of a cup using a 'fist' gesture or holding a flashlight using a 'semi-fist' gesture). In order to create the labels of the hand poses during the *label recording* step, we aim to allow users to grab the *bounding contour* in a same way analogous to grabbing the corresponding physical object. As addressed in the previous section, the *bounding contour* duplicates the characteristic features of the physical object. Therefore, using ARnnotate, a user first grabs the physical object in a preferred way, and aligns it with the *bounding contour* in mid-air. Then, the user releases the physical object while keeping the gesture unchanged. ARnnotate rigidly attaches the *bounding contour* with a virtual anchor where its pose is calculated as the average position and rotation of the user's five fingertips. Further, the ultimate goal of collecting the dataset is to train the networks used by the same user in daily living and work. So, ARnnotate suggests that the user performs the gestures that will be commonly used to interact with the current object, while the user still has the freedom to determine what gestures to perform. The corresponding guidance and suggestion are provided in clearly listed sequential bullet points via the user operational interface described in Section 3.5.

Moreover, unlike holding a physical object, it is difficult for the user to keep the gesture unchanged when grabbing a virtual object without haptic feedback. We introduce the ***gesture indicators*** that are attached to the user's grabbing hand's five fingertips to assist the user to keep the hand gesture unchanged (Figure 4a). During recording, each *gesture indicator* turns from green to yellow or red to warn the user if the finger moves too much from the initial position, while the thresholds for the two colors are empirically set to 1cm and 1.5cm. Moreover, ARnnotate marks the frame as invalid when more than two *gesture indicators* turn red, and deletes the corresponding data during the data post-processing.

3.3.2 Label recording by manipulating the bounding contour. Prior works [49, 50] discuss the benefits of DoF separation when manipulating a virtual object in 3D, and prove that the DoF separation achieves precise manipulation outcome. In order to ensure users can accurately follow the recorded *interaction clip* in the *image recording*, we encourage users to interact with the *bounding contour* in two manners, namely, translation-dominant manipulation and

rotation-dominant manipulation. For the translation-dominant manipulation, users move the *bounding contour* in any trajectory they prefer without rotating their wrists, while for the rotation-dominant manipulation, users majorly rotate the *bounding contour* without moving it in space. The textual guidance is provided through the text panel illustrated in the user interface in Section 3.5.

Meanwhile, an effective neural network requires the dataset to be diverse and contain an adequate number of data [31]. To guarantee the robustness of the training network, we introduce two indicators guiding users to achieve the corresponding requirements, namely, the ***orientation indicator*** and ***progress indicator***. Users can use the *orientation indicator* to check whether they have manipulated the objects from different viewpoints, which ensures that different poses of the current object are included in the dataset. The *progress indicator* informs users how many labels have been created so far and how many labels are left to reach a recommended dataset size. In ARnnotate, we attach the *orientation indicator* onto the *bounding contour*, which is a spherical surface with numerous hexatiles (Figure 4b). After the user grabs the *bounding contour*, we cast a ray from the AR-HMD along the direction between the AR-HMD and the *bounding contour* center. When the label recording starts, we visually hide the tiles that lie around the intersection point between the raycast and the *orientation indicator*. With this indicator, users are encouraged to record multiple *interaction clip*s in as many ways as they may grab the object in real application scenarios. For the *progress indicator*, it is a progress bar representing the percentage of the collected labels out of the target number of the dataset (Figure 4c). Users are expected to achieve full progress for the dataset collection. Note that since our work is a proof-of-concept and the network is trained for a user's customized usage, we do not need a large-scale dataset and the target data size is empirically set according to the preliminary training performance assessment mentioned in Section 4.

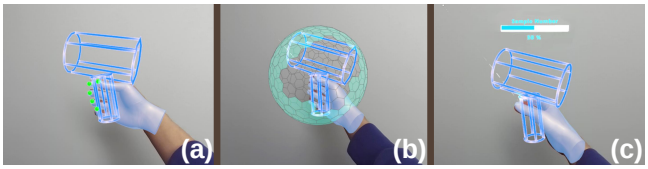


Figure 4: The visual indicators for the *label recording* step. (a) The *gesture indicators* are attached on the five fingertips of the hand that is grabbing the *bounding contour* to help keep the gesture unchanged during the recording. (b) The *orientation indicator* is bound with the *bounding contour* where the hexa tiles turn transparent if the corresponding orientation has been covered in the *interaction clip*s. (c) The *progress indicator* floating above the *bounding contour* indicates the overall dataset collection progress.

3.4 *Image Recording*

During the *label recording*, each *interaction clip* is recorded as a list of pose frames $[f_1, f_2, ..., f_n]$, where each frame contains the timestamp (f_i.**t**), the positions (f_i.*pos*) and rotations (f_i.*rot*) of the *bounding contour*, the set of the 21 hand joint positions (f_i.*h*), and the current image (f_i.*g*). At this moment, f_i.*g* is empty since no image has been recorded. Before and after the *image recording*, we adopt three algorithms that assist users to record the images f.*g* that can be accurately paired with the corresponding labels.

3.4.1 Interaction clip smoothing. When recording the labels, users may suddenly start/stop a movement or rapidly change the translation/rotation direction, which causes significant velocity changes. According to the prior arts [16, 75], such cases will increase the spatial inaccuracy of 3D manipulation. Following their guidance, before starting the *image recording* step, we pre-process the replaying *interaction clip* by clamping the linear and angular accelerations of the *bounding contour*. We adopt the Algorithm 1 for the pre-processing. We first batch the recorded frames $[f_1, f_2, ..., f_n]$ as $[b_1, ..., b_{n/k}]$, where each batch b_i contains k consecutive frames, denoted as $[f_{i1}, f_{i2}, ..., f_{ik}]$. For each batch b_i, we first calculate the average linear (b_i.**v**) and angular (b_i.ω) velocities, then calculate the linear and angular accelerations as: $b_i.\mathbf{a} = (b_i.\mathbf{v} - b_{i-1}.\mathbf{v})/(f_{i0}.t - f_{(i-1)0}.t)$, and $b_i.\alpha = (b_i.\omega - b_{i-1}.\omega)/(f_{i0}.t - f_{(i-1)0}.t)$, where $i = 1, ..., n/k$. When an acceleration value b_i.**acce** (**acce** $\in \{\mathbf{a}, \alpha\}$) exceeds the corresponding threshold, $\mathbf{acce}_{max}$, we increase the timestamps of all the frames within and after b_i by a calculated value so that the new acceleration of the current batch is equal to the corresponding threshold (Algorithm 1). We empirically set the thresholds $\Delta t_{max} = 0.5\ s$, $\mathbf{a}_{max} = 0.1\ m/s^2$,$\alpha = 1\ rad/s^2$ for the added time duration, linear acceleration, and angular acceleration respectively.

Algorithm 1 Pre-process an *Interaction Clip*

```
procedure PreprocessInteractionClip([b_1, b_2, ..., b_N])
    for i ← 2, N do
        if b_i.acce > acce_max then
            ▷ Extend the time duration of the current batch
            t_end ← (b_i.acce(f_ik.t − f_i1.t)) / (sign(b_i.acce)acce_max)
            Δt ← t_end − f_ik.t
            if Δt > Δt_max then
                Δt ← Δt_max
            end if
            for j ← 2, k do
                f_ik.t ← f_i1.t + (j − 1) Δt/(k−1)
            end for
            ▷ Postpone the timestamps of all the later frames
            for m ← i + 1, N do
                for j ← 1, k do
                    f_mj.t ← f_mj.t + Δt
                end for
            end for
        end if
    end for
end procedure
```

Now, a user can start the *image recording* using the processed *interaction clip*. ARnnotate further allows the user to adjust the average replaying speed to a proper value so that the user feels confident to follow the *interaction clip* accurately.

3.4.2 Image-label pair shift. After the *image recording* step, the recorded image-label pairs may involve temporal mis-alignments due to the latency between a user's movement and the animated

interaction clip. Following [16, 75], we utilize the velocity information calculated before ($b_i.\mathbf{v}$ and $b_i.\omega$), together with the replay speed ratio set by the user, $r_{global} \in (0.5, 1.5)$ to proportionally shift the label afterwards if the replaying linear/angular velocity ($b_i.\mathbf{vel} * r_{global}$) exceeds the corresponding velocity threshold, $\mathbf{vel_{max}}$ ($\mathbf{vel} \in \{\mathbf{v}, \omega\}$) (line 4-12 in Algorithm 1). Specifically, we add the time offset $\delta t = (b_i.\mathbf{vel} * r_{global} - \mathbf{vel}_{max}) * ratio$ to the timestamps of the labels (line 13-17 in Algorithm 1), where we adopted the same thresholds of ω_{max} and v_{max} as suggested in [16, 75], and the $ratio$ value is set to 20. The hand joint set $f_i.h$ keeps bound with the object poses in this step.

3.4.3 Hand label correction. Last but not least, in some cases where a user's hand is mostly perpendicular to the AR-HMD, tiny spatial errors of the hand joint positions may cause the labels completely detached from the hand. To avoid such scenarios in the final dataset, leveraging the depth perception supported by the AR-HMD, we record the depth images for each frame, and segment the images into background and foreground using [11]. After the *image-label pair shift*, ARnnotate checks whether all the 21 labeled hand joints fall into the foreground of each image (line 3-6 in Algorithm 2). When there is at least one joint lying in the background, ARnnotate searches around the neighboring m frames in case there is a better match between the current image and a nearby hand label (line 7-14 in Algorithm 2). We denote the number of joints of the p-th hand label ($f_p.h$) falling in the foreground of the q-th image ($f_q.g$) as $\mathbf{Num_{qp}}$. When searching for the alternative label for the q-th frame, ARnnotate first selects the labels with the maximum of Num_{qj} ($j \in \{q - \frac{m}{2}, \ldots, q + \frac{m}{2}\}$), i.e., the labels where most joints fall into the current frame's foreground, then matches the temporally nearest label with the current image (line 19-21 in Algorithm 2). The matching algorithm is shown in Algorithm 2 and the m value is empirically set to 8. Additionally, ARnnotate deletes the data where more than 3 joints lie in the background and the data with the *bounding contour* or hand out of view.

Algorithm 2 Image and Hand Label Matching

```
1:  procedure GetNewLabelArray([f_1, f_2, ..., f_N])
2:      Labels ← array[]
3:      ▷ Checks if all labeled joints fall into the foreground of the image
4:      for i ← 1 + m/2, N − m/2 do
5:          maxIdx ← i
6:          if Num_ii < 21 then
7:              ▷ Find a better matched label for the current image
8:              for j ← i − m/2, i + m/2 do
9:                  if Num_{i,j} > Num_{i,maxIdx} then
10:                     maxIdx ← j
11:                 end if
12:                 if Num_{i,j} = Num_{i,maxIdx} and |f_i.t − f_j.t| < |f_i.t − f_maxIdx.t| then
13:                     maxIdx ← j
14:                 end if
15:             end for
16:         end if
17:         add f_maxIdx.h to Labels
18:     end for
19:     for i ← 1 + m/2, N − m/2 do
20:         f_i.h ← Labels[i]
21:     end for
22: end procedure
```

3.5 ARnnotate Operational Interface

We describe the AR interface that supports all the needed operations. A *bounding contour creation menu* attached on a user's left hand is used to declare the name of the target physical object via the 'Object Name' button, and to create a *bounding contour* using the two methods described before via the 'Create' button (Figure 5a-1). A 'coordinate icon' (Figure 5a-1) fixed in mid-air indicates the initial orientation of the created *bounding contour*. The user can freely switch the method by pressing the corresponding button. For the *free-sketch* method, a brush sphere is attached on the user's right index fingertip. When the user performs a pinch gesture on the left hand, the system starts to draw 3D strokes based on the brush tip's position. The user can directly touch the physical object to get haptic feedback to improve the drawing accuracy (Figure 5a-2). For the *primitive* method, the user translates/rotates/scales it using the bare-hand interaction supported by our system (Figure 5a-3). The user can delete a stoke or a primitive by first pressing the 'Mode' button (Figure 5a-1), then simply touching the element to be deleted using the right index finger. Meanwhile, a text panel is also provided for showing the necessary textual guidance (Figure 5a-1). A *label recording menu* will appear above the *bounding contour* created by the user, where the user can 'Grab' the *bounding contour*, 'Record' *interaction clip*s, toggle on/off the *orientation indicator* and the *progress indicator*, and 'Delete' the *bounding contour* respectively (Figure 5b). For each recorded *interaction clip*, an *image recording menu* will be instantiated above the *bounding contour* (Figure 5c). The user can press the 'Animation' button to toggle on/off the corresponding *interaction clip*, adjust the speed of the clip using the *slider*, and start the *image recording* via the 'Record' button. ARnnotate will show a 5-second count-down, then hide all the unnecessary UIs for the user to concentrate on the object manipulation. After the current *interaction clip* is played once, all the UIs reappear to allow the user to start another recording session of the current *bounding contour*, or start over with a different physical object. The user can 'Delete' the *interaction clip* if needed.

4 IMPLEMENTATION

We build the hand-tracking-capable AR-HMD by combining an Oculus Quest 2 [56] with a front-facing stereo camera (ZED Dual AMP camera [63]). The system is developed using Unity3D (2020.3.18f1) and the freehand interaction for UI operation is supported by Microsoft Mixed Reality Toolkit (MRTK)[1]. During the *image recording*, images are saved at 1280 × 720 pixels on a connected PC at 15 frames per second. Object annotations are saved following the Objectron dataset format [1] and hand annotations are saved following the Panoptic Dataset format [36]. We select two networks for the evaluation of the datasets collected by our system. For the 3D object pose detection, we adopt the CenterPose [47] network which

[1] https://github.com/microsoft/MixedRealityToolkit-Unity

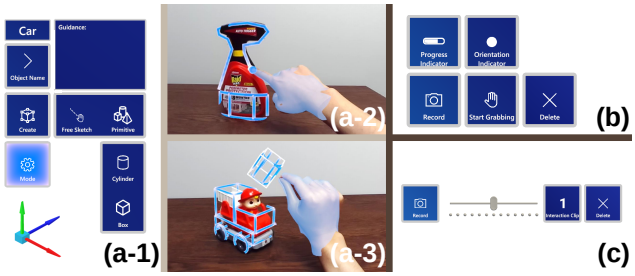


Figure 5: The AR interface of ARnnotate. (a-1) The *bounding contour creation menu*. Bottom-left: the 'coordinate icon' indicating the orientation of the *bounding contour*. (a-2) Users create 3D sketches using the brush tip attached on the right index fingertip while directly touching the physical surfaces. (a-3) Users manipulate a primitive using freehand interaction provided by ARnnotate to align it with the corresponding part on the physical object. (b) The *label recording menu* displayed above the *bounding contour*. (c) The *image recording menu* will be instantiated above the *label recording menu* after a new *interaction clip* is recorded.

was originally trained on the Objectron dataset. For the hand pose estimation, we first implement the OpenPose [10] approach, a robust hand pose estimation network against occlusions, to predict the 2D hand joint positions on the RGB image, then infer the 3D joint positions from 2D positions following [57]. Our work targets user-specified application scenarios rather than common datasets. To determine a proper data size that can achieve a decent training result, we trained the target object and hand pose estimation models with different data sizes ($1k, 1.5k, ..., 5k$), and chose the data size to be 2.5k for each object category since it is the minimum data size that reaches a result comparable with the bench-marking datasets [1, 82]. During training, data in each object category was shuffled and 80% of images were used for training while 20% of images were used for testing. To avoid the overfitting issue, we applied cross-validation method for each user's collected dataset. It took 3 hours for object pose estimation training per category per user and 4.5 hours for each user's hand pose estimation on a PC (Intel Core i7-9700k 3.6 GHz, 32 GB RAM, NVIDIA RTX 2080 GPU).

5 SYSTEM EVALUATION

We conducted a two-session user study to evaluate the three main considerations addressed in the above sections respectively: (1) The spatial accuracy and quality of the dataset created by our system, (2) whether we can train object and hand pose estimation neural networks with decent performance using the datasets generated by our system, and (3) the overall usability of the entire system.

12 users (8 males and 4 females, aging from 21 to 29) were recruited on campus. None of them had used our system before the user study. 3 users had machine learning and CV experience. Yet, this was not a comparison-based user study since our system targets any end-users and supports them to collect the dataset for their own usage. The entire study took 1.5 hours, and each user was paid with a $20 e-gift card. After a user came, we first introduced the background of our system and let the user get familiar with the entire system workflow, the system UIs, and the depth occlusion visual effect supported by the AR-HMD. During the user study, there was no additional guidance provided by the researchers besides the initial workflow introduction. For the guidance and suggestion described in Section 3.3, the users referred to the texts shown on the text panel (Figure 5 (a-1)) to complete each collection trial. After the two sessions, each user completed a 5-scale Likert-type questionnaire together with a standard System Usability Scale (SUS) questionnaire. Lastly, we took a conversation-type interview with the users to get their subjective feedback of the system.

5.1 Quantitative Evaluation of the Dataset Accuracy

Unlike the concurrent and post-hoc processes where the labeling inaccuracy may come from the hardware, algorithm, and annotator's expertise, the potential labeling error of ARnnotate is mainly attributed to the mis-alignments in the space-time domain. Thus, in this study session, we evaluate whether the labels generated for the corresponding images are accurate after all the pre- and post-processing approaches discussed in Section 3.

5.1.1 Procedure. Since a user is holding the physical object during the *image recording* step, the AR-HMD cannot provide the ground truth data. Meanwhile, it is impossible for the researchers to label the data post-hoc since no such interface exists as addressed in the Related Work section. Therefore, we adopt two external resources to obtain the ground truths of both the object poses and hand joint positions in real-time. This session was completed in the space shown in Figure 6a. We attached an up-facing Leap Motion Controller[2] to provide the ground truths of the hand via its hand tracking module[3], and mounted a downward-facing webcam for the 3D pose detection enabled by Vuforia Image Targets[4] while an image target was attached on a 10cm cube. Before the session started, the Leap Motion Controller and the webcam were calibrated into the AR-HMD's coordinate system. Each user was required to use our system to collect datasets of the cube in 6 trials, where each trial lasts for 20 seconds. To address the DoF separation concern discussed in Section 3, the first three trials required the users to move the cube in a translation-dominant way in three different directions, while the last three asked the users to rotate the cube in mid-air. Meanwhile, we recorded the 3D pose of the cube and the hand skeleton detected from the two devices for the real ground truths. The users were asked to hold the cube mostly from the bottom so that the Leap Motion Controller could detect the hand skeleton. But the users had the freedom to choose their own gestures. Figure 6b illustrates some of the gestures performed by the users. Following the prior works in the object manipulation area [16, 38, 66], we calculated the translation and rotation error between the ground truths and the labels generated by ARnnotate.

5.1.2 Result and Discussion. After the post-processing of ARnnotate, we collected 15130 and 15016 valid image-label-pair for the translation-dominant and rotation-dominant trials respectively. The spatial accuracy of the labels generated by ARnnotate is shown

[2]https://www.ultraleap.com/product/leap-motion-controller
[3]https://developer.leapmotion.com
[4]https://library.vuforia.com/features/images/image-targets.html

in Figure 6c. Regarding the object pose accuracy, the average translation errors were 0.94cm (SD=0.068) and 0.44cm (SD=0.021) for the translation-dominant and rotation-dominant trials respectively, with an overall average error to be 0.69cm (SD=0.033). Meanwhile, the average rotation errors were 1.70° (0.087) and 3.77° (SD=0.343) for the translation-dominant and rotation-dominant trials respectively, with an overall error to be 2.73° (SD=0.18). Given that the cube size is 10cm, following the results discussed in the prior works [16, 38, 66], ARnnotate could reach exceedingly accurate results in both the translation and rotation object manipulation. Typically, the remarkable performance of the translation accuracy in the rotation-dominant trails and the rotational accuracy in translation ones further proved the effectiveness of decoupling translation and rotation during virtual object manipulation discussed in Section 3.3. Regarding the hand, we followed the broadly used criterion to evaluate the hand joint accuracy [57]. We calculated the Mean Per Joint Position Error (MPJPE) of the 21 hand joints' 3D positions, and got 0.77cm (SD=0.026) and 0.92cm (SD=0.043) for the translation and rotation trials (overall error was 0.85cm with SD=0.011). The quantitative results demonstrated that the processing of the *interaction clips* could support users in accurately aligning the physical objects and hands with the moving virtual counterparts. Yet, as a dataset collection work, we still need to evaluate the quality of the dataset by investigating the training results.

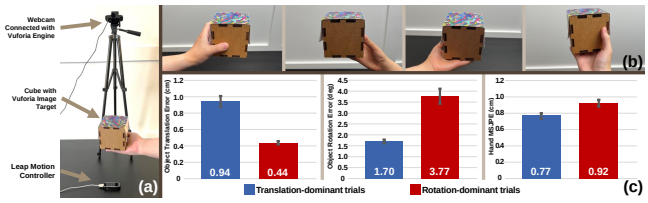


Figure 6: (a) The hardware setup of the dataset accuracy study. (b) Example gestures the users performed to hold the cube. (c) The results of the study.

5.2 Evaluation of the Dataset via Neural Network Training

In this session, we used the datasets generated by ARnnotate to train two broadly adopted networks in the hand and object pose estimation areas to assess the feasibility of our system.

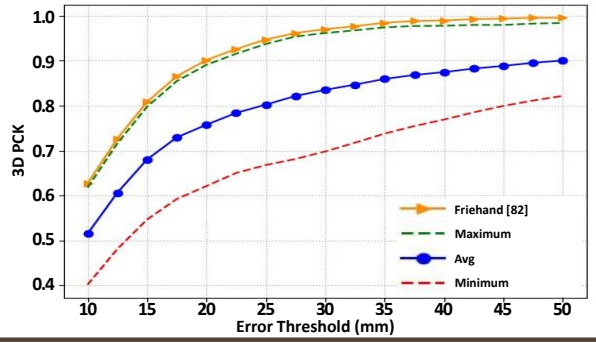
5.2.1 Procedure. We asked the users to collect the datasets for 4 hand-held objects: a soft drink bottle, a cereal box, a toy car, and a flashlight (Figure 7a), where the bottle and the cereal box were included in the Objectron dataset [1], and the other two were chosen considering the varied shapes and aspect ratios. The users were encouraged to manipulate the objects with any gestures they preferred. The session finished when the number of the collected images reached the fore-mentioned empirically set target number, which was indicated by the *progress indicator*. We recorded the total completion time for each user.

For each user and each object, we trained the two networks mentioned in Section 4 for detecting the 3D poses of the hand and object. And we adopted the same network training details. Following the general process in the 3D object pose estimation area [1, 47], we calculated Average Precision (AP) of the 3D Intersection over Union (IoU) with a threshold of 0.5, 2D pixel projection error, AP of azimuth with a threshold of 15°, and AP of elevation with a threshold of 10°. Specifically, the 3D IoU computes the intersection-of-volume of the predicted 3D bounding box and the ground truth. Given the estimated and ground truth poses of a 3D bounding box, the 2D pixel projection error measures the mean normalized distance between the projections of the keypoints of the bounding box. For the hand, we report the Mean Per Joint Position Error (MPJPE) and the Percentage of Correct Points (PCK) following [52, 61]. The PCK refers to the probability that a joint is within a distance threshold of its ground truth location [77].

Figure 7: (a) The four objects used for the dataset collection: A bottle, a cereal box, a toy car, and a flashlight. The first two were included in the Objectron [1] dataset. (b) Example *bounding contours* created by the users.

5.2.2 Result and Discussion. All the participants successfully completed the dataset collection task using ARnnotate. The average completion time to collect 2.5k images for each user and each object category was 12.37 minutes (SD = 3.25). Specifically, Figure 7b shows some *bounding contours* created by the users, while Figure 9 illustrates sample gestures performed by the users for each object. Overall, by utilizing the *bounding contour* creation methods, the users could build *bounding contours* that met the requirements for the *image recording* step. Meanwhile, the users could grab the object with different but common gestures when following the *orientation indicator* and textual suggestions.

For the object pose estimation (Figure 8a), the average precision at 0.5 3D IoU was 0.7073 and the 2D pixel projection error was 0.0569, showing comparable performance to the result of 0.7218 (3D IoU) and 0.0520 (projection error) in CenterPose [47]. As for the viewpoint estimation, the azimuth error mean was 0.8147 and the elevation error mean was 0.8681. Typically, the performance of the toy car was not as good as the other categories. This was partially due to its complicated shape and the users were not able to draw a precise bounding box. We will discuss it later. The cereal box category was also challenging because of the larger length-width ratio, which made it more difficult to follow in rotation-dominant manipulation. We also observed that users were having difficulty in aligning the cereal box's width with the bounding box's width during image recording. For the hand pose estimation, the average 3D error of MPJPE for each user was 15.10mm, showing state-of-the-art performance compared with the result of 12.32mm in [52]. For the PCK results, the blue solid line in Figure 8b shows the average evaluation result for all the users' hands. The two dashed lines stand for the two users' datasets with the best and worst performance during training. We choose a popular 3D hand dataset [82] trained on the same network as the baseline, shown as the



Dataset	Bottle	Cereal box	Toy car	Flashlight	Mean
Average precision at 0.5 3D IoU (Higher is better)					
Objectron [1]	0.8021	**0.8211**	-	-	**0.7218**
Ours	**0.8326**	0.6741	0.5319	0.7905	0.7073
Mean pixel error of 2D projection (Lower is better)					
Objectron [1]	**0.0400**	**0.0379**	-	-	**0.0520**
Ours	0.0458	0.0569	0.0816	0.0433	0.0569
Average precision at 15° azimuth error (Higher is better)					
Objectron [1]	**0.9561**	**0.9361**	-	-	**0.8398**
Ours	0.9503	0.7874	0.5832	0.9380	0.8147
Average precision at 10° elevation error (Higher is better)					
Objectron [1]	0.8881	**0.9467**	-	-	0.8560
Ours	**0.9492**	0.8213	0.7969	0.9048	**0.8681**

(b)

Figure 8: (a) The training results of the 3D object pose estimation neural network. (b) The training results of the 3D hand pose estimation neural network.

orange solid line. The hand pose estimation results show that the user-collected datasets have adequate performance compared with state-of-art datasets. Figure 9 demonstrates some examples of the hand pose and object pose test results on the user-collected datasets. Overall, the decent training results indicate that ARnnotate has the capability to support users to create high-quality custom datasets used for 3D hand and object pose estimation.

5.3 Qualitative Feedback on the System Usability

As an AR-based user interface, the Likert-type questionnaire ratings illustrate the users' subjective feedback on the system usability, which are shown in Figure 10. Regarding the *bounding contour* creation, the users agreed that the specific shape of the *bounding contour* helped them place and manipulate the physical object (Q6, AVG=4.92, SD=0.29), and they were confident to create an accurate *bounding contour* using our system (Q1, AVG=4.33, SD=0.89). *"The [bounding contour] creation is a brilliant idea, and is fun as well. Now, I know where I should move and rotate the object when I look at it. (P3)"* Yet, one user raised that *"I need more time to think about how to arrange those primitives for that toy car. I thought maybe there could be a better solution. (P11)"* We will discuss this concern in Section 6. Meanwhile, the majority of the users provided positive feedback about the hand manipulation with the virtual *bounding contour* (Q2, AVG = 4.33, SD = 0.65) and the depth occlusion provided by AR-HMD (Q5, AVG = 4.17, SD = 0.83). *"It was super cool that I could feel the depth in AR with that occlusion visual effect. I thought I followed the clip precisely. (P8)"* But, this user also commented: *"For that hand mesh, maybe a skeleton model could be better. Because using that hand mesh, sometimes I could not see the fingers in the back. (P8)."* This could be addressed by allowing users to select different visual effects of the hand model, and guiding users to walk around the in-situ animation from different perspectives before recording. One key feature of our system is to process the *interaction clips* to help users align the objects more accurately. This feature received complimentary feedback (Q7, AVG = 4.83, SD= 0.58). *"I was surprised that the turning of the animation was so fluent that I could easily follow it. (P12)"* Further, the clear UI of the system was welcomed by the users (Q8, AVG = 4.67, SD = 0.65). *"I like the idea that you put those buttons right next to the virtual objects. Really straightforward to follow. (P5)"* Last but not least, the standard SUS survey result was 90.67 out of 100 (SD=8.84), which indicated the satisfactory usability of the system.

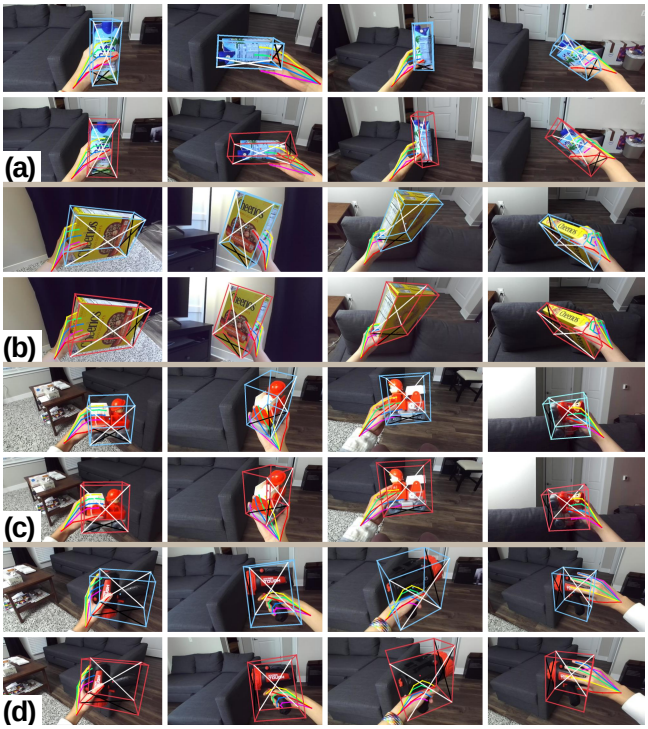


Figure 9: Example test results of the hand and object pose estimation neural networks trained using the datasets collected by the users. (a) Bottle. (b) Cereal box. (c) Toy car. (d) Flashlight. For each object category, top row shows ground truths, while bottom row contains the prediction results.

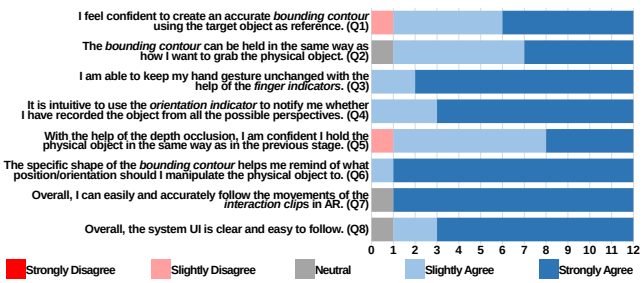


Figure 10: The results of the qualitative feedback on the system usability.

6 LIMITATION AND FUTURE WORK

6.1 Complicated Hand-Object Interaction

The hand-object interaction supported by ARnnotate focuses on the objects that can be grabbed with a fixed gesture (e.g., using a 'fist' gesture to hold the handle of a cup, or using a 'grasp' gesture to hold a milk box). And the approach of binding a *bounding contour* onto the user's hand was proved to be feasible in the user study. Yet, for some objects such as scissors and smaller objects, users may grab them in more complex ways (e.g., only using three fingers to hold scissors, or unconsciously rotating a pill bottle when using fingertips to hold the cap portion). In these scenarios, additional DoFs of the hand gesture are introduced due to the free movements of the fingers, which may reduce the accuracy of the hand pose labeling. One way to resolve this issue is to introduce physics-based hand-object contact point models [3, 30, 55]. By detecting the contact points between the fingers and the *bounding contour*, users can hold *bounding contour*s more precisely. The adoption of this model also enables the double-hand interaction that is not currently supported since most of the current hand pose estimation networks [24, 82] focus on single-hand interaction detection. Meanwhile, by scanning a physical object as 3D mesh using LiDAR-capable devices [34, 71], we can generate a meshed model of the physical object for the implementation of the contact point model. In addition, some gestures may introduce self-occlusions of the hand so that the bare-hand tracking module cannot output the correct gestures. Potential solutions can be deleting those data samples where the confident values of the hand tracking module are lower than a threshold, and warning users to avoid such gestures using the *gesture indicators*.

Further, the complexity may also be introduced due to the geometric features of the objects. During the user study, we noticed that it was challenging for the users to follow some *interaction clip*s. *"That cereal box is too thin and tall. When I manipulated the virtual one, I didn't realize it could be so hard to follow those rotational movements. (P9)"* In light of this, it would be a future research direction to study that given different types of objects (e.g., cereal boxes), what movement patterns are not recommended (e.g., rotate the *bounding contour* of a cereal box too fast). In addition, we observed that different users preferred different ways of manipulating the objects (e.g., some users were good at following translation movements). Such personal preferences can be considered together to provide personalized suggestions for recording *interaction clip*s that are more easily to be followed. In the next sub-section, we will also discuss how to deliver these suggestions to users.

6.2 Additional Supports and Suggestions for Users with Different Levels of Expertise

During the user study, the users had different levels of expertise in CV and sketching. As mentioned before, P11 was not confident about the *bounding contour* creation for the objects with complicated geometric shapes. Meanwhile, we observed that the network performance of the toy car was slightly lower than the others, which was partially attributed to the difficulty in creating an accurate *bounding contour* of the toy car. We also observed that for complex objects, different users selected different primitives to construct the *bounding contour* (Figure 7b), while some choices may not be optimal. To help novice users generate a better *bounding contour* for complex objects, we could embed a web-based search engine to fetch suggested 3D sketches given the name or sample image of the target object, and show them in AR as an additional reference. Meanwhile, the mesh scan of a physical object mentioned in the previous sub-section could serve as another reference.

Moreover, the user study results proved the feasibility of the workflow we propose to support any end-user to collect decent-quality datasets. Yet, a user, who is a researcher in graphics and CV, raised that *"I know the process of labeling datasets well, so I want the system to show me the results and let me double-check if the labels are accurate enough. (P4)"* Previous works suggest that an interactive system needs to adapt to users' personal experience [3, 15, 32]. We could address this consideration by providing an in-situ placed video editor for expert users to directly review the datasets, and manually delete the unsatisfactory data. On the other hand, for the users who do not perform well in following the *interaction clip*, the system could adaptively show additional guidance. For instance, the color of the *bounding contour* could change when it is about to move in another direction. Meanwhile, following prior works in the MR tutoring area [32, 65], in-situ virtual elements such as an arrow attached next to the *bounding contour* can be used to indicate its moving direction and speed, while the same idea can also be used in the *label recording* to indicate the suggested patterns of manipulating the *bounding contour*.

6.3 Fatigue When Moving Hands in AR

While we suggest collecting a specific number of images to guarantee satisfactory performance of the neural network, ARnnotate supports users to record an *interaction clip* with arbitrary time length. However, several users pointed out the arm fatigue problem. On one hand, it is partially because the current AR-HMD's image saving rate is only 15 fps. Yet, with the advances in hardware, we envision that images can be saved more efficiently. Furthermore, earlier works [28, 35] also address the fatigue problem and provide plausible solutions by using mid-air pointing or other ways of indirect manipulation. Unfortunately, these solutions do not fit our needs. Another solution could be gamifying the repetitive and boring dataset collection process so that users will be distracted from the tiredness [68, 69].

6.4 Expansion of the Dataset

We have shown the effectiveness of enabling end-users to collect RGB-based hand-object interaction datasets using a novel AR-based workflow. In the future, we would like to expand the system capability in several directions. We could provide hand and object segmentation labels [6, 7, 70] by projecting the *bounding contour*s and the hand mesh model detected by the AR-HMD onto the images. Specifically, fusion algorithms [34] can be used to directly scan physical objects as accurate *bounding contour*s, while calibration of different users' hands will also be considered. Meanwhile, it would be straightforward to add the depth information besides the RGB images into the datasets given that current AR-HMDs have already embedded the depth perception capability. Moreover, if users can keep accumulating their own datasets, their personal annotation preferences and the corresponding data can be used for transfer learning and dataset synthesizing [21, 46, 64].

6.5 Large-scale Dataset Collection

Currently, our system mainly focuses on the collection of user-specific datasets, and therefore, only suggests end-users collecting a specific number of data samples to train the networks. Yet, our system also has the potential to be leveraged for constructing a large-scale bench-marking dataset for the development of image-based pose estimation research in the CV area. To this end, we envision allocating our system to a large number of users and allow for collecting data with different hand skin colors in various environment backgrounds via crowd-sourcing [12, 31]. Meanwhile, deploying our system to phone-based platforms would help further expand the scale of the bench-marking dataset collection. With the advents of hardware (e.g., LiDAR) and software (ARKit [4]), AR-capable cell phones have been utilized to collect datasets [41]. We envision that bare-hand tracking would be embedded into cell phones in the near future. By mounting the phone in front of the user, ARnnotate can be used in the same way as the HMD-based system. Yet, the limited field-of-view and data storage issues may require designing additional supportive features.

7 CONCLUSION

In this paper, we presented ARnnotate, an AR interface for end-users to create a custom dataset for hand and object pose estimation. In order to address the issues of complicated hand-object occlusion and the needs for collecting custom data, we introduce a workflow that fully leverages the spatial awareness and hand-tracking capability enabled by an advanced AR-HMD. Specifically, a user can create a realistic virtual bounding box using the target physical object as spatial reference in AR. ARnnotate records dataset labels when the user manipulates the virtual bounding box, and records dataset images when the user manipulates the physical object to follow the in-situ replayed animation of the bounding box and hand model. We introduced a series of visual hints and data processing approaches to facilitate the dataset creation and increase the accuracy of the labels. In the user study, we first quantitatively proved that, with ARnnotate, users were able to accurately create image-label pairs in terms of position and rotation accuracy. Then, we trained deep neural networks using the datasets collected by the users. The satisfactory results compared with the existing benchmark dataset works illustrated the feasibility of our system. Meanwhile, the complimentary subjective feedback from the users further proved the usability of the interface of ARnnotate. To sum up, we believe this work opens up a novel perspective of utilizing AR as assistance to resolve problems in 3D-domain dataset collection. And we envision it fosters more HCI applications leveraging hand-object pose estimation in the future.

ACKNOWLEDGMENTS

We thank the reviewers for their invaluable feedback. This work is partially supported by the NSF under the Future of Work at the Human Technology Frontier (FW-HTF) 1839971. We also acknowledge the Feddersen Distinguished Professorship Funds. Any opinions, findings, and conclusions are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. 2021. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7822–7831.

[2] Hugo Alvarez, Iker Aguinaga, and Diego Borro. 2011. Providing guidance for maintenance operations using automatic markerless augmented reality system. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 181–190.

[3] Dafni Antotsiou, Guillermo Garcia-Hernando, and Tae-Kyun Kim. 2018. Task-oriented hand motion retargeting for dexterous manipulation imitation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

[4] Apple Inc. 2022. Apple ARKit. https://developer.apple.com/augmented-reality/arkit/.

[5] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15.

[6] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. 2020. ContactPose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision*. Springer, 361–378.

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. 2018. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1209–1218.

[8] Yuanzhi Cao, Xun Qian, Tianyi Wang, Rachel Lee, Ke Huo, and Karthik Ramani. 2020. An exploratory study of augmented reality presence for tutoring machine tasks. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–13.

[9] Yuanzhi Cao, Tianyi Wang, Xun Qian, Pawan S Rao, Manav Wadhawan, Ke Huo, and Karthik Ramani. 2019. GhostAR: A time-space editor for embodied authoring of human-robot collaborative task with augmented reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 521–534.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7291–7299.

[11] Fu Chang, Chun-Jen Chen, and Chi-Jen Lu. 2004. A linear-time component-labeling algorithm using contour tracing technique. *computer vision and image understanding* 93, 2 (2004), 206–220.

[12] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2334–2346.

[13] Yun Suk Chang, Benjamin Nuernberger, Bo Luan, and Tobias Höllerer. 2017. Evaluating gesture-based augmented reality annotation. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 182–185.

[14] Subramanian Chidambaram, Hank Huang, Fengming He, Xun Qian, Ana M Villanueva, Thomas S Redick, Wolfgang Stuerzlinger, and Karthik Ramani. 2021. Processar: An augmented reality-based tool to create in-situ procedural 2d/3d ar instructions. In *Designing Interactive Systems Conference 2021*. 234–249.

[15] Michael P Domjan. 2014. *The principles of learning and behavior*. Cengage Learning.

[16] Scott Frees, G Drew Kessler, and Edwin Kay. 2007. PRISM interaction for enhancing control in immersive virtual environments. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14, 1 (2007), 2–es.

[17] Markus Funk, Andreas Bächler, Liane Bächler, Thomas Kosch, Thomas Heidenreich, and Albrecht Schmidt. 2017. Working with augmented reality? A long-term analysis of in-situ instructions at the assembly workplace. In *Proceedings of the 10th international conference on pervasive technologies related to assistive environments*. 222–229.

[18] Lei Gao, Huidong Bai, Gun Lee, and Mark Billinghurst. 2016. An oriented point-cloud view for MR remote collaboration. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. 1–4.

[19] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. 2018. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 409–419.

[20] Danilo Gasques, Janet G Johnson, Tommy Sharkey, and Nadir Weibel. 2019. What you sketch is what you get: Quick and easy augmented reality prototyping with pintar. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[21] Georgios Georgakis, Arsalan Mousavian, Alexander C Berg, and Jana Kosecka. 2017. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836* (2017).

[22] Duncan Goudie and Aphrodite Galata. 2017. 3D hand-object pose estimation from depth with convolutional neural networks. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 406–413.

[23] Anhong Guo, Xiang'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P Bigham. 2016. Vizlens: A robust and interactive

screen reader for interfaces in the real world. In *Proceedings of the 29th annual symposium on user interface software and technology*. 651–664.
[24] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. 2020. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3196–3206.
[25] Ping-Hsuan Han, Kuan-Wen Chen, Chen-Hsin Hsieh, Yu-Jie Huang, and Yi-Ping Hung. 2016. Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective. In *Proceedings of the 7th Augmented Human International Conference 2016*. 1–4.
[26] Ping-Hsuan Han, Jia-Wei Lin, Chen-Hsin Hsieh, Jhih-Hong Hsu, and Yi-Ping Hung. 2018. tARget: limbs movement guidance for learning physical activities with a video see-through head-mounted display. In *ACM SIGGRAPH 2018 Posters*. 1–2.
[27] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11807–11816.
[28] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1063–1072.
[29] Thuong N Hoang, Martin Reinoso, Frank Vetere, and Egemen Tanin. 2016. Onebody: remote posture guidance system using first person view in virtual environment. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. 1–10.
[30] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. 2018. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 175–182.
[31] Jonggi Hong, Kyungjun Lee, June Xu, and Hernisa Kacorri. 2020. Crowdsourcing the perception of machine teaching. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–14.
[32] Gaoping Huang, Xun Qian, Tianyi Wang, Fagun Patel, Maitreya Sreeram, Yuanzhi Cao, Karthik Ramani, and Alexander J Quinn. 2021. Adaptutar: An adaptive tutoring system for machine tasks in augmented reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
[33] Ke Huo and Karthik Ramani. 2017. Window-shaping: 3d design ideation by creating on, borrowing from, and looking at the physical world. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 37–45.
[34] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 559–568.
[35] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3328–3339.
[36] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2015. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*. 3334–3342.
[37] Thomas Kosch and Albrecht Schmidt. 2020. Enabling Tangible Interaction through Detection and Augmentation of Everyday Objects. *arXiv preprint arXiv:2012.10904* (2020).
[38] Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketom, Christian Sandor, and Hirokazu Kato. 2017. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics* 24, 2 (2017), 1038–1048.
[39] Kin Chung Kwan and Hongbo Fu. 2019. Mobi3dsketch: 3D sketching in mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.
[40] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. 2021. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10138–10148.
[41] Michael Laielli, James Smith, Giscard Biamby, Trevor Darrell, and Bjoern Hartmann. 2019. Labelar: a spatial guidance interface for fast computer vision image collection. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 987–998.
[42] Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. Em-sense: Touch recognition of uninstrumented, electrical and electromechanical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 157–166.
[43] Jangwon Lee and Michael S Ryoo. 2017. Learning robot activities from first-person human videos using convolutional future regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1–2.
[44] Kyungjun Lee and Hernisa Kacorri. 2019. Hands holding clues for object recognition in teachable machines. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[45] Hanchuan Li, Can Ye, and Alanson P Sample. 2015. IDSense: A human object interaction detection system based on passive UHF RFID. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2555–2564.
[46] Kailin Li, Lixin Yang, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. 2021. ArtiBoost: Boosting Articulated 3D Hand-Object Pose Estimation via Online Exploration and Synthesis. *arXiv preprint arXiv:2109.05488* (2021).
[47] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. 2021. Single-stage Keypoint-based Category-level Object Pose Estimation from an RGB Image. *arXiv preprint arXiv:2109.06161* (2021).
[48] Yao Lu and Walterio W Mayol-Cuevas. 2021. The Object at Hand: Automated Editing for Mixed Reality Video Guidance from Hand-Object Interactions. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 90–98.
[49] Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2010. The effect of dof separation in 3d manipulation tasks with multi-touch displays. In *Proceedings of the 17th acm symposium on virtual reality software and technology*. 111–118.
[50] Daniel Mendes, Filipe Relvas, Alfredo Ferreira, and Joaquim Jorge. 2016. The benefits of dof separation in mid-air 3d object manipulation. In *Proceedings of the 22nd ACM conference on virtual reality software and technology*. 261–268.
[51] Microsoft HoloLens 2022. Microsoft HoloLens | Mixed Reality Technology for Business. https://www.microsoft.com/en-us/hololens.
[52] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*. Springer, 548–564.
[53] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. 2019. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8688–8697.
[54] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. 2020. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 163–172.
[55] Kiran Nasim and Young J Kim. 2018. Physics-based assistive grasping for robust object manipulation in virtual reality. *Computer Animation and Virtual Worlds* 29, 3-4 (2018), e1820.
[56] Oculus Quest 2 2022. Oculus Quest 2: Our Most Advanced New All-in-One VR Headset | Oculus. https://www.oculus.com/quest-2.
[57] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. 2018. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 436–445.
[58] Xun Qian, Fengming He, Xiyun Hu, Tianyi Wang, Ananya Ipsita, and Karthik Ramani. 2022. ScalAR: Authoring Semantically Adaptive Augmented Reality Experiences in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*. 1–18.
[59] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
[60] Richard Sennett. 2008. *The craftsman*. Yale University Press.
[61] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1145–1153.
[62] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 567–576.
[63] Stereolabs. 2022. ZED Mini Stereo Camera - Stereolabs. Retrieved March 1, 2022 from https://www.stereolabs.com/zed-mini/.
[64] Jonti Talukdar, Sanchit Gupta, PS Rajpura, and Ravi S Hegde. 2018. Transfer learning for object detection using state-of-the-art deep neural networks. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 78–83.
[65] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Björn Hartmann. 2019. TutoriVR: A video-based tutorial system for design applications in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
[66] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. 2009. Influence of degrees of freedom's manipulation on performances during orientation tasks in virtual reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. 51–58.
[67] Ana Villanueva, Zhengzhe Zhu, Ziyi Liu, Feiyang Wang, Subramanian Chidambaram, and Karthik Ramani. 2022. ColabAR: A Toolkit for Remote Collaboration in Tangible Augmented Reality Laboratories. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–22.
[68] Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 319–326.

[69] Luis Von Ahn and Laura Dabbish. 2008. Designing games with a purpose. *Commun. ACM* 51, 8 (2008), 58–67.

[70] Fan Wang and Kris Hauser. 2019. In-hand object scanning via rgb-d video segmentation. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 3296–3302.

[71] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 552–567.

[72] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Ke Huo, Yuanzhi Cao, and Karthik Ramani. 2020. CAPturAR: An augmented reality tool for authoring human-involved context-aware applications. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 328–341.

[73] Tianyi Wang, Xun Qian, Fengming He, and Karthik Ramani. 2021. LightPaintAR: Assist Light Painting Photography with Augmented Reality. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–6.

[74] Yeping Wang, Gopika Ajaykumar, and Chien-Ming Huang. 2020. See what i see: Enabling user-centric robotic assistance using first-person demonstrations. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 639–648.

[75] Curtis Wilkes and Doug A Bowman. 2008. Advantages of velocity-based scaling for distant 3D manipulation. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. 23–29.

[76] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2017. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199* (2017).

[77] Yi Yang and Deva Ramanan. 2012. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence* 35, 12 (2012), 2878–2890.

[78] Yang Zhang, Yasha Iravantchi, Haojian Jin, Swarun Kumar, and Chris Harrison. 2019. Sozu: Self-powered radio tags for building-scale activity sensing. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 973–985.

[79] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-cost touch sensing using electric field tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1–14.

[80] Qian Zhou, Sarah Sykes, Sidney Fels, and Kenrick Kin. 2020. Gripmarks: Using Hand Grips to Transform In-Hand Objects into Mixed Reality Input. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.

[81] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*. 4903–4911.

[82] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. 2019. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 813–822.