ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Subramanian Chidambaram[2]
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: subbuc@amazon.com

Rahul Jain[1,2]
School of Electrical and Computer Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: jain348@purdue.edu

Sai Swarup Reddy
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: saiswarupreddy4399@gmail.com

Asim Unmesh
School of Electrical and Computer Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: aunmesh@purdue.edu

Karthik Ramani
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907
e-mail: ramani@purdue.edu

AnnotateXR: An Extended Reality Workflow for Automating Data Annotation to Support Computer Vision Applications

Computer vision (CV) algorithms require large annotated datasets that are often labor-intensive and expensive to create. We propose AnnotateXR, an extended reality (XR) workflow to collect various high-fidelity data and auto-annotate it in a single demonstration. AnnotateXR allows users to align virtual models over physical objects, tracked with six degrees-of-freedom (6DOF) sensors. AnnotateXR utilizes a hand tracking capable XR head-mounted display coupled with 6DOF information and collision detection to enable algorithmic segmentation of different actions in videos through its digital twin. The virtual–physical mapping provides a tight bounding volume to generate semantic segmentation masks for the captured image data. Alongside supporting object and action segmentation, we also support other dimensions of annotation required by modern CV, such as human–object, object–object, and rich 3D recordings, all with a single demonstration. Our user study shows AnnotateXR produced over 112,000 annotated data points in 67 min.
[DOI: 10.1115/1.4066180]

Keywords: dataset, machine learning, computer vision, annotation, extended reality, human–computer interfaces/interactions, virtual and augmented reality environments

1 Introduction

The field of computer vision (CV) has made significant progress in the last decade with the help of advances in machine learning (ML) algorithms. CV has demonstrated a large variety of practical applications in many fields such as autonomous driving [1,2], biomedical imaging [3], robotics [4], and point cloud mapping [5,6]. However, most current state-of-the-art ML algorithms rely heavily on high-quality and high-quantity annotated data sets [7–11] for training and test sampling. Hence, researchers in the CV community are constantly producing annotated data sets tailored to specific problems and applications.

These standardized data sets offered by the CV community have facilitated the creation, validation, and improvement of algorithms. Currently, these data sets are annotated post hoc by manual annotators (e.g., Mturkers) with tools such as Mechanical Turk [12], Sagemaker Ground Truth [13], Supervisely [14], and Anolytics [15]. Data set annotations are often time, money, and labor-intensive endeavors [16]. This bottleneck inhibits users from quickly and efficiently creating customized data sets for end-user applications [17]. Apart from labor intensity, modern CV algorithms target applications requiring multiple types of annotations within the same data. For example, works in action segmentation such as Action Genome [18] and Home Action Genome [11] have shown that additional annotation information regarding human–object (H–O) interaction alongside action segmentation information has improved performance. However, supporting the need for multiple annotations currently compounds labor intensity limitations, prevents scalability, and limits research.

To address the need for high-quality annotations while reducing dependency on manual labor, works such as Playing for Data [19], O2O [20], and Tremblay et al. [21] propose using synthetic environments to generate data sets. Synthetic data sets are becoming accessible due to advances in rendering pipelines and generative adversarial models [22]. While being more efficient for data set creation than manual annotation, purely synthetic data sets are limited in their utility as the generated data has no grounding in the real world, such as lack of RGB frames [23]. Synthetic data sets inspired us to look at virtual environments for generating large quantities of annotated data. The shortcomings of synthetic data sets motivated us to ground the virtual environment to a real-physical environment. Hence, we propose generating a virtual equivalent of the physical world and updating the virtual based on the changes in the physical.

We present AnnotateXR, an extended reality (XR) application capable of simultaneous collection and auto-annotation of data to support several different applications with a single demonstration. Applications such as object detection [24], semantic segmentation

[1]Corresponding author.
[2]These authors contributed equally to this work.
Manuscript received December 14, 2023; final manuscript received July 27, 2024; published online September 9, 2024. Assoc. Editor: Shana Smith.

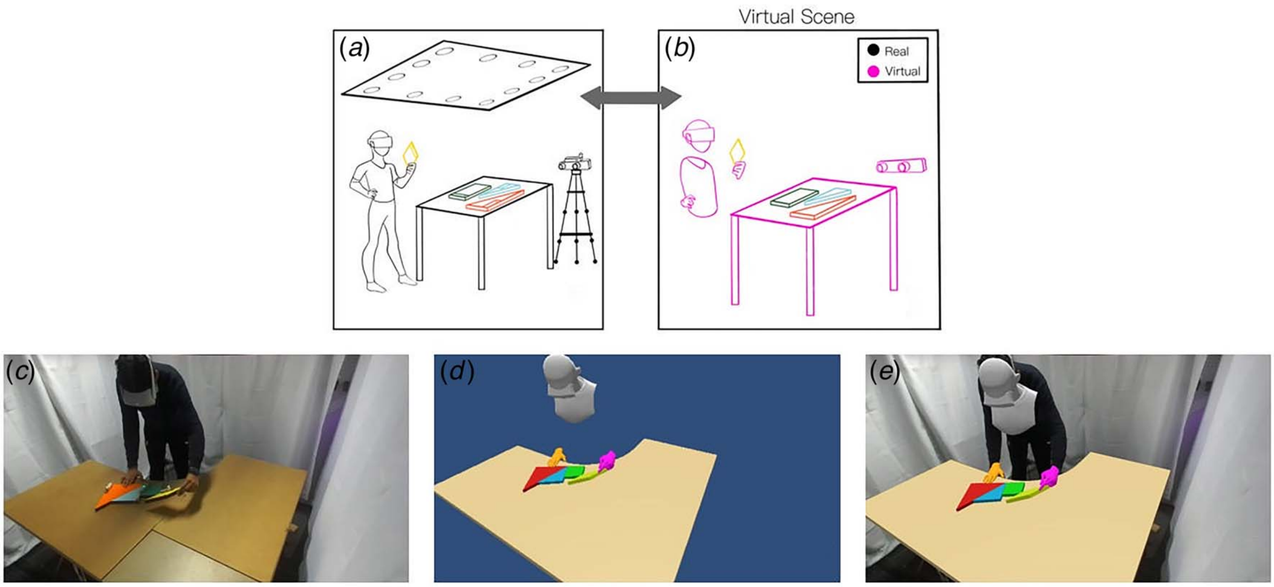

Fig. 1 Overview of the AnnotateXR data collection workflow: (*a*) a user performing a task with actively tracked objects within a tracking area in front of an actively tracked RGB camera, (*b*) a virtual digital twin of the real-world interactions of the user and objects, (c) a raw 2D image of the user performing the task, (*d*) a virtual 3D replica of the user's action, and (*e*) a one-to-one overlay of the virtual and real images utilized to generate segmentation masks

[25], video action segmentation [26], 6DOF predictions [27], H–O interaction [28], object–object (O–O) interaction [20,29], and rich 3D scene recording [30] are supported by AnnotateXR.

AnnotateXR explores leveraging the strength of XR to record and annotate data. We achieve this by capturing a digital twin of the real-world action. A digital twin is defined as "an executable virtual model of a physical thing or a system." [31,32] An external six degrees-of-freedom (6DOF) sensor (Antilatency [33]) is attached to every physical object and tracked. At the same time, the user interaction is captured via head-mounted display (HMD) Oculus Quest 2 [34].

A virtual replica (widely available in the form of computer-aided design (CAD) models and 3D assets [35–38]) of the tracked object is aligned by the user over its physical equivalence to record the digital twin. AnnotateXR empowers the users to perform this alignment without requiring sophisticated calibration techniques. This virtual–physical alignment, in turn, also provides passive haptics for the user while performing the task and working in XR. Finally, we capture the RGB data with physical cameras. We also track the position and orientation of these physical cameras within a tracking volume. We then build a corresponding virtual capture of the objects. So, for every RGB frame captured with a physical camera, a virtual frame of the virtual world from the exact location of the physical camera is captured and used to annotate and label the data set by mapping the virtual over the physical (refer Fig. 1). This approach avoids human intervention for annotation and automates the process, thus ensuring quality while reducing cost and improving speed.

To test the strength of our system and the quality of annotation generated by our workflow, we performed a preliminary user study on 12 users and compared AnnotateXR's annotations with manually user-generated annotations. Our approach enabled even novice users to generate a large quantity (over 112,000) of multiple data annotations. Furthermore, in a post-study interview, all users preferred using AnnotateXR for large-scale data collection. The following are our contributions to the current work:

- We propose an extended reality application capable of recording physical activity and creating a virtual equivalent in parallel for capturing and annotating data to support the growing needs of modern computer vision algorithms.
  - An auto-labeling protocol capable of handling dynamic moving objects utilizes the 3D virtual model aligned with the physical object to obtain object labels and semantic segmentation masks for the corresponding RGB image frames in a video.
  - Utilizing H–O/O–O interaction information obtained via mesh collision detection in the digital twin to produce action segmentation data for action recognition.
- A user study to evaluate the difference in performance and quality of annotation between current state-of-the-art methods [14,39] and our approach. Our study shows that AnnotateXR can produce a large quantity (112,737 annotated data points in 66.55 min; total for 12 users) with statistically insignificant differences in annotation quality compared to manual annotations.

2 Related Work

Computer vision algorithms require a sufficiently large amount of data with variations to ensure coverage [40]. Furthermore, having enough data is crucial for the generalization capabilities of machine learning systems [41]. In the past decade, the variety of problems that CV has tried to solve has grown tremendously. Problems ranging from object detection [24,42,43] to segmentation [25] are being explored. In video analytics [15], action understanding tasks [44], detecting human–object interaction [18], and object–object interaction [20,29] are actively researched. To support such a large variety of problems, an equally well-annotated data set is required. Most current approaches try to provide specialized and problem-specific solutions for creating data sets [7,8]. Since there is a rising trend of multi-modal data sets with annotations for various problems in computer vision [11,18], there is also a need to support tools capable of creating such diverse annotations.

Researchers have begun expanding previously available large data sets to support additional annotations. For example, MS COCO [45] started off as a purely object label data set but has now expanded to support semantic segmentation [46], scene segmentation [47], human pose [48], image captions [49], and task detection [50], enabling the data set to support a larger domain of CV problems. Hence, problem-specific annotations are becoming outdated. To support these current trends, we have pursued a

Table 1 Positioning AnnotateXR with respect to prior related work on data annotation supports different annotation modalities for CV

	Image	Video	3D mesh	Object detection	Object segmentation	H–O	O–O	6DOF	Hand pose	Human pose
AnnotateXR (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	
LabelAR [53]	✓			✓						
LineMod [54]	✓			✓	✓			✓		
6DOF [55]	✓	✓		✓				✓		
O2O [20]			✓	✓	✓		✓			
Home Action Genome [11]		✓		✓		✓				
Action Genome [18]		✓		✓		✓				
H2O [28]	✓	✓		✓		✓		✓	✓	
GRAB [56]			✓					✓	✓	✓
3DPW [57]	✓	✓	✓							✓
Interacting Objects [29]		✓		✓		✓	✓			

Note: The categories can be grouped into the first three columns: "Image," "Video," and "3D Mesh," representing input modalities, while all other categories in the data set are various applications of the data.

more generalized strategy by allowing humans to collect data within an XR environment and creating a virtual–physical equivalence of human and object interactions to auto-generate multiple types of annotations.

2.1 Image 2D Annotation

2.1.1 Manual Annotation. The demand for high-quality annotated data sets in machine learning has enabled the development of several commercial tools. Supervisely [14], AIMultiple [51], Mechanical Turk [12], Sagemaker Ground Truth [13], and Analytics [15] are all web-based User Interface (UI) tools supporting crowdsourced and manual annotation of data. However, this approach is labor-intensive [52] and expensive [16] to support modern CV's need for data sets with multiple annotations. Thus, with AnnotateXR, we provide a workflow to automate the process while simultaneously supporting multiple annotation capabilities (Table 1).

2.1.2 Semi-Automatic Annotation. Since manual annotation is expensive and time-consuming, especially for semantic segmentation annotations [52], researchers have focused on developing approaches to aid the human annotator. For example, works such as Beat the MTurkers [58] and Xie et al. [59] use available 3D models and human-generated 3D bounding boxes to align and produce the segmentation masks. Other works, such as Castrejon et al. [60] and Acuna et al. [61], model the boundary of an object using recurrent neural networks to aid humans with annotating the images with object boundaries. Unlike these past works, AnnotateXR does not rely on human input for every frame, instead requiring virtual–physical alignment only at the beginning.

2.1.3 Mixed Reality Annotation. Works such as LabelAR [53] and Objectron [62] propose using spatial tracking technology, such as phone-based augemented reality (AR), to draw a bounding area/volume through which objects are tracked and annotated. Recent works such as ARnnotate [63] and Immersive-Labeler [64] have explored data annotation with immersive reality. ARnnotate uses an AR headset, such as Hololens 2, to annotate 3D Hand-Object Interaction Pose Estimation, and Immersive-Labeler uses a virtual reality (VR) headset to annotate 3D point clouds. Another recent work by Zhou and Yatani [65] explores the concept of real-time annotation with deictic gestures to segment objects of interest. While these works are interesting, they are domain-specific; for example, LabelAR is for 2D object detection labeling, while Objectron provides a data set for 3D object detection. These works are also limited to static objects. However, AnnotateXR tries to provide a generalized solution for a larger domain and can handle dynamic objects moving through 3D space.

2.2 Video Annotation

2.2.1 Action Understanding. Well-known data sets such as Epic Kitchens [66], Charades [67], and ActivityNet [68] provide annotated data on action segmentation for household activities. Works such as AVA [69], COIN [70], and Kinetics [71] provide

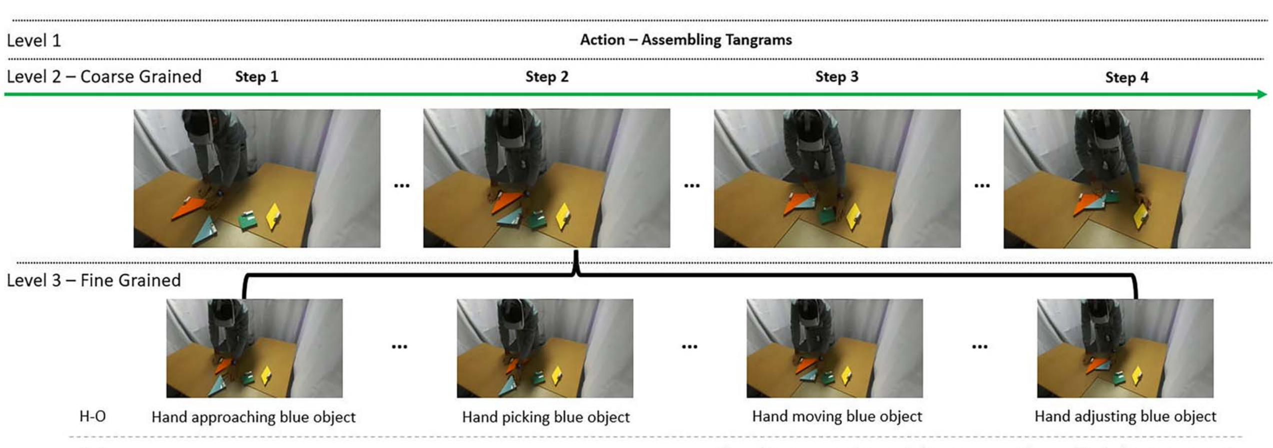

Fig. 2 There are three levels of hierarchy in videos (action, step, and interaction) [18]: level 1 is the larger task action (e.g., assembling tangrams), level 2 is coarse-grained (e.g., picking up and positioning the triangle), and level 3 is fine-grained, which involves H–O and O–O interactions (e.g., hand approaching object). The coarse-grained layer involves multiple sequential fine-grained interactions that constitute a step in level 2. Sequential combinations of level 2 steps constitute a level 1 action.

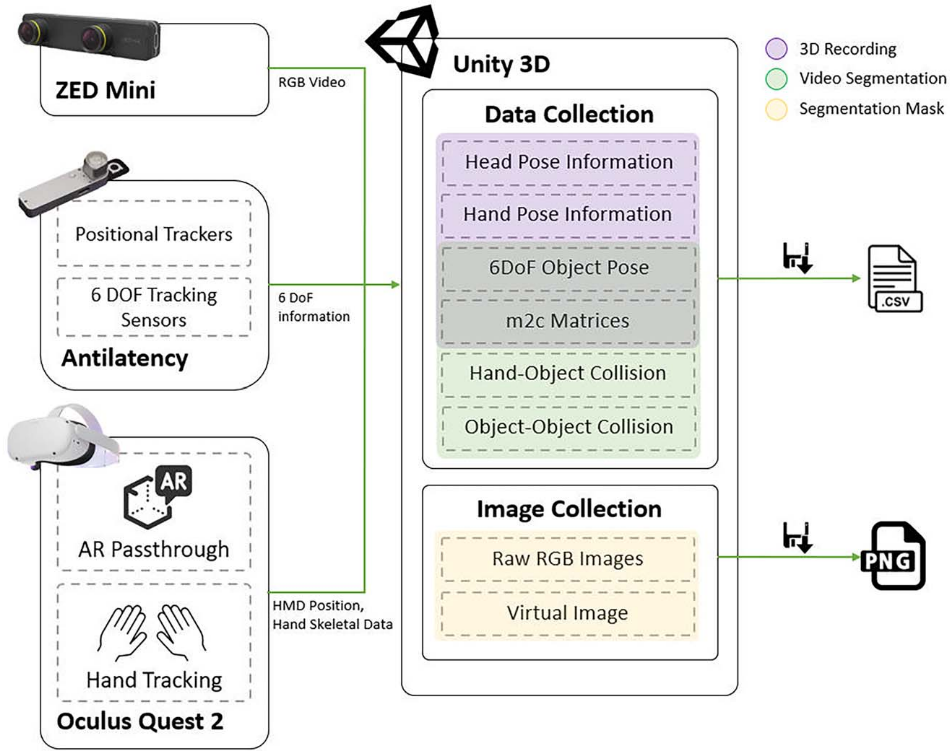

Fig. 3 System architecture: overview of the data flow from the different hardware used for various sub-systems and data collection

annotated data from open sources such as movies and YouTube. However, these past works rely on manual annotation to classify each video frame into a specific action class category.

Unlike the coarse-grained annotations (refer Fig. 2) offered by the works mentioned above, a recent trend in action segmentation has been to explore the concept of segmenting fine-grained actions. Works such as Something Else [72], FineGym [73], and FineAction [44] differentiate phases of complex real-world actions such as gymnastics and soccer video clips based on how objects within each frame relate to each other (H–O and O–O relations). However, this strategy is far from satisfactory due to the need for detailed annotations.

Recent work such as Action Genome [18] and Home Action Genome [11] allows for grouping and annotating five frames together instead of annotating each frame individually. This reduces the workload on the annotator while trying to annotate fine-grained interactions, such as between human and object or object and object. However, complete reliance on manual annotations, as pursued by these past studies, is not a viable solution for developing generic large-scale customized annotated data sets due to resource intensity. Hence, in AnnotateXR, we have explored the concept of recording a digital twin from observed human and object physical actions. The digital twin provides necessary cues of H–O and O–O interactions via collision detection, which are used to automate both fine-grained and sparse action segmentation for every frame with less effort, providing a scalable alternative to existing approaches.

2.2.2 Semantic Segmentation in Videos. Works such as DAVIS [74] and CamVID [75] provide segmentation mask annotation of objects in a video. However, the masks were obtained by manual annotation over every frame. Other works, such as Vijayanarasimhan and Grauman [76], allow the user to annotate the first frame in a video and try to propagate the annotation over subsequent frames. However, due to a lack of confidence in tracking, these methods are viable only for short video clips before the tracking propagation loses accuracy. Recently, IKEA ASM [77] has explored annotating segmentation masks, human pose, and object pose only on keyframes within a video to reduce human effort. However, this approach still required keyframes to be "identified" and annotated. AnnotateXR, however, relies on virtual–physical model mapping and physical object tracking to generate a dynamic segmentation mask capable of annotating every frame in a video with minimal effort.

2.3 Data Collection Via Sensor Capture. Several past works have explored the idea of gathering 3D object information with sensors [56,55,54–56]. GRAB [56] provides rich 3D pose information of the human body and object captured with a body-tracking suit and several embedded markers. Work such as Garon et al. [55] has explored the concept of using smaller markers and removing them post-collection by pixel masking. Work such as Ahmad et al. [78] generates automatic datasets from CAD models. Other well-known works such as Linemod [54] utilize depth cameras such as Kinect [79] and available CAD models for mapping and pose estimation dataset generation.

However, these past works only provide annotations relevant for 3D tasks and 3D pose estimation. These data sets are not well suited for synergistic research, such as incorporating object pose information for action recognition, due to the lack of corresponding RGB frame information. This limitation of these data sets is partly due to the past trend in computer vision to focus on solving subproblems. AnnotateXR overcomes this limitation by enabling multi-modal annotations through generating a digital twin of the real world, thus providing ground truth RGB information alongside other 3D information such as 6DOF, hand pose, and head pose.

3 AnnotateXR Workflow

The main idea of AnnotateXR is to demonstrate a holistic design of workflow to generate annotations for various computer vision tasks. We create a 3D spatio-temporal digital twin of real-world interaction with a single demonstration. We provide the user a tool for generating detailed annotations containing object pose, object segmentation mask, action segmentation, H–O, O–O, head/hand pose, and the 3D digital twin with ease (refer Fig. 3).

3.1 Architecture and Hardware. AnnotateXR is an XR-based environment that generates a virtual replica of the real world by actively tracking objects of interest, head position, and hand pose information. This environment was deployed on a PC (AMD Ryzen 7 5800X eight-core processor 3.80 GHz CPU, 32 GB RAM, NVIDIA RTX 2080TI GPU) using an Oculus Quest 2 VR head-mounted display connected via an Oculus Link cable [34]. The application was developed in Unity 3D (2019.4.33f) with the Oculus SDK (used to visualize avatar, hands, and AR pass-through). For 6DOF tool and object tracking, we use Antilatency's development kit [33] (refer Fig. 4) that allows a $10\,\text{ft} \times 10\,\text{ft} \times 10\,\text{ft}$ ($3.048\,\text{m} \times 3.048\,\text{m} \times 3.048\,\text{m}$) tracking area. The tracking area is a ceiling-based $7\,\text{ft} \times 7\,\text{ft} \times 7\,\text{ft}$ ($2.1334\,\text{m} \times 2.1334\,\text{m} \times 2.1334\,\text{m}$) aluminum structure constructed using 80/20 Quick Frame. The tracking modules are comprised of Antilatency's "Alt Tags" and "Alt Trackers," with a footprint of $18\,\text{mm} \times 66\,\text{mm}$. The sensor wirelessly transmits to Unity3D via Antilatency's "HMD Radio Sockets' (refer Fig. 4). The tracking area contains 12 tracking markers (on ceiling) that are used as reference points by the tracking modules to determine their spatial positions. At the same time, orientation is obtained by an in-built inertial measurement unit. A comparison was conducted between Optitrack V120 Duo [80] and Antilatency to determine the best option for real-time object tracking. Antilatency was particularly chosen for its ease of use and reduced setup time. Antilatency requires just one sensor per object for reliable tracking (error rate less than 2 mm [33]), whereas Optitrack requires at least three reflective markers (more required for an increase in tracking quality) attached in a unique pattern for each object. However, the system was designed to use any adequate real-time tracking solution.

An external RGB camera is also utilized to capture a video representation of the user performing the task. AnnotateXR uses a ZED mini camera [81] as the RGB camera due to its integration with Unity via the ZED-Unity plugin as well as the ease of access to accurate camera intrinsic parameters. We do not use the depth information offered by ZED for our capture. This camera is modified with an Antilatency HMD radio socket to track its position and orientation actively.

3.2 Virtual–Physical Alignment. Similar to works such as Refs. [82–85,58], AnnotateXR assumes the availability of 3D virtual models to align with the physical models. This is a reasonable assumption due to the availability of large CAD repositories: GrabCAD [37], TraceParts [38], McMasterCarr [36] and reliable 3D scanning tools such as: Qlone [86], Cognex [87], and display.-land [88].

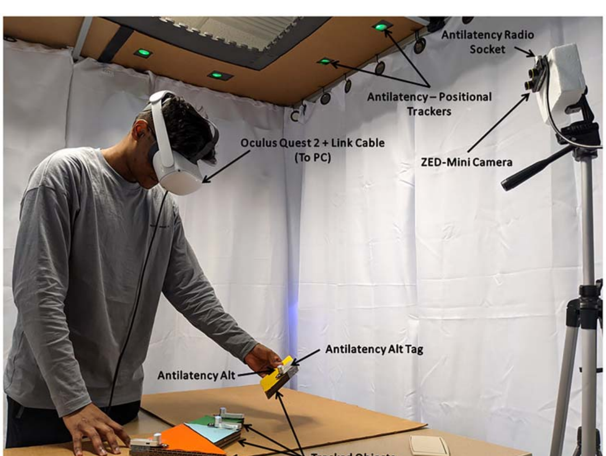

Fig. 4 Hardware setup for AnnotateXR implementation

To begin, virtual replicas (CAD) of the objects are made available in the virtual space with an XR-UI. Objects within the workspace can be categorized into either static environmental objects (such as workbenches, mounts, clamps, etc.) or dynamic objects (such as hand tools). Initially, all dynamic objects need to be tagged with an Antilatency tracking module and aligned with the virtual models for the calibration. This allows initial alignment of the virtual and physical objects (refer Fig. 5(*a*)). To achieve this, we use passthrough functionality of VR headset that lets user see the real environment as well as the virtual objects allowing them align the physical and virtual objects. Then user pinch (gesture) the calibrate button to confirm (shown in Fig. 5(*a*)), which allows for the virtual models to be aligned with the real objects during the data collection. This virtual–physical alignment was previously proposed in the work EditAR [84] to create a digital twin of real-world actions for extended reality content generation. The position and orientation of the static objects can be fixed by only initially tagging and aligning with the corresponding virtual models (i.e., the static objects need not be continuously tracked throughout the process).

3.3 Data Collection. AnnotateXR enables users (even novices) to capture and auto-annotate data with just a task demonstration. Moreover, it reduces the amount of training required for users to simultaneously generate 6DOF object tracking, object recognition, human–object, and object–object interaction annotations.

3.3.1 6DOF Data. The compiled data set contains 6DOF information for all objects of interest, head position, and hand pose information. Along with the 6DOF information, an RGB image frame of the events is captured from an external camera and stored. The 6DOF information is comprised of position vectors, rotation quaternions represented in the global coordinate space. Hand pose information, position, and rotation for each individual joints are stored. AnnotateXR also provides 4×4 model to camera (m2c) matrices that represent the position and orientation of each object transformed into the camera coordinate space. At every frame, AnnotateXR stores all the aforementioned information in a comma-separated values (CSV) file along with associated timestamps, frame numbers, and image file paths.

3.3.2 Segmentation Mask Generation. In addition to capturing the real task via an RGB camera, AnnotateXR simultaneously generates instance segmentation masks with unique colors for each object of interest. This process is conducted within a virtual 3D space, where images are captured from a virtual camera that mirrors the real camera's orientation and position, allowing for one-to-one equivalence between the generated segmentation masks and the real images. To assign unique colors to each object, unique materials are attached to each object of interest when importing the virtual model. In addition to storing the images, our system also stores the RGB color values of the materials, enabling automatic segmentation of objects without requiring any additional human intervention. This allows for efficient annotation and segmentation of thousands of image frames.

3.3.3 Human–Object and Object–Object Interactions. Human–object and object–object interactions are important [18] for the division of the task into relevant steps. In the case of spatial and sequential tasks this can be classified by determining objects that are interacting with each other and objects that the hands of the user are interacting with. AnnotateXR generates a virtual replica of the task and relies on unity's physics engine to keep track of interactions between objects and hands in a virtual replica of a spatial and sequential task, generating collision [89]

(a)

(b)

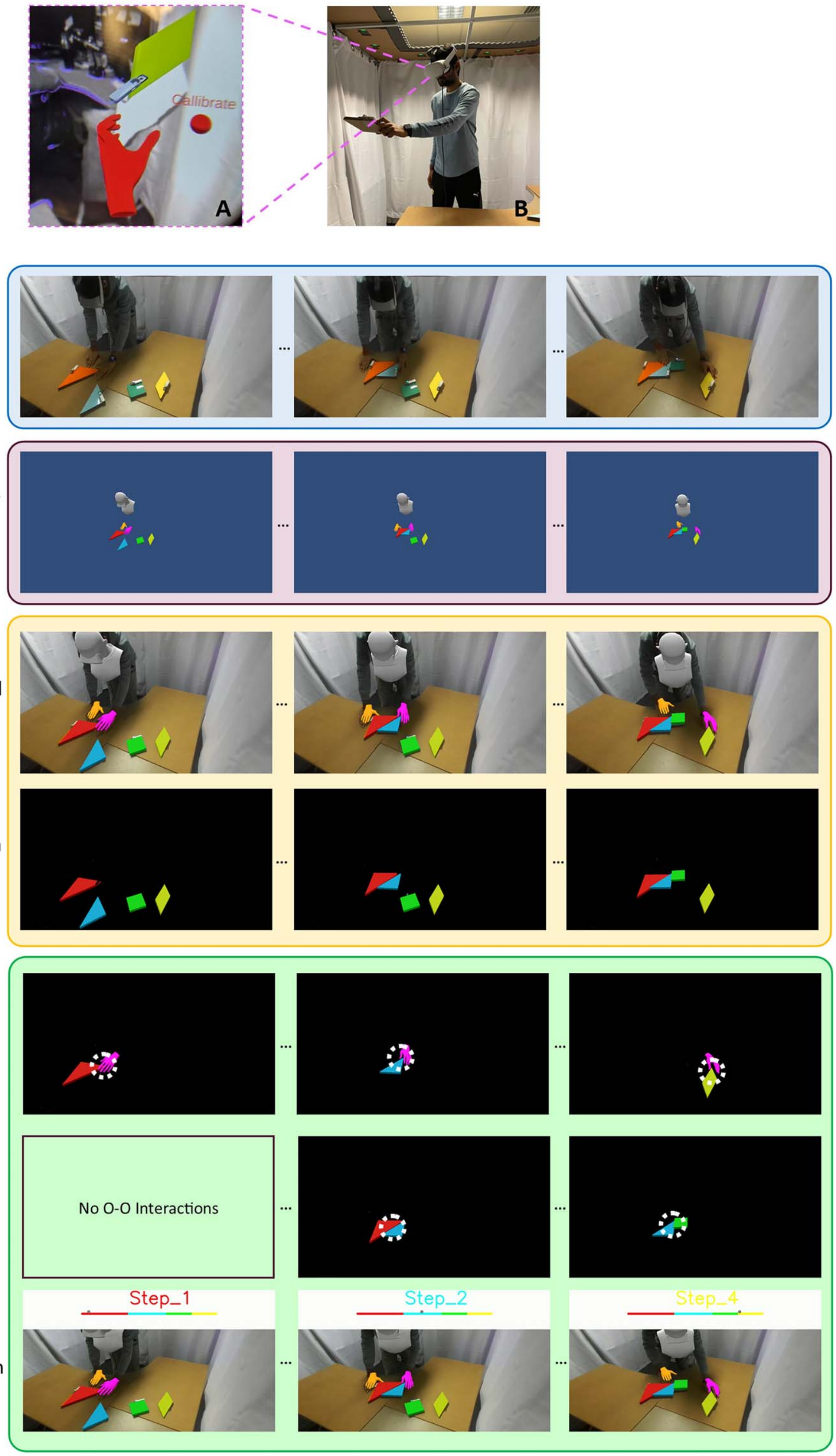

Fig. 5 An AnnotateXR workflow for data generation for assembling a four-piece tangrams puzzle. In (*a*) and (*b*), the user utilizes an AR passthrough application to align the physical model with a virtual replica. The user then assembles the puzzle to generate a detailed dataset. A set of raw 2D images (*c*) is taken every frame via an RGB camera. A digital twin (*d*) of the performed action is generated during the task, which is utilized to generate a one-to-one virtual–physical mapping image (*e*). These overlay images are then used to create semantic segmentation masks (*f*). The collision data between hands and objects generates fine-grained human–object (*g*) and object–object (*h*) interactions. This fine-grained information and the provided end and start conditions are used to create coarse-grained action segmentations (*i*).

information that is stored in a CSV file. This collision information, combined with manual video action segmentation annotations (constitute level 3 annotations as shown in Fig. 2), allows us to segment the video into coarse-grained level 2 annotations as shown in Fig. 2. This approach allows for the fine-grained detection of human–object and object–object interactions, providing a novel way to classify and divide tasks into relevant steps. Please refer Algorithm 1 for details.

Algorithm 1: Action segmentation from collision information

```
Inputs: CSV file
/* Start and end collision information of each step            */
DECLARE list : array of size (N−1)
  /* where N is number of steps in the video.
   list contains information of number of
  objects collide at the end of each step                      */
INITIALIZE step_start_time: array of size(N) = []
  step_end_time: array of size(N) = []
  /* where N is number of steps                                 */
for frame = 0 To N do
  object_colliding = x
    /* where x is number of objects colliding in frame          */
  for i = 0 To length(list) do
    if list[i] == object_colliding then
      step_end_time[i] = frame/fps
        step_start_time[i + 1] = frame/fps
    end
  end
end
```

3.3.4 Handling Tracking Loss. As mentioned earlier, AnnotateXR relies on Antilatency for 6DOF object tracking and Oculus SDK for hand tracking. Though both these are fairly reliable, there are cases where tracking could be lost. In the case of Antilatency, if the sensors are directly occluded, there tend to be discrepancies in the way the objects are tracked. In the case of hand tracking, when the user moves their hands out of view of the HMD, tracking can be lost. When such discrepancies in tracking occur, the associated virtual models automatically snap to the virtual world's origin. Such discrepancies are unwanted since accurate virtual–physical mapping is essential to generate precise 6DOF and segmentation mask data sets. To address this, whenever the tracking of the objects or the hands is lost, corresponding frames are dropped, and a UI element is rendered (shown in Fig. 6) to the user indicating that the tracking is lost. The users are then instructed to re-perform the task.

3.4 Use Cases. The two tasks mentioned are examples of how the proposed workflow can be used in a practical setting. In the first task, a simulated action of welding two steel flat plates is shown, while in the second task, a simulated action of drilling into a block of wood is demonstrated. A sample of this can be found in Fig. 7. In both cases, only a simulation of the action was performed instead of the actual task. For the welding use case, this was done to conform to safety standards. While for the drilling task, it was not possible to track the moving spindle of the drill during task operation. Due to the complexity of the geometry involved, object alignment required multiple attempts.

These tasks were chosen to highlight the generalizability of the workflow and to identify potential limitations (mentioned in Sec. 6), as they are both spatial tasks that require detailed data sets. The use of XR applications for data annotations is still an area of active research, and in this work, a more straightforward use case was explored using tangrams. This approach reduces the complexity of the pre-processing steps involved in task performance for the users and provides insight into the task heuristics and workflow verification.

4 Evaluation

Since the goal of AnnotateXR is to produce auto-annotated data for CV, we designed our study and utilized evaluation metrics used by the CV community [7] to ***measure the quality*** of annotated data obtained from AnnotateXR against human annotated data. We aim to address

- What is the effect on performance of standard ML models using annotations from AnnotateXR. We measured this across three applications: object detection (refers to identifying and localizing of the objects in the image), object segmentation (refers to classify each pixel in the image), and action segmentation (refers to temporally segment a video and each segment is then classified to different action labels).
- What is the quality of annotated data across three applications and the sensitivity of these annotations with respect to capture distance, capture orientation, and occlusion percentage.

We chose to evaluate three CV applications (object detection, segmentation, and action segmentation). Since action segmentation incorporates information from three other parameters: H–O, O–O

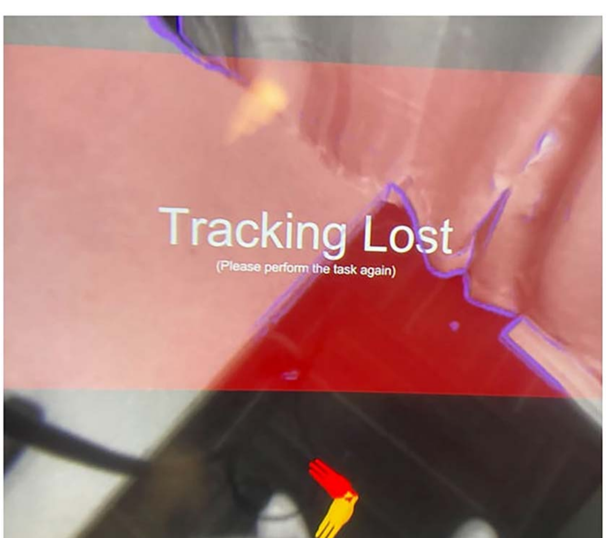

Fig. 6 UI element to warn users of tracking loss during data collection

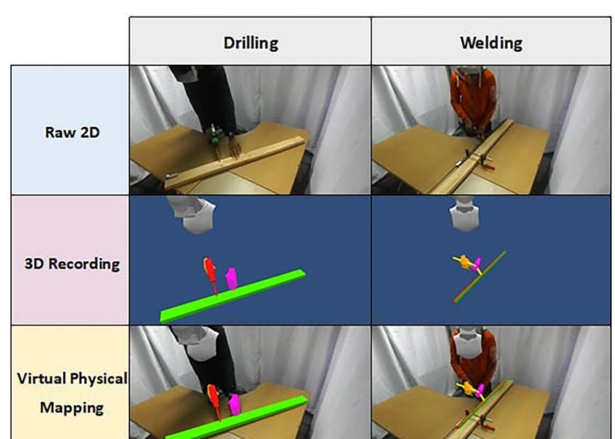

Fig. 7 Example applications to show the generalizability of AnnotateXR: (left) drilling and (right) simulated actions of welding

interactions, and 3D scene recording (refer Sec. 3.3.3 and Algorithm 1), insights from action segmentation performance results will lead to insights into these other parameters (H–O, O–O, and 3D). This approach is similar to recent CV experimentation of indirect validation; for example, Action Genome [18] verifies the importance of H–O interaction annotation by evaluating corresponding action recognition performance. In addition, it would be needless to make users manually annotate for all variations of data leading to user fatigue.

Due to the lack of an equivalent baseline system and the nascent stage of research into XR-based data annotation interfaces, we limited our evaluation to 12 users. This was done as part of a "first-use" study, aimed at the initial assessment of our AnnotateXR system. A "first-use study" is a controlled experiment conducted in a laboratory to evaluate the ease of use and effectiveness of a tool or system [90].

4.1 Participants. We invited 12 participants (three female, nine male) (P1–12) from a technical university's graduate and undergraduate programs. The mean age was 23.5 years. Five participants had prior experience with machine learning or computer vision, two of whom use ML-based CV algorithms for research. The other three have taken courses in CV. Five users reported using a VR headset (less than three times) within the past year, four users had no experience with VR and three users reported regular use of VR for games (ranging from once a week to once a month).

4.2 Study Design. We designed a two-session study to collect annotated data obtained under two conditions: (1) manual annotation via current state-of-art tools, and (2) automated AnnotateXR system. A four-piece tangrams puzzle was the chosen task for an in-lab study. A simple task was chosen to keep user training time to a minimum and reduce user fatigue (more complex applications were explored as part of the use case demonstration discussed in Sec. 3.4). This allow us to keep primary focus on system's usability and the interactions performed. Since the system requires multiple interactions such as alignment and tracking, ensuring it is user-friendly and intuitive is critical for its adoption and effectiveness. Both sessions lasted for about 2 h and 15 min, and the users were compensated with a $30 Amazon gift card.

Procedure Session 1: Upon users' arrival, an explanation of the study was provided, followed by a signature on the consent form. The researchers provided the users with instructions on the four-step assembly of the tangrams puzzle. Users were instructed not to change the sequence of steps and perform only one action at a time (i.e., not to assemble two pieces at the same time). The users were offered a 5 min practice time, after explanation. Sample puzzle pieces were provided alongside printed instructions for practice. After practice, the researchers tested each user to verify their familiarity with the task.

After training, the users were brought into the Antilatency tracking space. The researchers first demonstrated AnnotateXR's features, such as virtual–physical alignment and data capture. The users were provided with practice time to familiarize themselves with the system. All users said they were comfortable using the system with less than 5 min of practice. During the study, the users were asked to align the physical and the virtual models of all four tangrams. After which, the users were asked to assemble the tangrams in the same sequence as practiced. The users were then asked to perform the task under 12 different capture conditions.

The 12 capture condition parameters were: four different occlusion conditions ranging from 5%, 10%, 15%, and 20% occlusion of objects as shown in Fig. 8 (the occlusion conditions are determined by the amount of surface area obscured by another object in the tangrams); four different distances between capture (the camera) and assembly environment (i.e., the desk) varied by a delta of 20 cm, with starting distance of 30 cm; and four different camera locations chosen to evaluate view variance at 0 deg, 45 deg, 90 deg, and 135 deg from a horizontal axis to the desk.

Fig. 8 Occlusion conditions during user study: (*a*) 5 % occlusion and (*b*) 20 % occlusion

Procedure Session 2: In Session 2, we explained the concept of semantic segmentation and action segmentation to the users, and then demonstrated widely used annotation tools. We used supervisely [14] to annotate the segmentation mask and Vidat [39] to annotate action tasks in videos. Each user was provided with one image and one video for practice. After training, the users were asked to annotate two randomly chosen RGB frames from the data collected in the previous session. Each user was asked to annotate two frames/images per case for a total of 24 images, followed by one video per case for action segmentation. ***Segmentation mask labels*** and ***action labels*** were created before the study.

Measures

Prior to the study, the participants were asked to fill out a demographic questionnaire. Upon completing the data capture in Session 1, a System Usability Scale (SUS) [91] survey was administered to the users to test the usability of AnnotateXR. In addition to this, during Session 1 we collected the time taken for virtual–physical alignment, total number of data points collected and time taken for data collection. During Session 2, the time taken for manual annotation of images and videos was collected. After Session 2, the researchers showed the users visually generated segmentation masks and action segmentation data for both the manual and auto-annotated cases. Finally, a semi-structured interview was conducted to collect qualitative feedback on both systems.

5 Results and Discussion

We performed a comparative analysis to evaluate the data's quality and performance. Finally, we report the results along with the manual annotation time, total amount of data collected, usability, and qualitative results in the following section.

5.1 Data Collection. AnnotateXR was able to generate a total of 112,737 semantically segmented and labeled image frames during the entire course of the study, while users performed the assembly task for a total of 66.55 min (12 users). 144 videos were also annotated for action segmentation simultaneously by our system. The mean time for virtual–physical alignment for four objects with AnnotateXR by the users was M = 1.2 min; SD = 0.86. In the second session, the users manual annotation time for semantic segmentation per image were M = 1.61 min, SD = 0.93 for occlusion variation; M = 1.19 min, SD = 0.58 for distance and M = 1.06 min, SD = 0.52 for orientation. The annotation time for action segmentation are: M = 1.29 min, SD = 0.40.

The user's manual annotation time varied based on the capture parameters. The users spent more time annotating occluded data than the other conditions. This observation correlates with prior work [52] that reported difficulty with annotating segmentation masks over objects in a cluttered scene. However, by automating the annotation, similar to AnnotateXR, it is possible to circumvent this limitation with the variation of capture constraint.

5.2 Performance. ***Object Detection:*** For the performance evaluation we used Faster RCNN [92], a commonly used object detector pre-trained on MS COCO [93] data set. A mean average

Table 2 Results from the user study: virtual–physical alignment time; number of data points generated using AnnotateXR; manual annotation time for images and video action segmentation under various capture conditions (occlusion, distance, and orientation), and SUS scores

			Manual annotation												
			Time to annotate images (min)						Time to annotate video (min)						
			Occlusion		Distance		Orientation		Occlusion		Distance		Orientation		
User no.	Alignment time (min)	No. of datapoints	AVG	SD	AVG	SD	AVG	SD	AVG	SD	AVG	SD	AVG	SD	SUS
1	1.42	6864	3.86	1.72	3.68	0.66	3.40	0.61	1.61	0.14	1.35	0.21	1.15	0.07	90.0
2	0.42	8541	5.79	0.80	4.30	0.63	3.32	0.84	1.60	0.05	1.71	0.09	1.53	0.29	85.0
3	0.75	9444	5.48	1.59	2.83	0.43	3.29	0.17	2.24	0.77	1.48	0.05	1.56	0.41	100.0
4	0.58	7305	5.94	1.20	4.13	0.33	3.70	0.48	1.22	0.07	1.06	0.08	0.90	0.13	97.5
5	1.42	13,652	2.88	1.19	2.11	0.09	2.15	0.14	1.84	0.35	1.38	0.07	1.41	0.20	100.0
6	1.25	7703	2.16	0.26	1.74	0.08	1.69	0.20	1.63	0.47	1.19	0.19	0.91	0.04	85.0
7	1.00	13,205	3.26	0.81	2.31	0.86	1.60	0.13	1.53	0.23	0.99	0.21	0.91	0.03	82.5
8	0.58	8518	1.59	0.14	1.43	0.10	1.26	0.03	1.42	0.10	0.98	0.21	0.98	0.08	97.5
9	3.58	9210	2.37	0.03	1.85	0.47	1.43	0.11	1.78	0.48	1.12	0.15	1.11	0.08	85.0
10	0.42	9522	2.31	0.22	1.67	0.16	1.44	0.12	1.50	0.22	1.13	0.26	0.98	0.15	97.5
11	1.58	6586	1.51	0.24	0.95	0.05	0.83	0.10	1.23	0.34	0.95	0.09	1.08	0.54	85.0
12	1.50	12,187	1.59	0.46	1.56	0.21	1.38	0.20	0.98	0.08	1.04	0.18	1.31	0.25	87.5

Note: We provide the manual annotation time for users' image and video action segmentation, which will provide a helpful benchmark for future work.

precision metric is used to evaluate model performance with the intersection of union (IOU) threshold as 0.5 and 0.75 (well accepted by the CV community [7]). The data set was split as 70:30 for training and testing. The Faster RCNN model was trained on 200 users annotated images with 88 images for testing = 288 (2 per case $\times$ 12 cases $\times$ 12 users) and corresponding system annotated images until convergence; each image contained four tangram objects. These results are reported in Sec. 5.2. A paired sample t-test between user and AnnotateXR IOU for 0.5 and 0.75 were: $t(87) = 1.47$; $p = 0.14 > 0.05$; and $t(87) = 1.87$; $p = 0.06 > 0.05$; respectively.

Object Segmentation: The evaluation for segmentation was similar to object detection, except the model was a commonly used pre-trained mask RCNN [94] on MS COCO [93]. These results are reported in Sec. 5.2 and a paired sample t-test between user and AnnotateXR IOU was $t(87) = 1.35$; $p = 0.18 > 0.05$.

Action Segmentation: Out of 144 videos collected from the users (12 users $\times$ 12 cases), 100 videos were set for training and the rest for testing (70:30). We separately trained the Bi-LSTM [95] model until convergence on both the user and system annotations. The gating mechanism in LSTM implicitly learn temporal dynamics and a representation within and between action [96], making it ideal for evaluation. We used frame level classification accuracy (widely accepted by the CV community [97]) for action segmentation evaluation. Results reported in Sec. 5.2 and a paired sample t-test user and AnnotateXR data was: $t(43) = 0.47$; $p = 0.63 > 0.05$.

Discussion: From the results, we realize that there is no statistical difference in performance between manual user annotation and auto-system annotated data across all three CV applications. This insight is interesting as this suggests that the data collected with tools such as AnnotateXR are able to perform just as well as currently commercially used interfaces. This coupled with the capability of AnnotateXR to handle multi-modal large-scale data annotations highlights our system strength and also suggests that AnnotateXR can have a significant impact the CV community to develop their models.

5.3 Quality.

We evaluate the annotation quality by first comparing it against a "standardized annotation" scrupulously created by the authors. It is similar to evaluation protocols established in prior work LabelAR [53], with "gold standard" labels. The gold standard labels were collected by three researchers, each with 1–3 years of experience in collecting and annotating computer vision datasets. Each image and video was individually annotated by these researchers, with final annotations determined through a consensus discussion among all three. This rigorous process ensures the reliability and accuracy of the annotations, providing a gold standard benchmark for evaluating the performance of the AnnotateXR system. We are expanding this approach to evaluate quality annotation metrics beyond labeling to include semantic segmentation of objects and action segmentation. Hence, all 288 image frames and 144 videos were carefully annotated by the researchers.

We used a bounding box IOU metric between our standardized and manual annotations and compared the results with the same IOU metric between standardized and AnnotateXR annotations for object detection. A similar analysis was performed between the three groups' annotations for semantic and action segmentation, but the metrics used were pixel-wise IOU and frame-level classification accuracy, respectively.

We then performed a paired sample t-test on the corresponding data capture conditions (occlusion, distance, and view orientation) and presented the results as follows:

Object Detection Occlusion $t(95) = 1.87$; $p = 0.06 > 0.05$; Distance $t(95) = 1.91$; $p = 0.06 > 0.05$; View orientation $t(95) = 1.77$; $p = 0.08 > 0.05$;

Object Segmentation Occlusion $t(95) = 1.90$; $p = 0.06 > 0.05$; Distance $t(95) = 0.56$; $p = 0.36 > 0.05$; orientation $t(95) = 1.8$; $p = 0.07 > 0.05$;

Action Segmentation $t(143) = 1.1$; $p = 0.27 > 0.05$; (Analyzed together as capture conditions don't play a role for segmenting videos)

Discussion: We realized no statistical difference in IOU accuracy and frame-level classification accuracy based on the analysis. The marginally higher p-value, above 0.05, may be due to the limited quantity of manually annotated data. Consequently, it might be challenging to derive meaningful insights regarding annotation quality. Nonetheless, we were still able to use the auto-annotated data to train a Faster RCNN and Bi-LSTM model, as described in Sec.5.2. This leads us to conclude that auto-annotated data remains usable. Our finding is still evidence that AnnotateXR can produce annotated data with reduced human effort while still maintaining the quality of the data that is usable for training ML models, despite the variation in data capture conditions (occlusion, capture distance, and view orientation).

5.4 Usability and Qualitative Feedback.

The user's reported an M = 91; SD = 6.86 SUS (refer Table 2). This score is promising since an average score of 70, and above translates to "excellent"

Table 3 Results of performance evaluation between AnnotateXR and manual user annotations

	Object detection (mIOU > 0.5)	(mIOU > 0.75)	Object segmentation mIOU	Action segmentation Accuracy
User	0.48	0.23	48.6	53.5
System	0.49	0.21	49.7	54.1

Note: For three applications: object detection, object segmentation, and action segmentation.

usability, as indicated in Bangor et al. [98]. Qualitative feedback obtained from users in post-study interviews also backs the quantitative score. All 12 participants stated they would prefer to use AnnotateXR over manual annotation due to its ease of use and automated annotation approach. P3: "*Mentally [cognitively], since I was performing repeated task, I got irritated with the manual approach* "; P5: "*I will prefer your [AnnotateXR] system so that I don't have to do the work.*" One of 12 participants commented they would prefer a manual method for a small amount of data. P4: "*For very, very few images, I might do it manually instead of setting up. But if I had to do a lot of data, I would like it automated.*"

The participants were largely optimistic about AnnotateXR. In addition, 7 of 12 participants commented positively on applying extended reality for automating annotations. P11: "*It's a fascinating system for sure. Fascinating application. Very cool.*" P4: "*I think it's really cool. I think I can see the benefit, or we will have someone sit and do it manually versus having something done on real time.*" Comments on recommendations for improvement revolved around two categories: "tracking loss" (four participants) and "providing visual feedback for virtual–physical alignment" (three participants) both have been present in limitation and future work (Sec. 6).

The participants with prior experience with data annotation were able to provide additional insight into the effectiveness of AnnotateXR's workflow. In particular, they noted how quick the in situ data capture technique was compared to the post-hoc protocol that is currently prevalent in the field. This result is similar to the findings of recent work by Zhou and Yatani [65] on gesture-aware in situ object annotations. However, these users also mentioned the challenges of creating an elaborate tracking system for their applications. Despite this limitation, they believed that the benefits of the in situ approach outweigh the additional effort required to setup the tracking system. Overall, the feedback from these participants suggests that AnnotateXR's workflow is both effective and efficient for large-scale data annotation tasks.

6 Limitation and Future Work

In our study, we focused on exploring the potential of using XR applications for data annotations in a simple use case. However, we acknowledge that there are limitations to our approach, such as the assumption of ideal lighting and the limitations of sensor size and occlusion. In the following section, we will outline the limitations of our approach and provide recommendations for future research directions in the use of XR-based annotation tools (Table 3).

Object tracking and size: Direct occlusion of sensors or HMD prevents AnnotateXR from tracking the objects or hand pose. We currently handle this by dropping frames from recording (refer Sec. 3.3.4) and allowing users to redo the task with a UI prompt (refer Fig. 6). Four of the twelve participants mentioned this limitation during the post-study interview. P8: "*When I grabbed the object but accidentally touched the sensor, I was supposed to redo the task. I wish that could be better.*" Another limitation is concerning object size and flexible objects. Due to the size of the sensors, objects smaller or comparable to sensor size ($18\,mm \times 66\,mm$) would not be compatible with the system. These are current inherent limitations of sensor-based spatial tracking technology. We believe with advances in sensing hardware such as smaller tracking setup, markers, and electronics, these limitations can be addressed. In addition, our use of sensor-based protocols for data collection is in line with previous research in the field [55,56,78,95]. The users during the study were also asked to treat the sensors as part of the object. While previous work has explored removing sensors from RGB pixel information such as Ref. [55], we did not pursue this in our study as it is not the focus of our work.

Object Alignment: Three of the twelve participants commented on providing additional features for virtual–physical alignment. P12: "*I would recommend while doing the alignments, some sort of feedback [referring to visual widget] would be nice.*" However, these suggestions did not limit users from creating usable annotated data sets from AnnotateXR, as confirmed by results presented earlier (refer Secs. 5.3 ans 5.2). Prior work in human computer interaction (HCI) has explored virtual–physical alignment for AR creation in SnapToReality [99] and precise virtual model alignment for VR in Hayatpur et al. [100]. Incorporating such design principles in AnnotateXR workflow might improve the performance and quality of annotations.

Human Pose: Currently, AnnotateXR can partially capture human pose: head and hand pose. Although this would suffice for many real-world applications [101], our system can be improved by capturing the entire human pose better with advances in XR HMD hardware such as wearables [102,103] or the availability of smaller size sensors, leading to higher quality human pose annotations. Alleviating these limitations will lead to data sets of multiple synthetic humans with realistic poses and many human–object interactions. Furthermore, these challenging data sets can support research in higher performance algorithms to tackle challenging problems in computer vision related to human pose-based interactions.

User Study: In our current user study setup, we conducted a controlled "first-use" study with 12 participants [90] to establish a baseline for the system's performance and to gather initial feedback. While this study provided valuable insights, we recognize the importance of expanding our research to enhance the robustness and applicability of the AnnotateXR system.

Future studies should involve a larger number of participants, complex tasks, diverse objects, and varied environmental conditions. Conducting open studies with the AnnotateXR will allow us to better understand how the system manages real-world complexities and diverse scenarios. This will enable a comprehensive evaluation of AnnotateXR's capabilities across various real-world application domains. The study also evaluates computer vision algorithms using data annotated by humans and collected via AnnotateXR, providing comparative insights. Future work should include comparisons of annotation quality with existing datasets and assessments of computer vision algorithm performance.

7 Conclusion

This work introduces AnnotateXR, an extended reality application capable of in situ collection and annotation of data with a single demonstration to support several CV applications. AnnotateXR relies on the virtual–physical alignment to generate a digital twin coupled with hand tracking information offered by modern HMDs to obtain annotation cues. AnnotateXR uses the physical and virtual mapping information to generate segmentation masks for images and H–O/O–O interaction information to identify task actions automatically.

With the help of a user study, we showed that AnnotateXR could simultaneously collect and annotate over 112,000 image segmentation and 144 video based action segmentation in about 67 min. Extrapolating average 1.29 min/data point, it would take over 2000 h to manually collect and annotate the same dataset (based on mean user annotation rate). We performed a comparative analysis across three annotation applications: object detection, semantic segmentation, and action segmentation. Our study also collected data under various capture conditions that are present in real world such as varying occlusion, distance, and view orientation. AnnotateXR across all these conditions and is a promising tool for generating large-scale customized data for various CV applications.

We also have discussed limitations of the system and identified potential future research directions for the HCI and CV community to explore. We believe extended reality applications such as AnnotateXR have great potential for auto-annotation of data, which can aid in quicker advancement and deployment of research-based ML and CV approaches.

Acknowledgment

We wish to give a special thanks to the reviewers and editors 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 expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., and Yang, R., 2018, "The Apolloscape Dataset for Autonomous Driving," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 954–960.

[2] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., et al., 2011, "Towards Fully Autonomous Driving: Systems and Algorithms," 2011 IEEE Intelligent Vehicles Symposium (IV), IEEE, pp. 163–168.

[3] Ronneberger, O., Fischer, P., and Brox, T., 2015, "U-Net: Convolutional Networks for Biomedical Image Segmentation," International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, pp. 234–241.

[4] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., 2013, "Vision Meets Robotics: The Kitti Dataset," Int. J. Rob. Res., **32**(11), pp. 1231–1237.

[5] Ipsita, A., Duan, R., Li, H., Chidambaram, S., Cao, Y., Liu, M., Quinn, A., and Ramani, K., 2023, "The Design of a Virtual Prototyping System for Authoring Interactive Virtual Reality Environments From Real-World Scans," ASME J. Comput. Inf. Sci. Eng., **24**(3), p. 031005.

[6] Ipsita, A., Li, H., Duan, R., Cao, Y., Chidambaram, S., Liu, M., and Ramani, K., 2021, "VRFromX: From Scanned Reality to Interactive Virtual Experience With Human-In-The-Loop," Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, CHI EA '21, Yokohama, Japan, May 8–13, Association for Computing Machinery.

[7] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., 2009, "ImageNet: A Large-Scale Hierarchical Image Database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, June 20–25, IEEE, pp. 248–255.

[8] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S., 2016, "YouTube-8M: A Large-Scale Video Classification Benchmark," arXiv preprint arXiv:1609.08675.

[9] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., et al., 2015, "Shapenet: An Information-Rich 3D Model Repository," arXiv preprint arXiv:1512.03012.

[10] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T., 2011, "HMDB: A Large Video Database for Human Motion Recognition," 2011 International Conference on Computer Vision, Barcelona, Spain, Nov. 6–13, IEEE, pp. 2556–2563.

[11] Rai, N., Chen, H., Ji, J., Desai, R., Kozuka, K., Ishizaka, S., Adeli, E., and Niebles, J. C., 2021, "Home Action Genome: Cooperative Compositional Action Understanding," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, June 19–24, pp. 11184–11193.

[12] Amazon, 2005, Amazon Mechanical Turk. https://www.mturk.com/.

[13] Amazon Web Services, 2018, Sagemaker Ground Truth, https://aws.amazon.com/sagemaker/groundtruth/.

[14] Supervisely, 2017, Supervisely, https://supervise.ly/.

[15] Analytics, 2016, Analytics, https://www.analytics.ai/.

[16] Li, Q., Ma, F., Gao, J., Su, L., and Quinn, C. J., 2016, "Crowdsourcing High Quality Labels With a Tight Budget," Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, WSDM '16, San Fransisco, CA, Feb. 22–25, Association for Computing Machinery, pp. 237–246.

[17] Yu, G., Tu, J., Wang, J., Domeniconi, C., and Zhang, X., 2021, "Active Multilabel Crowd Consensus," IEEE Trans. Neural Netw. Learn. Syst., **32**(4), pp. 1448–1459.

[18] Ji, J., Krishna, R., Fei-Fei, L., and Niebles, J. C., 2020, "Action Genome: Actions as Compositions of Spatio-Temporal Scene Graphs," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, June 16–18.

[19] Richter, S. R., Vineet, V., Roth, S., and Koltun, V., 2016, "Playing for Data: Ground Truth From Computer Games," *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Springer International Publishing, pp. 102–118.

[20] Mo, K., Qin, Y., Xiang, F., Su, H., and Guibas, L., 2022, "O2O-Afford: Annotation-Free Large-Scale Object-Object Affordance Learning," Proceedings of the 5th Conference on Robot Learning, A. Faust, D. Hsu, and G. Neumann, eds., Vol. 164 of *Proceedings of Machine Learning Research*, PMLR, pp. 1666–1677.

[21] Tremblay, J., To, T., Sundaralingam, B., Xiang, Y., Fox, D., and Birchfield, S., 2018, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," arXiv preprint arXiv:1809.10790.

[22] de Melo, C. M., Torralba, A., Guibas, L., DiCarlo, J., Chellappa, R., and Hodgins, J., 2022, "Next-Generation Deep Learning Based on Simulators and Synthetic Data," Trends Cogn. Sci., **26**(2), pp. 174–187.

[23] Huang, L., Zhang, B., Guo, Z., Xiao, Y., Cao, Z., and Yuan, J., 2021, "Survey on Depth and RGB Image-Based 3D Hand Shape and Pose Estimation," Virtual Real. Intell. Hardw., **3**(3), pp. 207–234.

[24] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2016, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 16–July 1, pp. 779–788.

[25] Long, J., Shelhamer, E., and Darrell, T., 2015, "Fully Convolutional Networks for Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 7–12, pp. 3431–3440.

[26] Shou, Z., Wang, D., and Chang, S.-F., 2016, "Temporal Action Localization in Untrimmed Videos Via Multi-stage CNNS," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 16–July 1, pp. 1049–1058.

[27] Peng, S., Liu, Y., Huang, Q., Zhou, X., and Bao, H., 2019, "Pvnet: Pixel-Wise Voting Network for 6DOF Pose Estimation," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, June 16–20, pp. 4561–4570.

[28] Kwon, T., Tekin, B., Stühmer, J., Bogo, F., and Pollefeys, M., 2021, "H2O: Two Hands Manipulating Objects for First Person Interaction Recognition," Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10138–10148.

[29] Unmesh, A., Jain, R., Shi, J., Chaithanya Manam, V. K., Chi, H.-G., Chidambaram, S., Quinn, A., and Ramani, K., 2024, "Interacting Objects: A Dataset of Object-Object Interactions for Richer Dynamic Scene Representations," IEEE Rob. Autom. Lett., **9**(1), pp. 451–458.

[30] Murez, Z., As, T. V., Bartolozzi, J., Sinha, A., Badrinarayanan, V., and Rabinovich, A., 2020, "Atlas: End-to-End 3D Scene Reconstruction From Posed Images," European Conference on Computer Vision, Glasgow, UK, Aug. 23–28, Springer, pp. 414–431.

[31] Wright, L., and Davidson, S., 2020, "How to Tell the Difference Between a Model and a Digital Twin," Adv. Model. Simul. Eng. Sci., **7**(1), pp. 1–13.

[32] Hughes, A., 2018, Forging the Digital Twin in Discrete Manufacturing: A Vision for Unity in the Virtual and Real Worlds.

[33] Antilatency, 2021. Antilatency. https://antilatency.com/.

[34] Oculus, 2020. "Oculus Quest 2," From https://www.oculus.com/quest-2/, Accessed April 4, 2021.

[35] Jain, R., Shi, J., Duan, R., Zhu, Z., Qian, X., and Ramani, K., 2023, "Ubi-TOUCH: Ubiquitous Tangible Object Utilization Through Consistent Hand-Object Interaction in Augmented Reality," Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST '23, San Fransisco, CA, Oct. 29–Nov. 2, 2022, Association for Computing Machinery.

[36] Scheff-King, M., 2014, Download, Edit and Print Your Own Parts From McMaster-Carr. https://www.instructables.com/Download-Edit-And-Print-Your-Own-Parts-From-McMast/.

[37] Stratasys, 2022. Grabcad Community. Retrieved March 8, 2022, From https://www.traceparts.com/en.
[38] Traceparts, 1990. "TraceParts," Retrieved March 8, 2022, From https://www.traceparts.com/en.
[39] CVML, A., 2021, "Vidat," https://github.com/anucvml/vidat.
[40] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T., 2018, Bdd100k: A Diverse Driving Dataset for Heterogeneous Multitask Learning. DOI: 10.48550/ARXIV.1805.04687, https://arxiv.org/abs/1805.04687.
[41] Kawaguchi, K., Kaelbling, L. P., and Bengio, Y., 2017, "Generalization in Deep Learning," arXiv preprint arXiv:1710.05468.
[42] Redmon, J., and Farhadi, A., 2017, "YOLO9000: Better, Faster, Stronger," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26.
[43] Redmon, J., and Farhadi, A., 2018, YOLOV3: An Incremental Improvement. DOI: 10.48550/ARXIV.1804.02767, https://arxiv.org/abs/1804.02767.
[44] Liu, Y., Wang, L., Ma, X., Wang, Y., and Qiao, Y., 2021, Fineaction: A Fine-Grained Video Dataset for Temporal Action Localization. DOI: 10.48550/ARXIV.2105.11107, https://arxiv.org/abs/2105.11107.
[45] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., 2014, "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision, Zurich, Switzerland, Sept. 6–12, Springer, pp. 740–755.
[46] Microsoft, 2019. Coco 2019 Stuff Segmentation Task. Retrieved April 6, 2022, From https://cocodataset.org/#stuff-2019.
[47] Microsoft, 2020. COCO 2020 Panoptic Segmentation Task. Retrieved April 6, 2022, From https://cocodataset.org/#panoptic-2020.
[48] Microsoft, 2020. COCO 2020 Densepose Task. Retrieved April 6, 2022, From https://cocodataset.org/#densepose-2020.
[49] Microsoft, 2015. COCO 2015 Image Captioning Task. Retrieved April 6, 2022, From https://cocodataset.org/#captions-2015.
[50] Microsoft, 2020. COCO 2020 Keypoint Detection Task. Retrieved April 6, 2022, From https://cocodataset.org/#keypoints-2020.
[51] Analytics, 2017. Aimultiple. https://aimultiple.com/.
[52] Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L., 2016, "What's The Point: Semantic Segmentation With Point Supervision," Computer Vision - 14th European Conference, ECCV 2016, Proceedings, B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, pp. 549–565. 14th European Conference on Computer Vision, ECCV 2016, Oct. 8–16.
[53] Laielli, M., Smith, J., Biamby, G., Darrell, T., and Hartmann, B., 2019, "Labelar: A Spatial Guidance Interface for Fast Computer Vision Image Collection," UIST '19, New Orleans, LA, Oct. 20–23, Association for Computing Machinery, pp. 987–998.
[54] Rennie, C., Shome, R., Bekris, K. E., and De Souza, A. F., 2016, "A Dataset for Improved RGBD-Based Object Detection and Pose Estimation for Warehouse Pick-and-Place," IEEE Rob. Autom. Lett., **1**(2), pp. 1179–1185.
[55] Garon, M., Laurendeau, D., and Lalonde, J.-F., 2018, "A Framework for Evaluating 6-DOF Object Trackers," Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sept. 8–14, pp. 582–597.
[56] Taheri, O., Ghorbani, N., Black, M. J., and Tzionas, D., 2020, "GRAB: A Dataset of Whole-Body Human Grasping of Objects," Computer Vision – ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds., Springer International Publishing, pp. 581–600.
[57] von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., and Pons-Moll, G., 2018, "Recovering Accurate 3D Human Pose in the Wild Using IMUS and a Moving Camera," Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sept. 8–14, pp. 601–617.
[58] Chen, L.-C., Fidler, S., Yuille, A. L., and Urtasun, R., 2014, "Beat the Mturkers: Automatic Image Labeling From Weak 3D Supervision," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, June 24–27, pp. 3198–3205.
[59] Xie, J., Kiefel, M., Sun, M.-T., and Geiger, A., 2016, "Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 16–July 1, pp. 3688–3697.
[60] Castrejon, L., Kundu, K., Urtasun, R., and Fidler, S., 2017, "Annotating Object Instances With a Polygon-RNN," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26, pp. 5230–5238.
[61] Acuna, D., Ling, H., Kar, A., and Fidler, S., 2018, "Efficient Interactive Annotation of Segmentation Datasets With Polygon-RNN++," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, June 18–22, pp. 859–868.
[62] Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., and Grundmann, M., 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 (CVPR), pp. 7822–7831.
[63] Qian, X., He, F., Hu, X., Wang, T., and Ramani, K., 2022, "Arnnotate: An Augmented Reality Interface for Collecting Custom Dataset of 3D Hand-Object Interaction Pose Estimation," Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, UIST '22, Bend, OR, Oct. 29–Nov. 2, Association for Computing Machinery.
[64] Doula, A., Güdelhöfer, T., Matviienko, A., Mühlhäuser, M., and Sanchez Guinea, A., 2022, "Immersive-Labeler: Immersive Annotation of Large-Scale 3D Point Clouds in Virtual Reality," ACM SIGGRAPH 2022 Posters, SIGGRAPH '22, Vancouver, BC, Canada, Aug. 7–11, Association for Computing Machinery.
[65] Zhou, Z., and Yatani, K., 2022, "Gesture-Aware Interactive Machine Teaching With In-Situ Object Annotations," Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, UIST '22, Bend, OR, Oct. 29–Nov. 2, 2022, pp. 1–14.
[66] Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., et al., 2018, "Scaling Egocentric Vision: The Epic-Kitchens Dataset," Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sept. 8–14.
[67] Sigurdsson, G. A., Gupta, A., Schmid, C., Farhadi, A., and Alahari, K., 2018, Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos. https://arxiv.org/abs/1804.09626.
[68] Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J., 2015, "Activitynet: A Large-Scale Video Benchmark for Human Activity Understanding," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, June 8–10.
[69] Murray, N., Marchesotti, L., and Perronnin, F., 2012, "AVA: A Large-Scale Database for Aesthetic Visual Analysis," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, June 16–21, pp. 2408–2415.
[70] Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., and Zhou, J., 2019, "Coin: A Large-Scale Dataset for Comprehensive Instructional Video Analysis," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, June 16–20.
[71] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A., 2017, The Kinetics Human Action Video Dataset.
[72] Materzynska, J., Xiao, T., Herzig, R., Xu, H., Wang, X., and Darrell, T., 2020, "Something-Else: Compositional Action Recognition With Spatial-Temporal Interaction Networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, June 16–18.
[73] Shao, D., Zhao, Y., Dai, B., and Lin, D., 2020, "Finegym: A Hierarchical Video Dataset for Fine-Grained Action Understanding," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
[74] Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A., 2016, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 16–July 1.
[75] Brostow, G. J., Fauqueur, J., and Cipolla, R., 2009, "Semantic Object Classes in Video: A High-Definition Ground Truth Database," Pattern Recogn. Lett., **30**(2), pp. 88–97.
[76] Vijayanarasimhan, S., and Grauman, K., 2012, "Active Frame Selection for Label Propagation in Videos," European Conference on Computer Vision, Firenze, Italy, Oct. 7–13, Springer, pp. 496–509.
[77] Ben-Shabat, Y., Yu, X., Saleh, F., Campbell, D., Rodriguez-Opazo, C., Li, H., and Gould, S., 2021, "The IKEA ASM Dataset: Understanding People Assembling Furniture Through Actions, Objects and Pose," Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Virtual, Jan. 5–9, pp. 847–859.
[78] Ahmad, S., Samarawickrama, K., Rahtu, E., and Pieters, R., 2021, "Automatic Dataset Generation From CAD for Vision-Based Grasping," 2021 20th International Conference on Advanced Robotics (ICAR), Virtual, Dec. 6–10, pp. 715–721.
[79] Microsoft, 2022. Kinect For Windows. Retrieved April 6, 2022, From https://developer.microsoft.com/en-us/windows/kinect/.
[80] Inc., N., 2022. OptiTrack - V120:Duo. Retrieved April 6, 2022, From https://optitrack.com/cameras/v120-duo/.
[81] Stereolabs, 2021. Zed Mini. https://www.stereolabs.com/zed-mini/.
[82] Chidambaram, S., Huang, H., He, F., Qian, X., Villanueva, A. M., Redick, T. S., Stuerzlinger, W., and Ramani, K., 2021, "ProcessAR: An Augmented Reality-Based Tool to Create In-Situ Procedural 2D/3D AR Instructions," Proceedings of the 2021 ACM Designing Interactive Systems Conference, DIS '21, Virtual, June 28–July 2, Association for Computing Machinery, pp. 234–249.
[83] Ramani, K., Chidambaram, S., Huang, H., and He, F., 2022, System and Method for Generating Asynchronous Augmented Reality Instructions. US Patent 11,380,069.
[84] Chidambaram, S., Reddy, S. S., Rumple, M., Ipsita, A., Villanueva, A., Redick, T., Stuerzlinger, W., and Ramani, K., 2022, "Editar: A Digital Twin Authoring Environment for Creation of AR/VR and Video Instructions From a Single Demonstration," 2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 326–335.
[85] Ramani, K., Chidambaram, S., and Reddy, S. S., 2024, Digital Twin Authoring and Editing Environment for Creation of AR/VR and Video Instructions From a Single Demonstration. US Patent App. 18/480,173.
[86] qlone, 2020, qlone. Retrieved May 5, 2020, From https://www.qlone.pro/.
[87] cognex, 2020, Cognex. Retrieved February 5, 2021, From https://www.cognex.com/products/machine-vision/3d-machine-vision-systems/in-sight-3d-l4000.
[88] display.land, 2020, display.land. Retrieved May 5, 2020, From https://get.display.land/.
[89] Unity, 2020, Mesh Collider. Retrieved April 6, 2022, From https://docs.unity3d.com/Manual/class-MeshCollider.html.
[90] Hartmann, B., Abdulla, L., Mittal, M., and Klemmer, S. R., 2007, "Authoring Sensor-Based Interactions by Demonstration With Direct Manipulation and

Pattern Recognition," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, CA, Apr. 28–May 3, pp. 145–154.
[91] Brooke, J., et al., 1996, "SUS-a Quick and Dirty Usability Scale," Usability Eval. Ind., **189**(194), pp. 4–7.
[92] Ren, S., He, K., Girshick, R., and Sun, J., 2016, "Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks," IEEE Trans. Patt. Anal. Mach. Intell., **39**(6), pp. 1137–1149.
[93] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., 2014, "Microsoft COCO: Common Objects in Context," *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., Springer International Publishing, pp. 740–755.
[94] He, K., Gkioxari, G., Dollár, P., and Girshick, R., 2017, "Mask R-CNN," Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, Oct. 22–29, pp. 2961–2969.
[95] Lea, C., Reiter, A., Vidal, R., and Hager, G. D., 2016, "Segmental Spatiotemporal CNNS for Fine-Grained Action Segmentation," European Conference on Computer Vision, Amsterdam, Netherlands, Oct. 11–14, Springer, pp. 36–52.
[96] Singh, B., Marks, T. K., Jones, M., Tuzel, O., and Shao, M., 2016, "A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, June 16–July 1, pp. 1961–1970.
[97] Kuehne, H., Arslan, A., and Serre, T., 2014, "The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, June 24–27,pp. 780–787.
[98] Bangor, A., Kortum, P., and Miller, J., 2009, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," J. Usabil. Stud., **4**(3), pp. 114–123.
[99] Nuernberger, B., Ofek, E., Benko, H., and Wilson, A. D., 2016, "Snaptoreality: Aligning Augmented Reality to the Real World," Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, San Jose, CA, May 7–12, Association for Computing Machinery, pp. 1233–1244.
[100] Hayatpur, D., Heo, S., Xia, H., Stuerzlinger, W., and Wigdor, D., 2019, "Plane, Ray, and Point: Enabling Precise Spatial Manipulations With Shape Constraints," Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST '19, Honolulu, HI, Apr. 25–30, Association for Computing Machinery, pp. 1185–1195.
[101] Cao, Y., Qian, X., Wang, T., Lee, R., Huo, K., and Ramani, K., 2020, "An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks," Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, Apr. 25–30, Association for Computing Machinery, pp. 1–13.
[102] Yoon, S. H., Huo, K., Zhang, Y., Chen, G., Paredes, L., Chidambaram, S., and Ramani, K., 2017, "iSOFT: A Customizable Soft Sensor With Real-Time Continuous Contact and Stretching Sensing," Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST '17, Quebec City, Canada, Oct. 22–25, Association for Computing Machinery, pp. 665–678.
[103] Paredes, L., Reddy, S. S., Chidambaram, S., Vagholkar, D., Zhang, Y., Benes, B., and Ramani, K., 2021, "FabHandWear: An End-to-End Pipeline From Design to Fabrication of Customized Functional Hand Wearables," Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., **5**(2), pp. 1–22.