Interactive 3D Annotation of Objects in Moving Videos from Sparse Multi-view Frames

KOTARO OOMORI, The University of Tokyo, Japan
WATARU KAWABE, The University of Tokyo, Japan
FABRICE MATULIC, Preferred Networks Inc., Japan
TAKEO IGARASHI, The University of Tokyo, Japan
KEITA HIGUCHI, Preferred Networks Inc., Japan

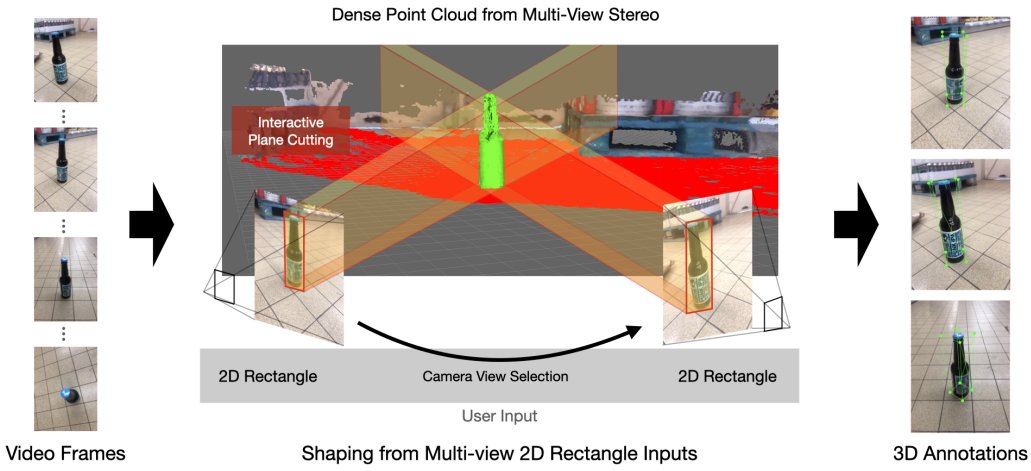

Fig. 1. The proposed method computes a 3D bounding box from multiple user-specified 2D bounding rectangles in selected frames of an RGB video covering different viewing angles.

Segmenting and determining the 3D bounding boxes of objects of interest in RGB videos is an important task for a variety of applications such as augmented reality, navigation, and robotics. Supervised machine learning techniques are commonly used for this, but they need training datasets: sets of images with associated 3D bounding boxes manually defined by human annotators using a labelling tool. However, precisely placing 3D bounding boxes can be difficult using conventional 3D manipulation tools on a 2D interface. To alleviate that

Authors' addresses: Kotaro Oomori, oomori-kotaro291@g.ecc.u-tokyo.ac.jp, The University of Tokyo, Japan; Wataru Kawabe, wkawabe@iis.u-tokyo.ac.jp, The University of Tokyo, Japan; Fabrice Matulic, fmatulic@preferred.jp, Preferred Networks Inc., Japan; Takeo Igarashi, takeo@acm.org, The University of Tokyo, Japan; Keita Higuchi, khiguchi@acm.org, Preferred Networks Inc., Japan.

2573-0142/2023/12-ART440 $15.00
https://doi.org/10.1145/3626476

burden, we propose a novel technique with which 3D bounding boxes can be created by simply drawing 2D bounding rectangles on multiple frames of a video sequence showing the object from different angles. The method uses reconstructed dense 3D point clouds from the video and computes tightly fitting 3D bounding boxes of desired objects selected by back-projecting the 2D rectangles. We show concrete application scenarios of our interface, including training dataset creation and editing 3D spaces and videos. An evaluation comparing our technique with a conventional 3D annotation tool shows that our method results in higher accuracy. We also confirm that the bounding boxes created with our interface have a lower variance, likely yielding more consistent labels and datasets.

CCS Concepts: • **Human-centered computing → Interactive systems and tools**.

Additional Key Words and Phrases: dataset creation, 3D annotation tool

ACM Reference Format:
Kotaro Oomori, Wataru Kawabe, Fabrice Matulic, Takeo Igarashi, and Keita Higuchi. 2023. Interactive 3D Annotation of Objects in Moving Videos from Sparse Multi-view Frames. *Proc. ACM Hum.-Comput. Interact.* 7, ISS, Article 440 (December 2023), 18 pages. https://doi.org/10.1145/3626476

1 INTRODUCTION

Recent advances in artificial intelligence (AI) have greatly extended the capabilities of computer vision (CV) to understand natural images and identify objects in them. 3D object recognition from 2D RGB images is an essential task for several kinds of applications, such as augmented reality, navigation, robotics, and autonomous driving. Public datasets for 3D object detection address a variety of contexts and scenarios, including city environments [3, 9, 10, 28] and indoor scenes [2, 38]. One of the key requirements of 3D recognition is the estimation of the 3D position and orientation of particular objects of interest. Recent deep learning techniques make it possible to infer such 3D information [1, 36], but labelled data is required to train the models. There are essentially two methods to create such datasets: 1) using computer graphics to generate automatically labelled synthetic data [12, 13, 30] and 2) manual annotation of captured images and videos (real data). Simulation can generate data at scale with no manual labour, but it is currently not practical for images requiring a high level of realism or which are difficult to model using 3D graphics. Recent studies also report "Sim-to-Real" problems with models trained on synthetic data not performing as well with real-world data [7, 42]. There is therefore still a need for datasets with real-world images labelled by human annotators and making that process accurate and efficient.

A common type of label for 3D object recognition is a 3D cuboid that envelops the object (bounding box). For example, the Objectron dataset [1] provides 2D short videos with 3D bounding boxes enclosing objects of interest. These bounding boxes are manually created and positioned by human annotators on a 3D world map corresponding to the video scene, ensuring that they project correctly onto the objects in the video, with adjustments made for other frames where mismatches occur. While this process is more efficient than annotating each frame individually, accurately fitting a 3D bounding box to an object in a 2D image remains challenging due to limited 3D perception. Additionally, annotations from different annotators may exhibit high variability.

In this work, we propose an iterative approach and a graphical tool for 3D object annotation from 2D video sequences that leverage 3D reconstruction techniques and simple image labelling with 2D rectangles. In the first preprocessing step, a dense point cloud of the scene and the planar surface on which objects of interest are placed are estimated from the video sequence. The location of the detected surface can be manually adjusted using the tool. Next, the annotator draws 2D rectangles on a few frames showing the object from different angles. The 2D rectangles are then back-projected to the 3D space to extract points belonging to the object and a rectangular cuboid tightly fitting those points is determined to create the object's 3D bounding box (Figure 1). These

bounding boxes can then be used as labels to train a 3D object detection neural network or select 3D objects to remove in videos, as we show in the application scenarios of our technique.

In a user evaluation, we compare our technique with labelCloud, a recent tool to create 3D bounding box labels for point cloud data [31, 32]. We employ two main metrics to assess the performance of our tool: 1) 3D intersection over union (IoU) between participant labels and ground truth, which measures accuracy, and 2) ratios of overlapping regions between the 3D annotations created by different participants, which measures label consistency. Results show that our method produces labels with higher accuracy and lower variance (i.e., there is more agreement on the solution) compared to labelCloud, which suggests that annotations created using our tool are of higher quality. Based on findings from the study, we discuss advantages of our approach and challenges to be addressed in future work.

In summary, our main contributions are twofold:

- A novel user interface for creating 3D bounding box annotations of objects in moving 2D videos using reconstructed dense 3D point clouds from the video and computes 3D bounding boxes of target objects selected by back-projecting the 2D rectangles.
- We performed our user study with 12 participants to discuss the advantages and challenges of the proposed interface . The results show that the proposed interface outperforms a 3D annotation tool (labelCloud [31, 32]) in metrics of annotation accuracy and variance among participants.

2 RELATED WORK

2.1 Variance, Bias, and Agreement of Annotation

Object recognition using supervised machine learning requires precisely annotated data. For 2D object detection, annotation errors and inconsistencies in bounding box labels lead to lower detection accuracy [15, 25]. Conversely, increasing the quality of annotations yields higher model performance [21]. We hypothesize that datasets with 3D bounding box annotations likely also contain errors and exhibit high variance, perhaps even more than with 2D labels, considering the potential for inconsistencies is increased in 3D. We seek to make the 3D annotating process more robust and straightforward as well as reduce variance among produced labels, by using simpler 2D labelling interfaces for images with which people are more familiar.

In annotation tasks, ensuring the generation of unbiased and consistent labels is a critical objective. To assess label reliability, inter-rater agreement has often been used as a key indicator [5]. In interaction research, there are several studies on annotation tools that consider agreement an important indicator. Higuchi et al. [11] uses annotation agreement from multiple annotators for video coding tasks in psychological assessments. For the evaluation of Glance [17], a tool with which multiple crowdworkers create annotations according to user-entered queries, the rate of agreement of annotations between crowdworkers is used as feedback on the quality of the query and annotation results. On the other hand, Tsandilas highlights the limitations of conclusions solely based on user consensus for gesture elicitation studies [37]. While elicitation studies likely generate more diverse and diverging results than image annotations for object detection, our user evaluation takes those considerations into account and we assess the performance of our technique using participant agreement for the produced annotations *as well as* accuracy when compared with ground truth data.

2.2 Object Annotation Tools

Object annotation applications range from fully manual interfaces to create shapes enclosing objects of interest [8, 20] to more modern tools which include semi-automatic segmentation assistance to

increase annotation efficiency [6, 14]. Labelme video [41] heuristically determines the 3D positional relationship for user-specified 2D mask annotations made by the user. LabelAR [16] uses mobile augmented reality devices to create 2D bounding box annotations of objects for machine learning detectors, where in the capturing process users are visually guided to move the camera around the object to cover a variety of angles.

The Pascal 3D Dataset [39] includes an annotation tool where users can select a template 3D CAD model of an object of interest and then manually position, orient, and resize the model so that it fits the object in the image. iLabel uses a few clicks on target objects and a multilayer perceptron to semantically segment objects from a scene captured with a handheld depth sensor [44]. 3D BAT (3D Bounding Box Annotation Tool) [46] is a web-based tool to annotate 3D LiDAR data of street scenes. Annotations are created in 3D space and transferred to the image domain using a geometric model. ARnnotate [29] leverages a hand-trackable AR device, guides the user to manipulate the virtual 3D bounding box, and supports the creation of a first-person, 3D posture dataset of the object and the hand grasping it. While these annotation tools aim to reduce the costs of manual labelling, they all require some form of 3D data as input, whereas our interface works with RGB videos.

The Objectron dataset [1] is manually labelled by human annotators using a 2D/3D tool. The application provides a side-by-side view that shows a 2D video frame and a 3D world map obtained from AR Core / AR Toolkit. Thanks to camera motion tracking and ground plane estimation, annotators can place bounding boxes on 2D frames that reflect the position and orientation of desired objects [1]. LabelAR provides a similar technique to define 3D object bounding boxes in its preparation process [16]. However, it can be difficult or tedious to precisely position 3D bounding boxes in 2D images due to limited 3D perception. This can result in high variance in the produced labels, as reported by the authors of Objectron in their investigation of annotation consistency for chair objects. LabelCloud [31, 32] provides a 3D view for visualizing dense point clouds from various inputs (e.g., LiDAR and coloured point clouds). The tool allows users to place a 3D bounding box for objects of interest onto a reconstructed surface plane using 3D manipulations. This approach based on direct 3D positioning may similarly result in imprecision and inconsistencies in the produced labels.

The goal of this work is to enhance the quality of 3D annotations of RGB videos for object detection without increasing human labour. Instead of working with 3D objects and tools to position bounding boxes, annotators need only create 2D rectangles on a few frames with objects of interest seen from different angles, which we hypothesize yields more precise and consistent labels.

3 PROPOSED METHOD

We aim to develop a method to create precise 3D annotations for datasets that can be used to train models for 3D object detection and pose estimation. We consider scenarios where objects of interest are captured in an RGB video, and 3D bounding box labels defined by human annotators. Like in 3D object recognition datasets ([1, 36], we assume target objects are placed on a planar surface.

The basic idea of our technique is to extract the points belonging to objects of interest from a dense point cloud reconstructed from the input RGB video and fit a box to those extracted points (Figure 1). Simple 2D graphical operations are used to define rectangles on selected frames showing the object from different perspectives. 3D points of the target object are then extracted by back-projecting the 2D rectangles to the 3D space. Finally, a 3D bounding box is tightly fitted to the extracted points. We expect this approach to result in more precise and consistent annotations.

[1]https://google.github.io/mediapipe/solutions/objectron.html (Accessed Feb. 15, 2023)

3.1 Interface Overview

Figure 2 shows the user interface (UI) of our proposed method. The upper part of the UI shows (A) a video frame image, (B) projected points onto the frame, and (C) the resulting 3D annotation, respectively. The user can change the current video frame by clicking on a thumbnail in the timeline (Figure 2 (D)). The interface provides functions to compute the 3D bounding box of annotated objects, as shown in Figure 2 (E).

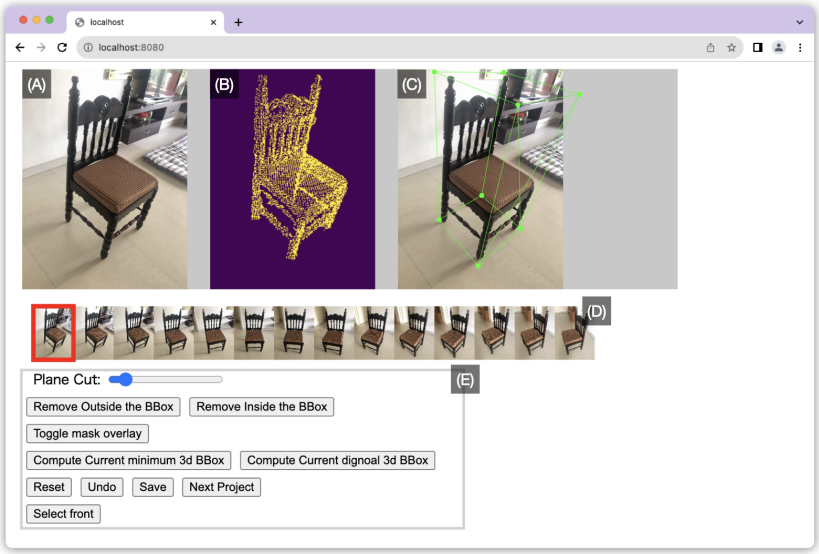

Fig. 2. Interface overview. (A) Input video frame (image view), (B) projected point cloud on the frame (point view), (C) computed 3D annotation result (result view), (D) video frame thumbnails, and (E) UIs for annotation

Given an input video capturing an object (e.g. a bottle) from various angles, the 3D annotation process with our tool is as follows (Figure 3): In a preprocessing step, the camera motion and the 3D structure of the scene of the video are computed. Before the user can start with the annotation process, the surface on which the object is placed needs to be identified via interactive plane cutting. Next, the user selects a video frame and adds 2D rectangles on the selected image or point views. Based on those 2D rectangle labels, the system determines the points belonging to the object and then automatically estimates its 3D bounding box.

3.2 Preprocessing

Typical input data for our annotation tool are shot-clip videos that move around an object of interest. Figure 4 shows an example of a video of a chair used as source data. From this video, a structure-from-motion technique is used to estimate camera motion and sparse point clouds [34, 35]. A dense-point cloud is then obtained using a multi-view stereo algorithm that reconstructs the details of the captured scene and the target object [4]. For reconstruction, the method samples 14 to 28 video frames from the input video at uniform intervals to reduce computational costs. Next, the method computes 2D projections of the point cloud on the sampled video frames and shown in the interface (Figure 2 (D)). The planar surface on which the object is placed (e.g. a table or the floor) is further estimated by a RANSAC algorithm [45]. The planar surface can also be manually defined by selecting a set of points in the scene.

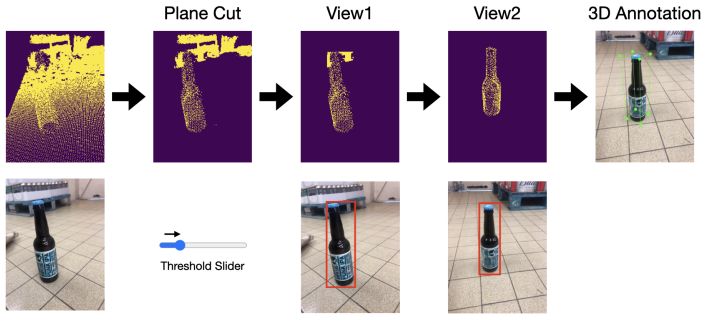

Fig. 3. A process of 3D annotation with the proposed interface

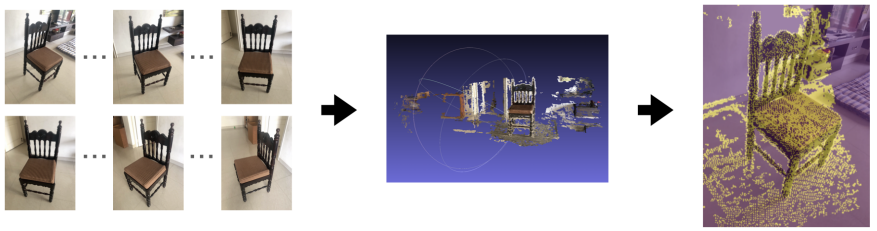

Fig. 4. Preprocessing pipeline: left) input video frames, middle) reconstructed points from structure-from-motion and multi-view stereo, and right) projected points onto a video frame.

3.3 3D annotation tools

Plane cutting. Automatic plane estimation cannot completely eliminate noise and outlier points, as shown in Figure 5. Therefore, we include a plane-cutting tool to interactively discard points belonging to the support plane from the input point cloud. The interface allows the user to change the (height) threshold of plane cutting, where modifications are dynamically reflected in the point cloud view. Properly filtering out the support plane allows the points belonging to the target object to be isolated for easier selection.

Object point extraction from 2D input. After removing points of the support plane, the user can start defining which points belong to the captured object with 2D rectangle selections. The interface provides two buttons for such selections to include or exclude points from the object (Figure 6). To completely remove background objects and walls, the user may need to create more than two rectangles from different views. There may further be non-target or noise points also remaining after the background is removed, which can adversely affect 3D bounding box estimation. This can be remedied by removing points inside the bounding box.

3D bounding box estimation. The interface provides two algorithms for tightly fitting a 3D bounding box from the object's 3D points in the width and depth directions. The algorithms assume that the object is placed on a support plane like in the Objectron dataset [1], so the bottom face of the 3D bounding box is always facing the plane. The first algorithm is the minimum bounding box algorithm that finds the cuboid with the smallest volume encompassing the object's points (Figure 7 top). This algorithm is applicable to most types of object shapes, such as cubic objects, cylinders, chairs etc. The second algorithm finds the longest edge of object points and uses it as the long edge

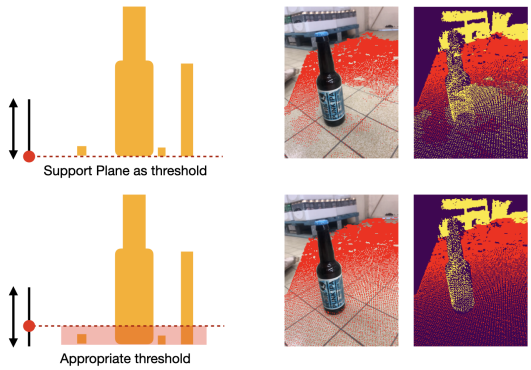

Fig. 5. Plane cutting: plane cutting with original plane height (top), and plane cutting with appropriate threshold (bottom). Red points and areas denote support planes to be removed.

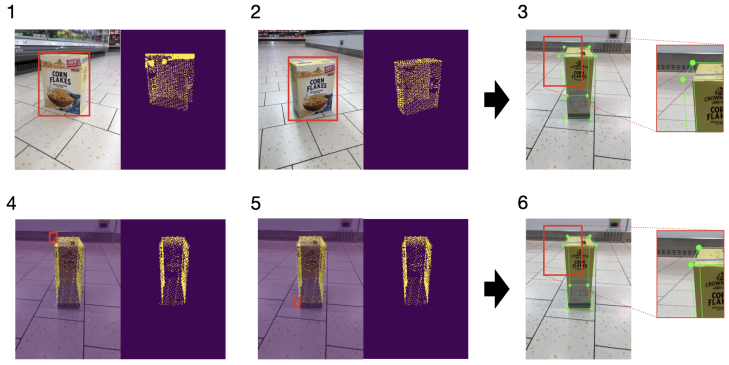

Fig. 6. Shaping a point cloud with 2D rectangle selections: 1 and 2) removing points outside bounding boxes, 3) computed bounding box with noise points, 4 and 5) removing points inside bounding boxes to remove noise, 6) fixed bounding box.

of a cuboid. We use this method if the results of the first algorithm are not satisfactory (Figure 7 bottom). In the height direction, the bottom face of the bounding box corresponds to the support plane and its parallel top face is defined so as to include the highest point of the object. The user can specify the front face of the bounding box by successively clicking two diagonal corners of the face.

3.4 User Interface Implementation

We implemented our tool as a web-based application with the front-end developed using p5.js [2] and the backend server created with the Flask web framework. We used Open3D [45] for point-cloud processing (e.g., plane estimation).

The web application can output annotation data in 3D and 2D formats, which include camera position and corner positions of detected 3D bounding boxes for each frame. 2D annotation data

[2]https://p5js.org/ (Accessed Sep 8, 2023)

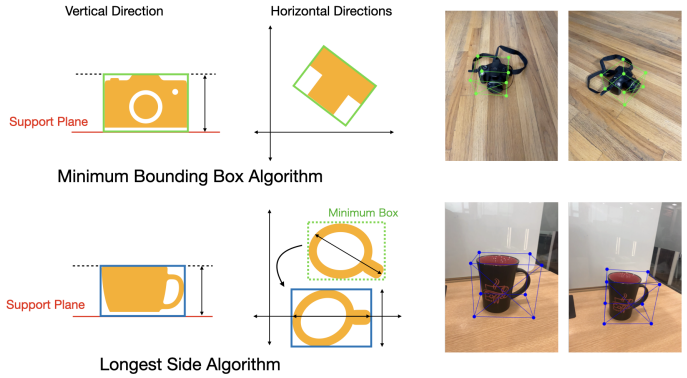

Fig. 7. Two algorithms to estimate the 3D bounding box: minimum bounding box (top), and longest edge (bottom).

includes reprojections of the 3D bounding box corners on the frames and binary masks of the regions covered by the bounding boxes in PNG format. This annotation data can be used for several applications, of which we give a few examples further below.

4 USER STUDY

To evaluate how accurately bounding box annotation data can be created using our tool, we conducted a user study comparing our (**proposed**) technique to a publicly available 3D annotation tool (**baseline**). We hypothesized that our **proposed** tool would produce 1) more accurate annotations of desired objects with 2) less annotation variance among participants compared to the **baseline** tool. We also sought qualitative feedback from participants to identify shortcomings of the **proposed** interface and better understand the needs for 3D annotation of video data.

We considered a scenario where machine learning developers and annotators would like to create a dataset to build a model for 3D object pose recognition. We recruited 12 participants (Female: 2, and Male: 10; average age 33.41 (SD: 5.62)) following our institution's code of ethics. 10 participants had a background in computer science (CS) education. The others had no CS background but were familiar with basic PC tasks and office work, which include creating documents, figures and charts. Seven of the participants with a CS background further had previous experience with annotation for machine learning.

4.1 Task and Data Preparation

Our task consisted in annotating videos (creating 3D bounding boxes) for 8 categories of objects (Figure 8) in the Objectron dataset [1]. This dataset contains videos for 9 object categories: bikes, books, bottles, cameras, cereal boxes, chairs, cups, laptops, and shoes. We prepared two sets of annotation targets: A and B, with each set containing 8 videos from each of the last 8 object categories, i.e. we omitted "bike" videos because most of them had mosaic processing to preserve privacy. For the "shoe" category, participants were requested to label the right shoe.

We sampled video frames (mean: 21.56) from the videos at uniform intervals and used COLMAP's structure-from-motion method [34, 35] followed by OpenMVS [4] to obtain a dense point cloud of the scene. We then ran the RANSAC-based support plane estimation algorithm of Open3D [45] to find the object's support plane. The support plane can be obtained by manually selecting three of

Fig. 8. 8 object types used in our user study from the Objectron dataset [1]

its points, but we used the results of automatic estimation for our study. Note that both **proposed** and **baseline** require support plane information.

4.2 Baseline

As the **baseline** 3D annotation technique, we employed labelCloud, a recently released tool [31, 32] which includes several 3D operations to manually place and adjust 3D bounding boxes on point cloud data. While previous studies typically positioned 3D bounding boxes on sparsely reconstructed surfaces (Objectron, and LabelAR), our proposed method works with densely reconstructed space using MVS, so we use dense clouds for our evaluation, which labelCloud also supports. A further reason for choosing this tool as a baseline is its availability to the public, contrary to the annotation tools used for Objectron and LabelAR, which have not been made available.

LabelCloud mainly provides two techniques to create bounding boxes; 1) selecting 3 points of a cuboid, and 2) adjusting the position, orientation, and size of a 3D box with successive mouse operations. The main differences between the **proposed** interface and labelCloud are threefold:

- **User Input**: The **proposed** method uses 2D rectangles to define object regions of projected point clouds. LabelCloud directly uses 3D bounding boxes in 3D space.
- **View Navigation**: With the **proposed** interface, frames are selected with a cursor on a timeline similar to a media player. With labelCloud, users navigate the 3D scene with the point cloud using 6-DoF controls like in 3D modelling and viewing software (e.g., Blender).
- **Bounding Box Computation**: The **proposed** interface computes bounding boxes based on users' 2D rectangular selections. In labelCloud, the 3D bounding box is directly defined and adjusted by the user.

4.3 Procedure

Participants used both the **proposed** and **baseline** tools to create 3D bounding boxes with sets A and B. We used a "within-subjects" design for the proposed and baseline conditions so that our participants with different backgrounds and familiarity with annotation tools could experience both techniques. This approach helped mitigate potential biases stemming from their differing levels of expertise. Given the distinct interaction designs of the two UIs, we anticipated minimal carryover effects when switching to the second technique. However, recognizing the higher potential of learning effects with the objects to be labelled, we designated the datasets used for the annotation task as a "between-subjects" factor. To ensure consistent annotation difficulty, we utilized identical

object types across the datasets. Both sets were to be annotated with an equal number of annotations for the **proposed** and **baseline** conditions.

The order of objects to be annotated in a set was always the same for all conditions (left to right of Figure 8). In all conditions, participants first practised to learn the use of the interfaces with example data for about four minutes. They then started the main task with the two tools, one after the other. We instructed participants to place and tightly fit a 3D bounding box to the target object as precisely as possible. We also asked them to specify the front face of target objects. We followed the Objectron dataset for front face orientations. Participants were allowed to ask for help about the operation of the two tools while performing the tasks. After completing all tasks, participants filled out a questionnaire about their experience and provided further oral feedback in an interview.

4.4 Evaluation measures

The evaluation of annotation quality typically involves measuring the consistency of annotations produced by multiple human annotators (inter-annotator agreement) [5, 11, 17], where an additional independent measure can be included to ensure that agreement is not merely coincidental [37]. We therefore used the following two metrics to evaluate the annotations produced by the proposed and baseline tools.

The first metric is 3D Intersection over Union (IoU) between participant and ground truth annotations, which addresses annotation precision and (non-)bias criteria. 3D IoU is defined from 0 to 1, with 0 representing no overlap and 1 representing complete overlap, i.e., higher values indicate higher accuracy. The ground-truth data creation procedure involved two authors reaching a consensus on object annotations following the guidelines described in Section 4.3, which require annotations to align with the object's direction and be as close as possible to the object. The process was iterative: an author initially created a bounding box using our tool and refined it using labelCloud. The two authors then reviewed the annotations in 2D and 3D viewers and each independently determined whether the annotations adequately followed the guidelines. A consensus was achieved when both authors agreed that requirements were met. If not, the bounding box was adjusted by its author until a consensus was reached.

The second metric is the ratios of overlapping regions between the different 3D annotations created by participants, which addresses inter-annotator agreement. The assumption is that if an annotation is precise, then annotations for the same object by multiple annotators should largely overlap. The Objectron dataset contains 3D bounding box annotations, with some inter-user variability for the 3D annotations, as reported by the authors [1]. We consider annotation variance between participants for the **proposed** and **baseline** interfaces. We also used 3D IoU to determine overlapping region ratios between 3D bounding boxes. In this case, higher values indicate lower annotation variance among participants.

As explained above, we conducted a mixed design study, where the interface, **proposed** and **baseline**, was a within-subjects factor, whereas the used dataset, A or B, was a between-subjects factor. For each condition, we obtained annotations from six participants on each dataset. With each dataset containing eight objects, this results in 96 annotations for the **proposed** and **baseline** interfaces. For the accuracy metric, we computed 3D IoUs for all 96 annotations. For the overlapping regions metric, we calculated 3D IoUs for all combinations of the six participants' annotations in both datasets. We thus obtained 30 IoUs for each of the 8 objects, i.e. a total of 240 samples.

We also recorded task completion times defined as the time from the start of the annotation to the time the participant declared it complete. We obtained 96 completion time for each interface.

In the post-experiment questionnaire participants were asked to answer 4 questions related to their experience with each tool on a 7-point Likert scale (ranging from 1: strongly disagree, 4: neutral, to 7: strongly agree): Q1) I was satisfied with the annotation results, Q2) I was able to

Table 1. Results of accuracy, overlapping region ratios (3D IoU), and completion times for the annotations. In the average column, * ($p < 0.01$) indicates the significance within the conditions revealed by the Mann–Whitney U test.

Object	mean accuracy (SD)		mean overlap region ratio (SD)		mean completion time (SD)	
	proposed	baseline	proposed	baseline	proposed	baseline
book	**0.87** (0.03)	0.81 (0.07)	**0.95** (0.03)	0.77 (0.10)	**104.2** (56.4)	129.1 (89.2)
bottle	**0.80** (0.03)	0.75 (0.07)	**0.88** (0.09)	0.71 (0.08)	**94.5** (52.0)	139.8 (75.1)
cereal	**0.90** (0.03)	0.85 (0.08)	**0.96** (0.02)	0.85 (0.05)	**88.9** (37.5)	119.0 (35.8)
chair	**0.93** (0.05)	0.85 (0.05)	**0.95** (0.05)	0.85 (0.06)	102.0 (44.7)	**85.4** (34.9)
cup	**0.87** (0.04)	0.78 (0.10)	**0.97** (0.04)	0.72 (0.11)	**68.7** (25.3)	119.5 (47.4)
shoe	**0.89** (0.03)	0.88 (0.06)	**0.89** (0.06)	0.85 (0.06)	**78.3** (37.6)	89.7 (26.2)
laptop	0.80 (0.03)	**0.81** (0.07)	**0.89** (0.05)	0.80 (0.06)	280.8 (159.4)	**75.1** (23.8)
camera	**0.83** (0.07)	0.82 (0.07)	**0.88** (0.08)	0.78 (0.06)	159.5 (84.4)	**80.0** (25.0)
average	**0.86*** (0.06)	0.82 (0.08)	**0.92*** (0.07)	0.79 (0.10)	122.1 (98.6)	**104.7** (43.9)

create annotations quickly, Q3) I was able to place bounding boxes precisely, Q4) The tool was easy to use. Note that Q2 asks participants' subjective perception of their task efficiency for each condition. Finally, interviews were conducted, where participants were asked about the pros and cons of each tool as well as suggestions for possible improvements.

Statistical tests were performed on the collected evaluation results. We also adjusted the results of p-values using the Benjamini–Hochberg procedure for multiple comparisons.

4.5 Results

Table 1 shows the results of the mean overlap region ratio (3D IoU) for each object and its average.

We confirm that the observed difference is statistically significant using a Mann–Whitney U test (the Anderson-Darling test did not reveal normality for the results of both interfaces) with $p < 0.0001$ ($r = 0.17$). Figure 9 (right) shows the distributions of the accuracy results.

Table 1 also shows the results of the mean overlap region ratio (3D IoU) for each object and its average. For all object types, the **proposed** interface outperforms the **baseline**. We confirm that the observed difference is statistically significant using a Mann–Whitney U test (the Anderson-Darling test did not reveal normality for the results of both interfaces) with $p < 0.00001$ ($r = 0.11$). Figure 9 (right) shows the distributions of overlap region ratios.

The results of mean and median task completion times are similar. The mean values for the **proposed** and **baseline** interfaces are 121.02 (SD: 86.86) and 109.49 (SD: 51.28) seconds, respectively. The median values for the **proposed** and **baseline** interface are respectively 92.5 and 96.5 seconds. As the Anderson-Darling test did not confirm normality for the results of the **proposed** interface, we used the Mann–Whitney U test to determine significance. The test did not show a statistically significant difference in the results of task completion time ($p = 0.300$).

Figure 10 shows the results of participants' responses to the four questionnaire questions. The results of Q1, Q2, and Q3 are similar for both interfaces. For Q4 (Easy to use), the median values of the **proposed** and **baseline** interfaces are 5 and 2, respectively. However, a Wilcoxon signed-rank test did not reveal significant differences in all questions.

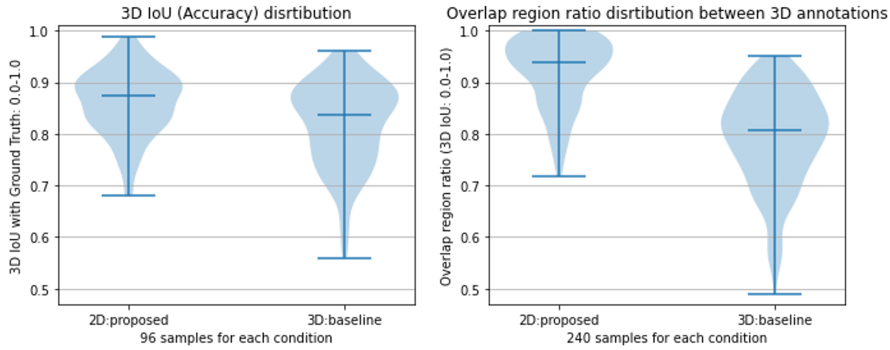

Fig. 9. Violin plots of results: Distributions of accuracy (left), and overlapping region ratios for the annotations (right). The three horizontal bars in each plot denote minimum, median, and maximum values respectively

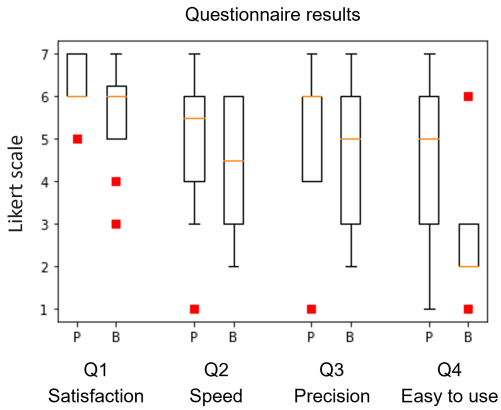

Fig. 10. Box plot for questionnaire results. Red dots show outliers.

4.6 User Feedback

Operability. Several participants stated that the proposed interface was easier to operate than labelCloud[3]: *"(For the 2D interface) I felt it was easy to check if the annotation was done well. I was confident about the result"; "With the 2D interface, the operations were easy"; "the 2D interface was good and intuitive"; and "Pattern A could be displayed by cutting the outside of the rectangle, so it was easy to check whether it was done well. I was confident that it was properly done."*.

Challenges of annotating specific objects. Several participants mentioned challenges of annotating some of the objects: *"With the 2D interface, annotating laptop and camera objects was difficult for me while other objects were easy"; "It was hard to include the camera strap and points behind the screen in dataset A (proposed)"*.

Negative effects of limited viewpoints. Several participants mentioned the negative effects of the limited viewpoint of the proposed method: *"Pattern A (proposed) was not captured from all directions of the object, so it was sometimes difficult to check that the point cloud corresponded to the object"; "The limited viewpoint meant that it was not possible to make the object's straight line parallel to the*

[3]Participant quotes are translated from Japanese, language in which the study was conducted.

window of deletion and delete the point cloud alongside the object"; "The 2D version was stressful, because I couldn't view the object from angles I wanted"; and "I felt that it was easier to improve the accuracy with B(baseline) than with A (proposed) because there is more freedom to look from different viewpoints".

Confidence of 3D annotations. Participants mentioned an advantage of the baseline to check results of annotations in 3D viewpoints: *"It was easy to see the 3D position of the bounding box with the 3D interface"; and "In contrast to the 3D interface, annotations of the 2D interface were not so clear based on what they looked like and whether they were correct or not. It would have been nice to have a view that can be displayed in 3D and rotated".*

4.7 Discussion

The result of the user study confirms our hypothesis that 3D bounding boxes created with our tool resulted in higher accuracy with respect to ground truth data and lower annotation variances between participants compared to the **baseline** interface. Figure 11 shows objects and their annotations using the proposed and baseline interfaces from 6 different participants [4]. The proposed interface uses **bounding box computation** algorithms based on extracted dense point clouds. With the proposed method, participants enclosed points of objects to label using 2D rectangles instead of manually placing a 3D box annotation.

Fig. 11. Annotation variances: each lines of cuboids show 3D annotation from 6 participants. Upper) results of the baseline and bottom) results of the proposed interface.

We observe that the difference in accuracy between the **proposed** and **baseline** methods is not as pronounced as the overlap regions. Moreover, there was a discrepancy between the results of accuracy and overlap ratios for objects like cups in the **proposed** method. This represents cases where, despite a high level of agreement among participants, the accuracy was not high, as shown in Figure 12. This is because the reconstruction of objects captured from limited viewpoints may be incomplete. To address this, it is important to ensure objects are captured from multiple different angles, or in the case of videos, where this is not possible, correct imperfect annotations post capture, similar to how the ground truth data was created. Our tool could possibly be extended to prompt the user to check produced annotations when it detects that object capturing angles are too narrow in the source data.

[4]We utilized two datasets, A and B. Every participant among the 12 participants for one of these datasets using the baseline and proposed methods. Consequently, the annotated results from a particular method on one dataset would be from six participants.

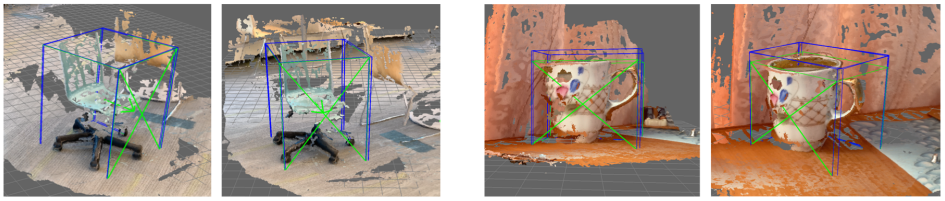

Fig. 12. Ground truth (green) and six annotations by the proposed interface (blue). Left: annotations of a chair have high agreement level among participants and high accuracy with the ground truth. Right: annotations of a cup have high agreements among participants, but offsets between annotations and the ground truth exist.

Participants' feedback highlighted the benefits of our tool with its 2D-based manipulations compared to labelCloud and its 3D interface. With our tool, the **user inputs** are simply 1) adjusting the plane cutting threshold and 2) defining 2D bounding boxes from different views. In contrast, labelCloud requires users to place 3D bounding boxes, and then repeatedly adjust their position and orientation.

We also observed cases where manipulations with our tool were more time-consuming, as shown in Table 1. Specifically, reconstructed point clouds of laptop videos in the study contained noisy points due to reflections of the laptop screen. Participants thus needed to carefully find and remove those noisy points in addition to the rectangle selection operations. Furthermore, participants reported difficulties with our tool to remove noisy points that were hard to see due to viewpoint constraints. In contrast, the 6-DoF **view navigation** of labelCloud has the benefit of allowing visual confirmation of object points from arbitrary perspectives. A simple improvement to our application would therefore be to add a full 3D view with 6-DoF view navigation to the 2D rectangle interface in order to handle those edge cases.

5 LIMITATIONS AND FUTURE WORK

While our proposed tool produces more consistent annotations for most videos, it relies on objects to annotate having sufficient perspective coverage in the input data, which may not always be the case. In the Objectron dataset, there are several video clips where objects are captured from limited viewpoints. In such cases, the object may only be partially reconstructed due to occlusions, with points at the back missing. As a result, the computed 3D bounding box using our method may not enclose the entire volume of the original object. Furthermore, the Objectron dataset includes video clips with mosaic content to preserve privacy, which may affect object reconstruction with structure-from-motion and multi-view stereo. To overcome those limitation, we plan to investigate point cloud interpolation methods for reconstructed points [26, 43] that help restore object volumes.

To circumvent the constraint of limited viewpoints with 2D frames from an RGB camera, objects of interest could be captured with a handheld depth camera, which would allow 3D point clouds to be produced in real-time and of higher quality, so that annotations could be created directly at capturing time. LabelAR [16], for instance, produces annotations in real-time using camera self-positioning.

A limitation of our user study is the biased participant distribution. In the study, we recruited 10 of 12 people with a computer science background. In addition, there were only two female participants. While video annotation might often be conducted by individuals without a computer science background, diverse users could produce different outcomes when interfacing with either

the proposed or baseline systems. However, this limited participant diversity of our study hinders a comprehensive analysis of potential background-based differences.

Another limitation is our gauge of participants' subjective experiences through our questionnaire. The task completion times varied depending on the object types, leading to potential discrepancies in participants' perceptions of subjective speeds, as shown in Table 1. Additionally, the effects of long-term usage for either tool were not explored, leaving the level of achievable efficiency and performance with prolonged use uncertain.

For future research, it would be invaluable to conduct a more in-depth study with participants from varied backgrounds (especially no computer science background). This would allow a better understanding of their annotation performance and subjective experiences with dedicated questionnaires (e.g., User Experience Questionnaire [18, 33]).

6 APPLICATION SCENARIOS

We show application scenarios, which use the labelled data produced by our proposed interface (i.e., 3D bounding box annotations or masks).

6.1 Dataset Creation for 3D Object Recognition

One of the main goals of our tool is to support the creation of object detection datasets like Objectron [1] and OnePose [36]. Such datasets contain moving videos capturing a particular static object of interest and corresponding 3D annotations (bounding boxes). Annotation data for these videos include camera and 3D object poses (i.e. position and orientation) typically estimated by camera tracking techniques such as ARCore [5] and COLMAP [34, 35]. The proposed interface can be used to annotate any object in such moving videos and output 3D annotation data.

Figure 13 shows a sample dataset for 3D recognition of daily objects using the proposed interface. The dataset contains object categories that are not included in existing dataset such as Ojbectron and OnePose [1, 36], e.g. sanitizers, supplements, and speakers. The dataset can be used for 3D pose recognition of further object categories (Objectron [1]) and specific items (OnePose [36]).

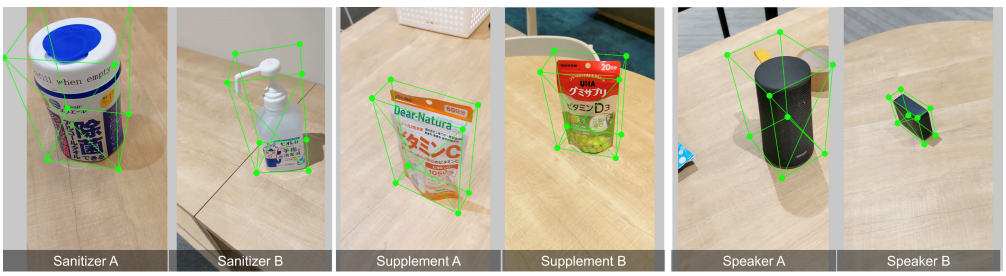

Fig. 13. Example results of 3D annotations for daily object recognition

6.2 Removing Objects from Videos and Reconstructed 3D Scenes

Editing reconstructed 3D spaces from moving videos is another key application of our tool. Recent advancements in computer vision make it possible to reconstruct 3D scenes from smartphone videos [22–24], where these scenes may sometimes contain objects that the user would like to remove. Deleting such objects requires a binary mask indicating the location of the object to be removed in each frame, which our tool helps to produce more efficiently.

[5] https://developers.google.com/ar/reference (Accessed Sep. 8, 2023)

Figure 14 shows an example of an outdoor scene reconstructed using Instant NGP [23]. The scene contains a triangle cone, which can be removed using the 2D object masks of projected 3D bounding boxes (Figure 14 (C)) created by our tool [6]. The object masks can also be applied to deep learning-based video inpainting applications [19, 27, 40] that remove specified objects in video frames and fill the gaps with with visually coherent background or context.

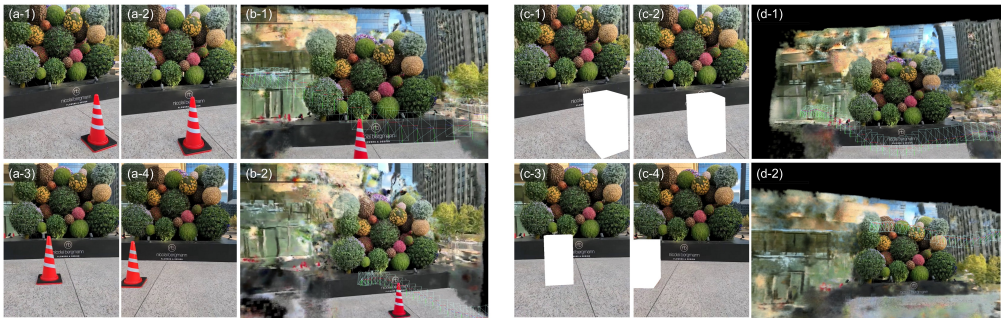

Fig. 14. Example of a reconstructed 3D scene containing an object to be removed. (A) Input video. (B) Reconstructed 3D scene [23]. (C) Video with object masks created with our tool. (D) Reconstructed scene with the object removed.

7 CONCLUSION

This work proposes a method to create 3D bounding box annotations of objects in RGB videos using shaping from generated dense point clouds and 2D selection rectangles in a few frames. We developed a user interface with functionality to support this process including plane cutting, shaping points from 2D rectangles, and 3D bounding box estimation from 3D object points. We performed a user evaluation to investigate the effectiveness and usability of the proposed interface compared to a publicly available annotation tool. Results showed that the proposed interface outperforms the baseline as measured by annotation accuracy and overlapping region ratios between user annotations. Based on those results and participant feedback, we discussed advantages and drawbacks of the proposed interface and presented suggestions for possible improvements. Finally, we illustrated how our proposed interfaces could be used in concrete application scenarios which rely on 3D annotations of objects in videos as input. We are considering publicly releasing our method as a plugin for existing annotation software in the future.

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] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. 2019. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2614–2623.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[4] Dan Cernea. 2020. OpenMVS: Multi-View Stereo Reconstruction Library. (2020). https://cdcseacave.github.io/openMVS

[5] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[6] An official implementation of Instant NGP accepts mask images as regions to be excluded from the reconstruction.

[6] CVAT.ai Corporation. 2023. Computer Vision Annotation Tool (CVAT) (v2.4.9). https://doi.org/10.5281/zenodo.8095553

[7] Robert DeBortoli, Li Fuxin, Ashish Kapoor, and Geoffrey A Hollinger. 2021. Adversarial training on point clouds for sim-to-real 3D object detection. *IEEE Robotics and Automation Letters* 6, 4 (2021), 6662–6669.

[8] Abhishek Dutta and Andrew Zisserman. 2019. The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 2276–2279. https://doi.org/10.1145/3343031.3350535

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

[10] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. 2020. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320* (2020).

[11] Keita Higuchi, Soichiro Matsuda, Rie Kamikubo, Takuya Enomoto, Yusuke Sugano, Junichi Yamamoto, and Yoichi Sato. 2018. Visualizing Gaze Direction to Support Video Coding of Social Attention for Children with Autism Spectrum Disorder. In *23rd International Conference on Intelligent User Interfaces* (Tokyo, Japan) *(IUI '18)*. Association for Computing Machinery, New York, NY, USA, 571–582. https://doi.org/10.1145/3172944.3172960

[12] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. 2021. SAIL-VOS 3D: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1418–1428.

[13] Mona Jalal, Josef B. Spjut, Ben Boudaoud, and Margrit Betke. 2019. SIDOD: A Synthetic Image Dataset for 3D Object Pose Recognition With Distractors. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 475–477.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. 2023. Segment Anything. *arXiv:2304.02643* (2023).

[15] Aybora Koksal, Kutalmis Gokalp Ince, and Aydin Alatan. 2020. Effect of annotation errors on drone detection with YOLOv3. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 1030–1031.

[16] 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* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 987–998. https://doi.org/10.1145/3332165.3347927

[17] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. 2014. Glance: Rapidly Coding Behavioral Video with the Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 551–562. https://doi.org/10.1145/2642918.2647367

[18] Bettina Laugwitz, Theo Held, and Martin Schrepp. 2008. Construction and Evaluation of a User Experience Questionnaire. In *HCI and Usability for Education and Work*, Andreas Holzinger (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–76.

[19] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. 2022. Towards An End-to-End Framework for Flow-Guided Video Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Tzuta Lin. 2015. LabelImg. https://github.com/heartexlabs/labelImg

[21] Jiaxin Ma, Yoshitaka Ushiku, and Miori Sagara. 2022. The Effect of Improving Annotation Quality on Object Detection Datasets: A Preliminary Study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4850–4859.

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[23] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Trans. Graph.* 41, 4, Article 102 (July 2022), 15 pages. https://doi.org/10.1145/3528223.3530127

[24] Oleg Muratov, Yury Slynko, Vitaly Chernov, Maria Lyubimtseva, Artem Shamsuarov, and Victor Bucha. 2016. 3DCapture: 3D Reconstruction for a Smartphone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 75–82.

[25] Jeffri Murrugarra-Llerena, Lucas N Kirsten, and Claudio R Jung. 2022. Can We Trust Bounding Box Annotations for Object Detection?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4813–4822.

[26] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 2003. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *2003 Shape Modeling International.* IEEE, 153–161.

[27] Hao Ouyang, Tengfei Wang, and Qifeng Chen. 2021. Internal Video Inpainting by Implicit Long-range Propagation. In *International Conference on Computer Vision (ICCV)*.

[28] Quang-Hieu Pham, Pierre Sevestre, Ramanpreet Singh Pahwa, Huijing Zhan, Chun Ho Pang, Yuda Chen, Armin Mustafa, Vijay Chandrasekhar, and Jie Lin. 2020. A 3D dataset: Towards autonomous driving in challenging environments. In *Proc. of The International Conference in Robotics and Automation (ICRA)*. IEEE, 2267–2273.

[29] 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 *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 41, 14 pages. https://doi.org/10.1145/3526113.3545663

[30] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10912–10922.

[31] Christoph Sager, Patrick Zschech, and Niklas Kuhl. 2022. labelCloud: A Lightweight Labeling Tool for Domain-Agnostic 3D Object Detection in Point Clouds. *Computer-Aided Design and Applications* 19, 6 (mar 2022), 1191–1206. https://doi.org/10.14733/cadaps.2022.1191-1206

[32] Christoph Sager, Patrick Zschech, and Niklas Kühl. 2021. labelCloud: A Lightweight Domain-Independent Labeling Tool for 3D Object Detection in Point Clouds. arXiv:2103.04970 [cs.CV]

[33] Andrea Schankin, Matthias Budde, Till Riedel, and Michael Beigl. 2022. Psychometric Properties of the User Experience Questionnaire (UEQ). In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 466, 11 pages. https://doi.org/10.1145/3491102.3502098

[34] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*.

[36] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. 2022. OnePose: One-Shot Object Pose Estimation without CAD Models. *CVPR* (2022).

[37] Theophanis Tsandilas. 2018. Fallacies of Agreement: A Critical Review of Consensus Assessment Methods for Gesture Elicitation. *ACM Trans. Comput.-Hum. Interact.* 25, 3, Article 18 (jun 2018), 49 pages. https://doi.org/10.1145/3182168

[38] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. 2019. RIO: 3D object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7658–7667.

[39] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*. IEEE, 75–82.

[40] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. 2019. Deep Flow-Guided Video Inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[41] Jenny Yuen, Bryan Russell, Ce Liu, and Antonio Torralba. 2009. LabelMe video: Building a video database with human annotations. In *2009 IEEE 12th International Conference on Computer Vision*. 1451–1458. https://doi.org/10.1109/ICCV.2009.5459289

[42] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 737–744.

[43] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3D point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1009–1018.

[44] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and Andrew J. Davison. 2021. iLabel: Interactive Neural Scene Labelling. *arXiv*.

[45] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847* (2018).

[46] Walter Zimmer, Akshay Rangesh, and Mohan Trivedi. 2019. 3d bat: A semi-automatic, web-based 3d annotation toolbox for full-surround, multi-modal data streams. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1816–1821.

Received 2023-07-01; accepted 2023-09-22