RoCap: A Robotic Data Collection Pipeline for the Pose Estimation of Appearance-Changing Objects

Jiahao "Nick" Li
UCLA HCI Research
Los Angeles, United States
ljhnick@g.ucla.edu

Toby Chong
TOEI Zukun Research
Tokyo, Japan

Zhongyi Zhou
University of Tokyo
Tokyo, Japan

Hironori Yoshida
Future University Hakodate
Hakodate, Japan

Koji Yatani
University of Tokyo
Tokyo, Japan

Xiang 'Anthony' Chen
UCLA HCI Research
Los Angeles, United States

Takeo Igarashi
University of Tokyo
Tokyo, Japan

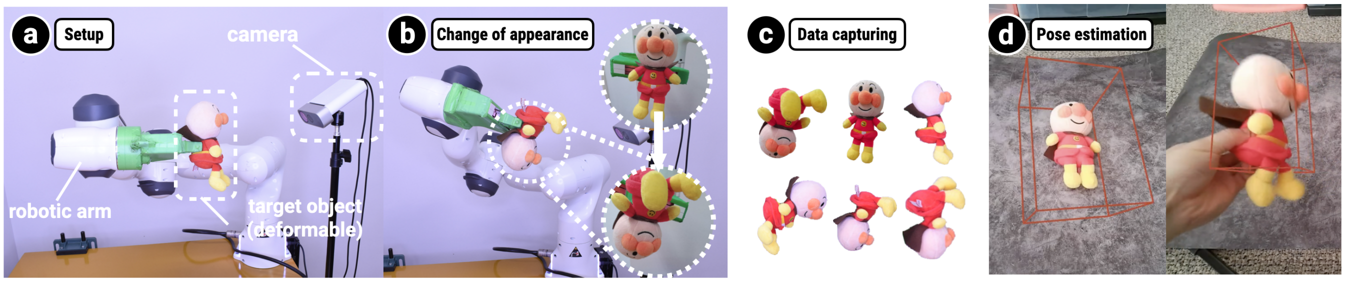

Figure 1: The RoCap pipeline is a robotic system designed to collect datasets for the purpose of pose estimation of appearance-changing objects, *e.g.*, a deformable plush toy (a). The system consists of a robotic arm and an RGB camera, which allows for data collection (c) of objects with appearance-changing features (b). Through data augmentation and training on off-the-shelf deep learning models using the collected data, the system can effectively estimate the pose of the plush toy during manipulation, even as it transitions through deformation (d).

ABSTRACT

Object pose estimation plays a vital role in mixed reality interactions when user manipulate tangible objects as controllers. Traditional vision-based object pose estimation methods leverage 3D reconstruction to synthesize training data. However, these methods are designed for static objects with diffuse colors and do not work well for objects that change their appearance during manipulation, such as deformable objects like plush toys, transparent objects like chemical flasks, reflective objects like metal pitcher, and articulated objects like scissors. To address this limitation, we propose RoCap, a robotic pipeline that emulates human manipulation of target objects while generating data labeled with ground truth pose information. The user first gives the target object to a robotic arm, and the system captures many pictures of the object in various 6D configurations. The system trains a model by using captured images and their ground truth pose information automatically calculated from the joint angles of the robotic arm. We showcase pose estimation for appearance-changing objects by training simple deep-learning models using the collected data and comparing the results with a model trained with synthetic data based on 3D reconstruction via quantitative and qualitative evaluation. The findings underscore the promising capabilities of RoCap.

KEYWORDS

datasets, pose estimation, interaction, mixed reality, deep learning

ACM Reference Format:
Jiahao "Nick" Li, Toby Chong, Zhongyi Zhou, Hironori Yoshida, Koji Yatani, Xiang 'Anthony' Chen, and Takeo Igarashi. 2018. RoCap: A Robotic Data Collection Pipeline for the Pose Estimation of Appearance-Changing Objects. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Leveraging existing tangible objects as controllers in mixed reality (MR) can significantly enhance the immersive experience and allow for a wider range of tangible motions, particularly for applications such as storytelling, skill training, and education. For example, employing plush toys to guide storytelling allows for a personalized and engaging experience, and using various handheld tools can facilitate more precise and diverse interaction mechanisms to make tasks feel natural and intuitive. Such an approach demands the capabilities for accurate predictions of 6D pose estimation — identifying an object's location and orientation in the 3D space.

Vision-based pose estimation has gained popularity in the past few years over tracker or sensor based methods, as it does not require additional hardware, alter the appearance or interfere with the normal use of the objects and it is cost effective and accessible. Researchers have adopted different approaches including mapping image feature to the 3D model of the object [20] and matching point cloud constructed by depth camera [2]. More recently, data-driven deep learning methods [38, 47] demonstrated accurate predictions of the 6D pose of pre-defined sets of object included in carefully crafted datasets [5, 26]. However, it remains unclear how well they work on objects where carefully labeled data do not exist such as personal objects. To address this issue, some prior work enables end-users to collect datasets for everyday objects 6D pose estimation [35, 39], introducing synthetic approaches to generate a large amount synthetic data given the 3D model of the objects [13], or adopts a few-shot learning method by training on 3D mesh reconstructed from a short clip of video [32].

A limitation of these existing methods is that they mainly focus on objects that are static objects with diffuse colors, with a less focus on objects that **change their appearances** when being manipulated, including objects with challenging appearance materials (*e.g.*, transparent and specular objects), deformable objects and articulated objects [45]. A pair of scissors will dramatically change its physical appearance due to mechanical operation and a model trained on the image of a closed pair of scissors might produce lower accuracy at recognizing the same pair in an open configuration. Similarly, a plush toy that changes its shape during manipulation when being affected by gravity will affect the performance of the pose estimation. While one intuitive approach is to capture data while a human user is manipulating the objects, annotating such data at scale would be costly and error-prone.

To address the challenge, we propose RoCap, an automated pipeline to collect image data of appearance-changing object for 6D pose estimation using a robotic arm with minimum human intervention. We deploy a robot arm to mimic human's hand to manipulate the objects while capturing the image data as shown in Figure 1. The 6D poses of the object of each image can be obtained with robotic forward kinematics as each joint of the robotic arm is precisely controlled. Specifically, RoCap performs the data capturing process for eight different appearance-changing objects with deformable, transparent, reflective and articulated properties (Figure 3).

We also implemented a simple pose estimation pipeline to quantitatively and qualitatively evaluate the pose estimation performance of the model trained on our collected data comparing against a few-shot learning pose estimation approach based on 3D reconstruction (Gen6D [32]). Both the quantitative and qualitative evaluation results demonstrate that existing work struggles with appearance-changing objects and our approach shows promise in overcoming these limitations with improved pose estimation accuracy.

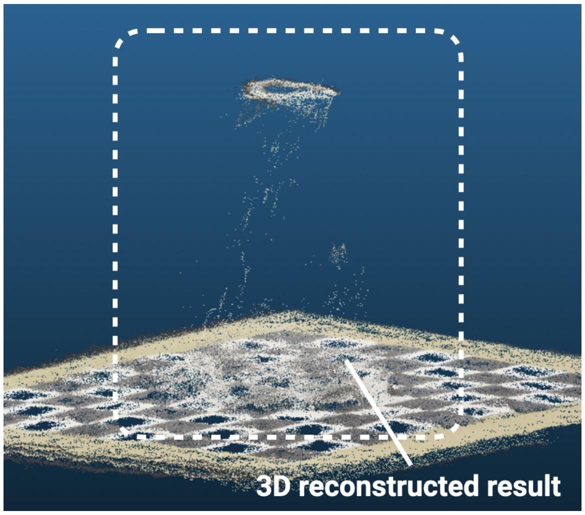

Figure 2: 3D reconstructed results for a transparent flask.

In summary, our contributions are two-fold:

- **A robotic data collection pipeline** with a 6 DoF robotic arm which captures and annotates 6D pose data for objects that change their appearance during manipulation, addressing limitations in existing data collection methods.
- **Quantitative and qualitative evaluations** to demonstrate the feasibility of the pipeline via improved accuracy of appearance-changing objects pose estimation by comparing with an advanced pose estimation method in the field of computer vision.

2 RELATED WORK

2.1 Object pose estimation

Object pose estimation plays a crucial role in various HCI applications such as augmented reality [2, 19, 20, 44] and robotics and automation [29]. Over recent decades, researchers have explored diverse approaches to predict an object's pose. These includes sensor applications like IMUs, physical marker techniques such as fiducial markers [17, 24, 46], optic trackers [49] 3D printed embedded QR code [12], computer vision techniques such as color-based tracking [44], feature point tracking [2] and point cloud alignment [20]. Recent advancement in deep learning has unlocked new challenging tasks such as predicting the poses of hand-object interaction [6, 18, 31], articulated objects [30] and other problem setups [1, 8, 33, 48]. Extending this line of research, RoCap focuses on a new problem setup where the objects will change their appearance during the manipulation. Note that RoCap does not contribute new model architecture or algorithm to improve the performance in the field of deep learning. Instead, RoCap contributes a novel data collection method and the data captured by the system can serve as

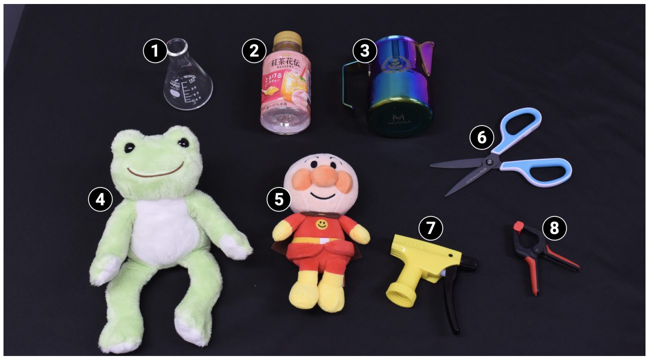

Figure 3: Example objects for each category that RoCap is focusing on, *Viewing-angle dependent*: (1) flask, (2) water bottle and, (3) pitcher, *Deformable*: (4) flexible frog and (5) stiff anpanman, *Articulated*: (6) scissors, (7) spray head and (8) clamp.

great resources for researchers in the community of computer vision and machine learning to solve the downstream tracking problems.

2.2 Pose data collection

Data-driven deep learning approaches require data annotated with ground truth labels. Yet, annotating 6D pose data is challenging, as it is hard to specify 3D bounding box on a 2D image. To address this, researchers have investigated various methods including three primary strategies: *(i)* training on synthesized data, *(ii)* utilizing publicly available datasets and *(iii)* designing interactive tools for data collection.

2.2.1 Synthetic data. One typical way is synthesizing data with the available resources such as the 3D model of the objects. This approach is commonly used in tasks such as object segmentation [40] and object detection [13]. And a standard way of using synthetic data in pose estimation is to obtain the 3D model and texture of the objects first and then render them with different target background [43]. Although synthetic data can be easily scaled, it comes with the drawback of a disparity between real and virtual data, which might impact model performance. Moreover, as illustrated in Figure 2, the necessary step of object reconstruction may fail for our target objects, such as the flask.

2.2.2 Real-world data. An intuitive way to bypass the issue of synthesized data is to collect data in the real world. In the recent years, researchers have adopted two major types of data collection methods. The first is **"static object + moving camera"**, where the pose of the object is calculated from the pose of the camera, which can be read from the embedded sensor. Normally it requires a certain level of human labor as first couple frames need to be manually annotated by matching the 3D model to the physical object. For example, several publically available datasets have been collected in this way for benchmarking in the pose estimation domain, such as YCB Video dataset [47], Linemod [3, 21] and T-Less [22]. Additionally, researchers have also developed interactive data collection pipeline to collect data on custom objects (*e.g.*, Label Fusion [35]. However, since the objects remain static, it is challenging to capture the appearance-changing features.

Another approach is **"moving objects + static or moving camera"**. While effective for capturing appearance-changing objects, this approach poses challenging for labeling ground truth. For instance, ARnnotate, used in augmented reality [39], requires users to hold and move the object along a recorded path, leading to potential errors, especially with objects like articulated items or deformed plush toys. RoCap adopts this approach and ensures the labeled ground truth to be precise by calculating the robotic arm's forward kinematics while it manipulates the object to capture the appearance-changing features.

3 APPEARANCE-CHANGING OBJECTS

In this section we define and explain the importance of three categories of appearance-changing objects that we aim to track using RoCap. We collected and captured eight items from the three categories with RoCap.

3.1 Deformation

Deformation refers to changes in the shape or size of an object due to external forces applied during manipulation (i.e., force of the hand and gravity). Objects with naturally deformable features can include soft and malleable objects such as fabric materials, clothing and plush toys/stuffed animals. During manipulation, the objects are affected by gravity all the time, leading to the deformation while the user is moving the objects into different orientation. We picked two plush toys of different stiffness, anpanman (stiffer) (Figure 3(5)) and frog (more flexible)(Figure 3(4)) as examples of the deformable objects.

3.2 Viewing-angle dependent

The visual appearance of viewing-angle dependent objects includes two main sub-categories of objects, transparent objects (e.g., glass) and reflective objects (e.g., polished metal). Appearance of transparent objects depends on the background behind them, which may contain the environment and the user's hands. Tracking and estimating the pose of such transparent objects is a known challenge [14] and hand manipulation may make this even harder. Appearance of reflective objects on the other hand depends on the environment in front and around it. We picked a conical flask and a plastic bottle(Figure 3(1, 2)) as representations of transparent objects of different level of translucency (Figure 7b). We also included a reflective pitcher to represent reflective object.

3.3 Articulated

Objects with articulated features refer to objects whose appearance changes through manual manipulation or interaction. These changes can occur due to the inherent function of the physical objects. For examples, various handheld tools will change their mechanical forms while being manipulated by human. We selected three manually-changing objects: a clamp, a pair of scissors and, a head of spray bottle to represent two different types of manual gripping and hand operation (holding and pinching).

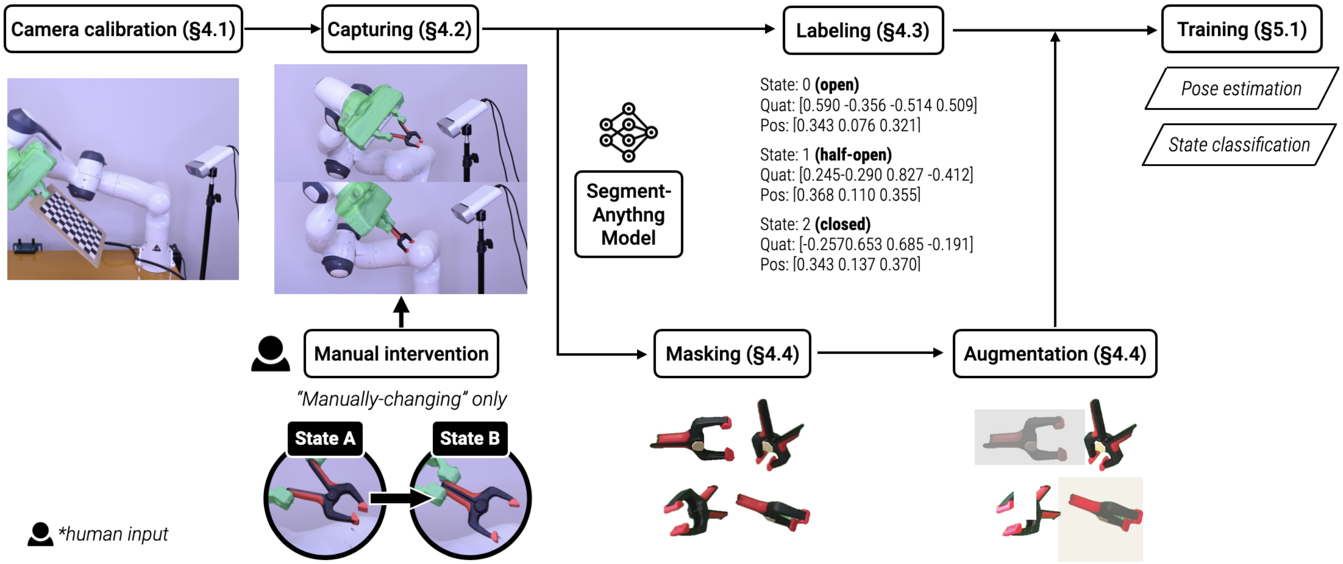

Figure 4: *Overview of RoCap.* RoCap pipeline consists of camera calibration (§4.1), data capturing (§4.2), data labeling (§4.3), data processing (§4.4) and data augmentation (§4.4). By training on an existing deep learning framework, RoCap achieves object segmentation, state classification and pose estimation for appearance-changing objects.

4 ROCAP PIPELINE

In this section, we will introduce the design of the RoCap pipeline, which is easily replicable using any 6-DoF robotic arm, we document the essential knowledge and technical challenges addressed including *(i)* camera calibration, *(ii)* data collection, *(iii)* data labeling and *(iv)* data pre-processing. Figure 4 shows the overview of the RoCap pipeline and we discuss each step in details as follows.

4.1 Eye-to-hand camera calibration

The first step of RoCap pipeline is to calibrate the camera to the robotic arm (Figure 4a). During data collection, the robotic arm will hold the target object using a gripper and the camera is standing on the side to capture the images. In this setup, the pose of the object in the image refers to the homogeneous transformation of the object from its reference frame to the camera's reference frame. This is a typical hand-eye calibration problem because as shown in Figure 5. Assuming the object has the same pose as the end-effector, the goal is to calculate the transformation matrix of the gripper to the camera: ${}^{c}T_g$, which can be calculated from the following equation:

$$ {}^{c}T_g \ = \ {}^{c}T_b \cdot {}^{b}T_g \tag{1} $$

In which ${}^{b}T_g$ refers to the transformation from the gripper to the base of the robotic arm which could be calculated by forward kinematics [10] while ${}^{c}T_b$ refers to the transformation from the base of the robotic arm to the camera frame, which is unknown.

To calculate ${}^{c}T_b$, a camera calibration step is required which can be accomplished by using a checkerboard with known size of the squares, which is illustrated in Figure 5b . By moving the robotic arm to multiple configuration, ${}^{c}T_g$ can be calculated from the following $AX = XB$ equations:

$$ \begin{gathered} {}^{g}T_b^{(1)}\ {}^{b}T_c\ {}^{c}T_t^{(1)} ={}^{g} T_b^{(2)}\ {}^{b}T_c\ {}^{c}T_t^{(2)} \\ \left({}^{g}T_b^{(2)}\right)^{-1}\ {}^{b}T_g^{(1)}\ {}^{g}T_c ={}^{g} T_c\ {}^{c}T_t^{(2)}\left({}^{c}T_t^{(1)}\right)^{-1} \\ A_iX = XB_i \end{gathered} \tag{2} $$

Here ${}^{c}T_t$ refers to the transformation from the checkerboard to the camera frame, which can be calculated knowing the size of the pattern [37]. Then the calibration target ${}^{c}T_b$ can be calculated from Eq. 1.

After the camera is calibrated, the next step is to capture the image data of the objects.

4.2 Data collection

As mentioned in the previous sections, RoCap collect data of the objects that exhibit the appearance-changing features. More specifically, RoCap collects objects categorized in four types of appearance-changing features: deformable, reflective, transparent and articulated.

4.2.1 Pose coverage. The goal of the capturing is simple: capture the images of the objects from as many angles as possible to have a good coverage of all the potential pose. Quaternions possess the advantage of representing each rotation without introducing any ambiguity. However, directly sampling quaternions proves to be a challenging task. To overcome this obstacle and achieve comprehensive coverage of poses, we opt for sampling Euler angles with a specific step of degrees for each yaw, pitch, and roll channel. Once we have obtained the Euler angles, they are converted into quaternions. These quaternions are then utilized to calculate the arc distance between each orientation. This methodology is employed due to the inherent

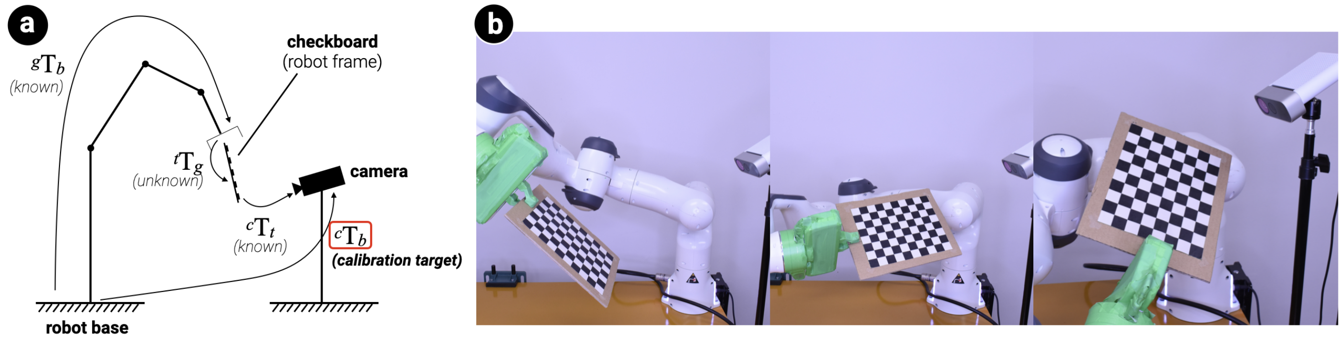

Figure 5: Illustration of the eye-to-hand camera calibration (a). The robotic arm grip a checkerboard and move to multiple positions and orientations for an accurate calibration (b).

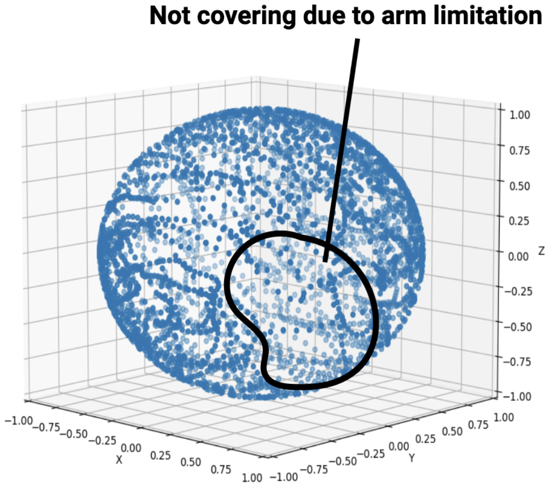

Figure 6: Pose coverage in RoCap capturing pipeline.

redundancies that can arise from sampling Euler angles. By computing the arc distance of quaternions, we effectively eliminate these redundancies. The threshold for eliminating redundancies is set at 0.35, roughly equivalent to a 20° azimuth angle.

However, due to the hardware limitation of the robotic arm (*e.g.*, the joints may have a limited range of motion), RoCap cannot cover the whole possible poses sampled in this process. We use the inverse kinematics solver and path planners in ROS and achieve the final sampling of the poses RoCap supports. Figure 6 visualizes the coverage of the poses in RoCap. Noted that in existing data collection method where the objects are placed on floor, there will be at least half of the poses not capturable because it is occluded by the contacting ground.

4.2.2 Capturing process. During the capturing, a human user will be required to hand the target object to the robotic arm and the robotic arm will move along the designed path and the camera capture the RGB images on each sampled point.

For deformable, reflective and transparent objects, there is no further actions from the users as the change of the appearance happens naturally when the object is oriented to different direction while being manipulated by the robotic arm 7abc. For articulated objects, actions need to be taken in order to change the mechanical states of the objects. The manually-changing action can be achieved either by human or the robotic arm automatically depending on the capability of the robotic arm to change the appearance. As is shown in Figure 1b, the size of the clamp is small enough to be grasped by the gripper. And the clamp is expected to have multiple states such as closed, open, and mid-open states. Without the help of human, the gripper could be able to change the states of the clamp by applying different forces on the parallel grippers. However, for the pair shown in Figure 7d, the robotic gripper is not able to automatically change the states because the handle is too wide for the parallel gripper when it is in open state. This is a typical robotic manipulability problem as mentioned in [29]. For the case that a robotic arm cannot establish firm gripping on the object, a human operator will be required to manually change the opening angle of the scissors in the interval between the capturing of different states.

4.3 Data labeling

The transformation from the base frame of the robotic arm to the target object is logged for each captured image in a 4×4 homogeneous transformation matrix. Then the transformation from camera frame to the object could be calculated using Equation 1. The rotation and the translation serve as the 6D pose label for the object in each image as shown in Figure 4c left.

4.4 Data processing and augmentation

After capturing the data with the ground truth labels of objects using RoCap, crucial processing step must be performed to facilitate subsequent pose estimation training. A typical object pose estimation task comprises two subtasks: *(i)* segmenting the object from the scene, and *(ii)* predicting the orientation of the segmented object. Therefore, the processing steps involves generating object masks for each label and augmenting the data to adapt to various environment in application.

4.4.1 Generating masks. RoCap leverages the recent emergence of Segment-Anything Model (SAM) [25] which is capable of producing high quality segmentations given points or bounding boxes

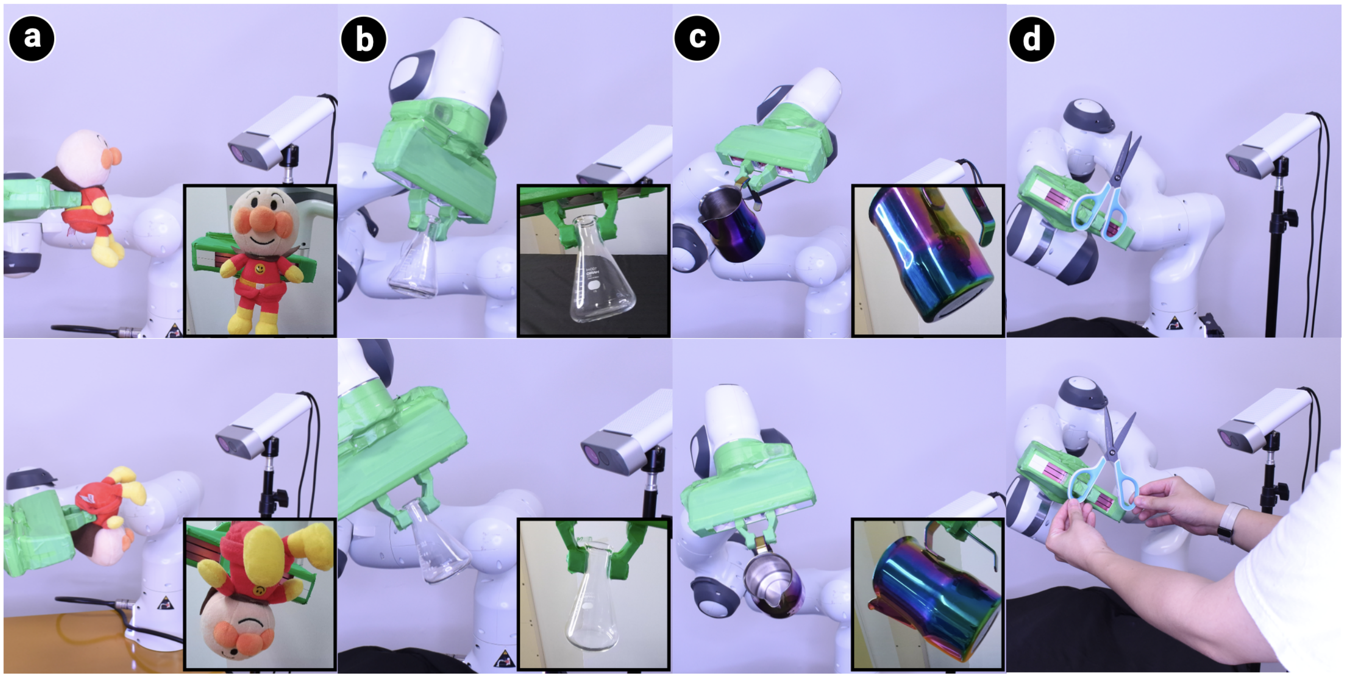

Figure 7: RoCap captures the appearance-changing feature of deformable objects (a), viewing-angle dependent objects including transparent objects (b) and reflective objects (c), and objects with articulated features (d). Human operator is needed if the robotic arm is not able to change the states automatically (d).

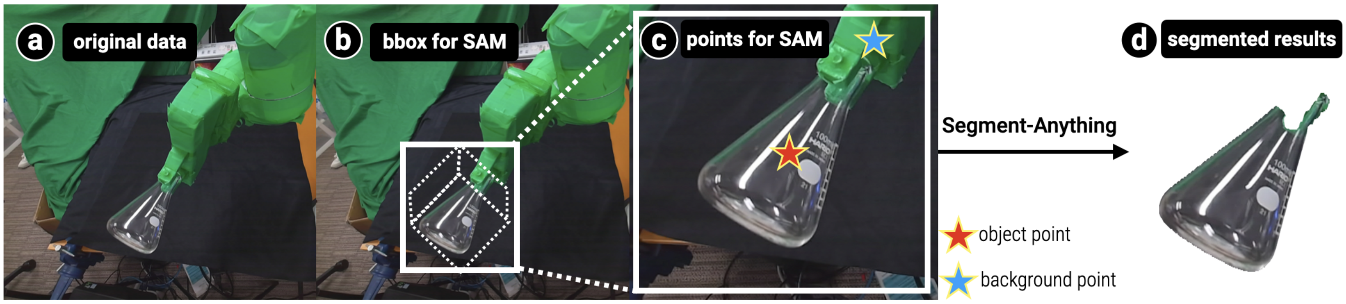

Figure 8: Data processing of data collected in by RoCap. RoCap generates mask for each image (d) by prompting SAM with bounding box (b) and points (c).

as prompts. For each image captured by RoCap, the subsequent procedures must be executed:

Bounding box. As the camera is calibrated to the robotic arm's coordinate frame, we generate the initial bounding box of the object by assuming the robotic arm is holding a 15x15x15 cm cube. We then project the cube's coordinates onto the camera's 2D plane to obtain the bounding box (Figure 8b). Generally, this method yields satisfactory masks for objects that are distinct and easily identifiable in the image. However, complications arise when objects are partially obscured by the robotic arm, difficult to distinguish (e.g., a flask whose appearance is influenced by the background), or even entirely invisible. To address these challenges, an additional filtering process is introduced. This process either segments the semi-occluded objects or discards the invisible data.

Filtering. To improve the quality of the masked objects, we leverages the interaction with SAM by providing additional prompts (points) to specify the objects and background (Figure 8c). Specifically, we incorporated two additional steps: *(i)* we provide additional prompts for the SAM to highlight the object's location and *(ii)* we wrap the gripper in green tape to reduce its potential interference with segmentation performance.

- **Providing additional prompts for the SAM to highlight the object's location**. Given that the gripper consistently holds the objects, we can infer that the center of the 15x15x15 cm cube corresponds to the object.Thus, we add the projected pixel coordinate of this center as a point prompt for the SAM, indicating the object's location.

- **Removing green background.** Given that the gripper can partially obscure the object, it might predominantly appear within the bounding box. This could lead the SAM to mistakenly segment the gripper as the target object. To counteract this, we detect the green regions in the image, which are presumed to represent the gripper. We then calculate the geometric center of these regions and use its coordinates to provide the SAM with a prompt, pointing out the undesired areas.

4.4.2 Data augmentation. We augment each masked image of the object with random exposure, contrast, saturation, etc. via Albumentations [4] to achieve better generalizability.

5 EVALUATION

To demonstrate the feasibility of our data collection pipeline, we conducted both quantitative and qualitative evaluation of the model trained on our data to compare with a few-shot learning pose estimation approach Gen6D [32]. Gen6D has shown competitive performance on any custom object by using a single video as input for 3D reconstruction (via COLMAP [41]) and performed feature matching based on the image and resulting pointcloud.

We evaluated the model in two settings, *controlled* setting where the ground truth can be reliably obtained for quantitative evaluation, and *application* setting, where the user manipulates the object during pose estimation for qualitative evaluation, as the ground truth pose cannot be obtained easily.

5.1 Implementation of pose estimation pipeline

Before we delve into the result of quantitative evaluation, We will discuss the pose estimation pipeline first. As mentioned earlier, the pose estimation pipeline should consists of one model for segmenting the target object and another for predicting the orientation based on the segmented output. For objects with manually-modifiable states (e.g., scissors), an additional state classifier is employed.

For the segmentation task, we leveraged a recent advancement based on SAM: HQTrack [50]. It is a zero-shot approach and requires no training while being able to consistently produce high-quality segmentation of target objects in videos.

For the orientation estimation, the model is a VGG16 model, pretrained on ImageNet [11], followed by a fully connected layer outputting the quaternion and the 2D pixel location of the object. The loss function is a combined loss of the Geodesic Loss on the quaternion prediction and the MSE Loss of the displacement prediction. We train the model on the augmented data for 120 epochs, using the Adam optimizer with a learning rate of 0.0001.

For the state classification, the model is a MobileNet V3 [23], pretrained on ImageNet [11], followed by a fully connected layer, and the output dimension is equivalent to the number of the states of the object. We train the model on the augmented data for 120 epochs, using the Adam optimizer with a learning rate of 0.0001.

5.2 Quantitative evaluation

For quantitative evaluation, we modified the environment by changing the camera angle and updating the background (Figure 9). Using a newly designed trajectory for the robotic arm, we sampled 1041 data entries. The accuracy threshold for pose estimation remained consistent with our training data parameters: set at 0.35 or an azimuth angle of 20°.

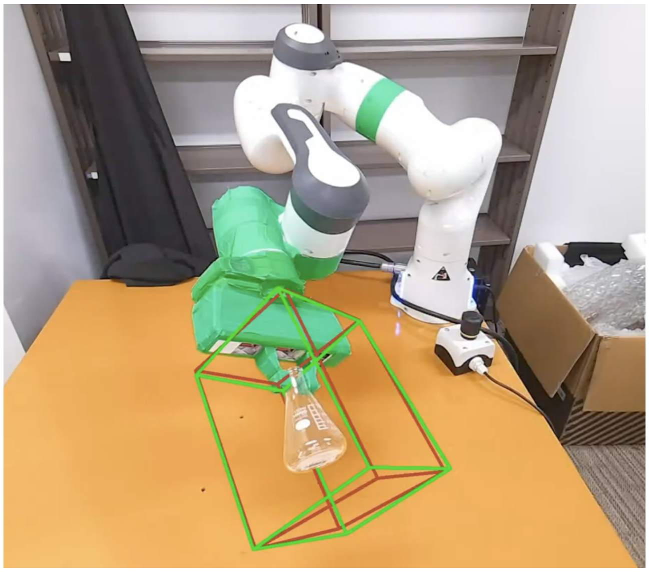

Figure 9: Quantitative evaluation setup. The green bounding box represents the ground truth and the read bounding box represents the predicted pose.

To clarify, our evaluation only focused on the accuracy of the orientation prediction, since RoCap rely on prior to determine the position of the object. For Gen6D, we could not modify its training pipeline to incorporate HQTrack to enhance its object detection. Instead, we adhered to the guidelines provided for pose estimation on custom objects as outlined in Gen6D's guidelines[1]. Specifically, we performed 3D reconstruction of the object using COLMAP [41] and followed the preprocessing procedure in the guideline.

For objects with multiple manual states, test data is gathered for each state, and the model is evaluated accordingly. The results represent the mean accuracy across all states. Specifically, the flask was tested against two different backgrounds: its original setting (a black background) and an alternate setting with a typical orange-colored desk surface. Table 1 shows the accuracy comparison between our approach and Gen6D.

We note that the accuracy is much lower in our testing result as compared to the result Gen6D demonstrated in their paper. This could be due to several factors:

- 3D reconstruction failures (*e.g.*, Figure 2).
- Data collection with objects in static positions, leading to challenges when the object's unseen side becomes visible during manipulation (*e.g.*, a plush toy might be placed face-up on a table during data collection).

[1] https://github.com/liuyuan-pal/Gen6D/blob/main/custom_object.md

	anpanman	frog	pitcher	flask	bottle	scissors	clamp	spray
RoCap	91.9	61.9	73.7	87.1(66.9)	71.9	83.4	42.0	87.6
Gen6D [32]	19.6	12.9	12.7	16.2	16.9	38.3	19.4	28.4

*The number in parentheses indicates flask accuracy in a different background (Figure 9).

Table 1: Quantitative evaluation result. The numbers indicate the average precision at 20° azimuth error.

- Gen6D's documented issue with size-changing objects in frames (as the object moves closer and further away from the camera), as mentioned in their GitHub issues[2].

The results indicate that a simple pose estimator trained with data from RoCap can deliver relative working pose estimation performance. However, the quantitative findings also reveal some limitations. For example, the accuracy of clamp is relatively low compared to other objects due to ambiguity caused by its symmetry. Additionally, objects whose appearances are environment-dependent demonstrate inconsistent performance under varying backgrounds. More details are discussed in the limitation in Sec. 6.1.

5.3 Qualitative evaluation

To test the performance in the application setting, due to the difficulty in collecting ground truth, we conducted a qualitative evaluation on videos of humans manipulating the objects. Figure 10 shows the qualitative comparison between model trained on RoCap data and Gen6D. For example, RoCap recorded both closed and open states during data collection for the pair of scissors. This allowed it to provide viable pose estimation for the open state (Gen6D which struggled with the unobserved state). Please refer to the supplementary materials for the video.

6 DISCUSSION

6.1 Limitations

The quantitative and qualitative evaluation has demonstrated the feasibility and potential of our data collection method. However, the result also shows certain limitations. Below, we will discuss the limitations from the perspectives of data capturing, model performance and other constraints.

Data capturing. While RoCap addresses the data capturing of appearance-changing objects, it requires the objects have distinct appearances in different defined poses. One typically example that is challenging for RoCap is cloth, which is highly deformable. Its extreme flexibility results in a loss of the pose information when being manipulated by the robotic arm. As illustrated in Figure 11, the piece of cloth is nearly identical in two different poses manipulated by the robotic arm.

On the other hand, as currently we target objects that people can easily change their appearances with hands, leading to the target object size ranging from 0.5x~1.5x of a palm size. Additionally, our robotic arm's mechanical gripper, with a maximum gripping width of 80mm, further constrains the size of objects it can handle. However, this limitation can be resolved when a system applies our method to a larger scale robotic arm (*e.g.*, in a mass manufacturing setting).

Model performance. Indicated in the evaluation results, the pose estimation pipeline does not handle symmetric object well. While this has been an open challenge in object pose estimation [45], recent work have proposed different network architecture to address this issue [42, 47]. While addressing symmetry is beyond the purview of this paper, future enhancements could incorporate a sophisticated pose estimator or gather supplementary data like depth via depth cameras.

Furthermore, variations in the environment from the capturing stage may also affect the model performance, especially for viewing-angle dependent objects. While it is feasible to maintain an environment similar to the capturing setup (*e.g.*, using a black background when operating a transparent flask), future improvements could include varying environmental factors such as different lighting conditions [9, 36]. Additionally, future work could introduce other augmentation method such as [51] to adapt to various environment.

Other constraints. Currently the need of a robotic arm may require a lab setting. However, the pipeline can also be applied to scenarios such as *(i)* product manufacturers collect data and train a model for their product, and include it as part of their solution package and *(ii)* home users asks their robot to train a model for their own object when robots are more accessible in the future, and later the user uses the model to estimation the pose in specific applications.

6.2 Handling Occlusion

Occlusion happens in different scenarios, including the objects being manipulated by hands during interaction, or the objects being held by the robotic arm during data capturing. While the hand occlusion does have an impact on the model performance trained on RoCap data, our pipeline is less impacted compared to Gen6D as shown in the qualitative results. This is due to the fact that during the capture phase, the robotic arm may partially obscure the object throughout the capturing process, which simulates hand occlusion in the training data.

To further address the occlusion problem, one possible approach is to introduce a hand-like robotic hand during the data capturing process. For example, anthropomorphic robotic hands, such as those presented in [15], can closely mimic human hand movements and provide more realistic interaction scenarios for data collection. By using a robotic hand, it is possible to better account for occlusions that occur during human-object interactions and develop models that can better predict user intent in such cases. Additionally, to address occlusion caused by the robotic arm during data capturing, multiple cameras could be employed to ensure the complete visibility of the objects being captured.

[2] https://github.com/liuyuan-pal/Gen6D/issues/29

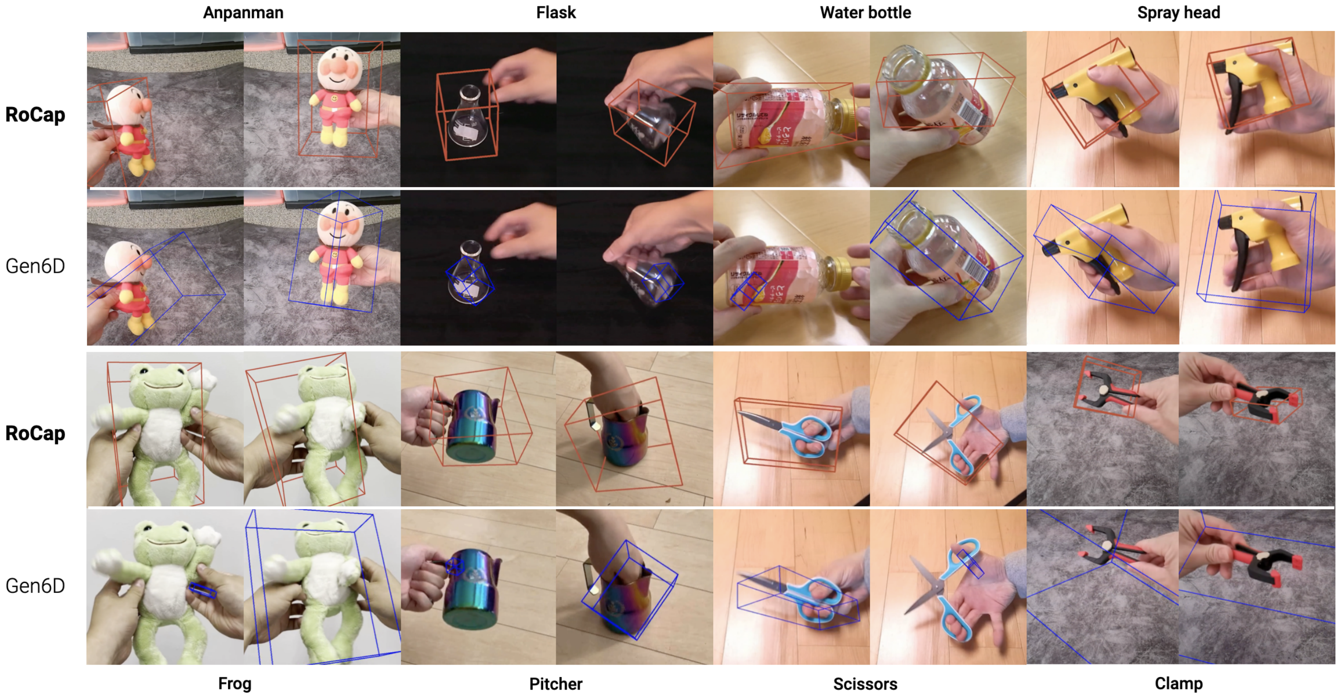

Figure 10: Qualitative evaluation of the eight objects.

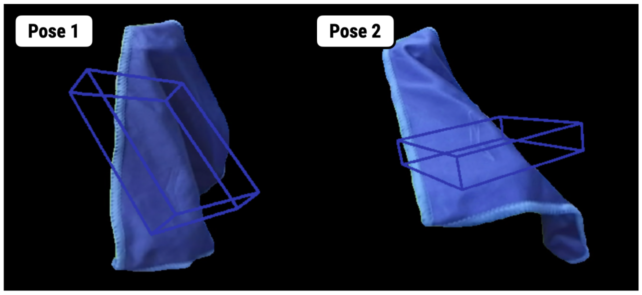

Figure 11: Failure case for a highly deformable cloth.

6.3 Automatic Changing of Mechanical States

As mentioned in the paper, certain articulated objects necessitate human intervention to change their states, as they cannot be manipulated by the parallel gripper of the robotic arm [29]. Examples of such objects include those that require a large range of motion or those that demand bi-manual operation. Recent research in HCI has proposed different methods of attaching mechanisms to the physical object to automatically actuate the motion without human intervention [27–29], which can be potentially leveraged by future data collection system using robotic arms to automatically collect a large amount of data. By automating the data collection process, it is possible to scale up the dataset and sample object states at smaller intervals. For instance, instead of having discrete states of a clamp, we can sample from a continuous parameter space evenly while capturing. This would enable the prediction of the continuous parameter such as the angle of a pair of scissors, thus opening up a wider range of applications. Future direction should include how to design mechanisms that will not affect the apperance of the objects during capturing while being able to actuate the objects.

6.4 Leveraging Robots for Large-Scale Data Collection

Robots possess the capability to perform repetitive tasks consistently and efficiently. Researchers in computer vision and HCI have explored various approaches to employing robots for data collection across a diverse range of applications [7, 16, 34]. This has opened up new opportunities for augmenting tasks that necessitate a substantial amount of repetitive work, such as data collection for multiple objects, through the integration of robotic systems. By leveraging robotic systems, researchers can not only streamline the data collection process but also minimize human error and fatigue. This can lead to the acquisition of more accurate and reliable datasets, which are critical for the development and evaluation of advanced algorithms and models.

In addition to automating repetitive tasks, robotic systems can be equipped with various sensors and end effectors to collect multimodal data, such as visual, tactile, and auditory information. This can significantly enrich the datasets and provide researchers with a more comprehensive understanding of the objects and environments being studied. As robotics technology continues to advance, we can expect even more sophisticated and versatile robotic systems to be employed in the data collection process. This will ultimately lead to more robust, accurate, and diverse datasets, which will contribute to the improvement of various computer vision and HCI applications.

7 CONCLUSION

In this paper, we present RoCap, a robotic pipeline for data collection of appearance-changing objects. This system addresses the challenge of pose estimation for objects with deformable properties (*e.g.*, plush toys), viewing-angle dependent properties including transparent materials (*e.g.*, glass flasks) and reflective materials (*e.g.*, pitcher) or objects to be actuated with multiple mechanical states (*e.g.*, clamps or spray bottle heads). By employing a robotic arm to hold these objects and capture image data, which can be utilized by anyone possessing a 6 DoF robotic arm, we can train a simple deep learning model to perform pose estimation on the objects. We conducted both quantitative and qualitative evaluation of our approach comparing to a few-shot learning framework, which was trained on 3D mesh reconstructed from a video. The results demonstrated the feasibility and potential of RoCap.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

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 (CVPR)*. 7822–7831.

[2] Connelly Barnes, David E Jacobs, Jason Sanders, Dan B Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. 2008. Video puppetry: a performative interface for cutout animation. In *ACM SIGGRAPH Asia 2008 papers*. 1–9.

[3] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. 2014. Learning 6d object pose estimation using 3d object coordinates. In *European conference on computer vision*. Springer, 536–551.

[4] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020). https://doi.org/10.3390/info11020125

[5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. 2015. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*. IEEE, 510–517.

[6] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. 2021. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9044–9053.

[7] Toby Chong, I-Chao Shen, Nobuyuki Umetani, and Takeo Igarashi. 2021. Per Garment Capture and Synthesis for Real-time Virtual Try-on. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 457–469.

[8] Anton Konushin Danila Rukhovich, Anna Vorontsova. 2022. ImVoxelNet: Image to Voxels Projection for Monocular and Multi-View General-Purpose 3D Object Detection. (2022), 2397–2406.

[9] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 145–156.

[10] Jacques Denavit and Richard S Hartenberg. 1955. A kinematic notation for lower-pair mechanisms based on matrices. (1955).

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[12] Mustafa Doga Dogan, Ahmad Taka, Michael Lu, Yunyi Zhu, Akshat Kumar, Aakar Gupta, and Stefanie Mueller. 2022. InfraredTags: Embedding Invisible AR Markers and Barcodes Using Low-Cost, Infrared-Based 3D Printing and Imaging Tools. In *CHI Conference on Human Factors in Computing Systems*. 1–12.

[13] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE international conference on computer vision*. 1301–1310.

[14] Heng Fan, Halady Akhilesha Miththanthaya, Siranjiv Ramana Rajan, Xiaoqiong Liu, Zhilin Zou, Yuewei Lin, Haibin Ling, et al. 2021. Transparent object tracking benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10734–10743.

[15] Erika Nathalia Gama Melo, Oscar Fernando Aviles Sanchez, and Darlo Amaya Hurtado. 2014. Anthropomorphic robotic hands: a review. *Ingeniería y desarrollo* 32, 2 (2014), 279–313.

[16] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. 2022. Objectfolder 2.0: A multisensory object dataset for sim2real transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10598–10608.

[17] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 47, 6 (2014), 2280–2292.

[18] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. 2020. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3196–3206.

[19] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D Puppetry: A Kinect-Based Interface for 3D Animation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 423–434. https://doi.org/10.1145/2380116.2380170

[20] Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1957–1967.

[21] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. 2013. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*. Springer, 548–562.

[22] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. 2017. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 880–888.

[23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. 2019. Searching for MobileNetV3. arXiv:1905.02244 [cs.CV]

[24] Michail Kalaitzakis, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. 2021. Fiducial markers for pose estimation. *Journal of Intelligent & Robotic Systems* 101, 4 (2021), 1–26.

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[26] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. 2015. Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In *Proceedings of the IEEE international conference on computer vision*. 954–962.

[27] Jiahao Li, Meilin Cui, Jeeeun Kim, and Xiang'Anthony' Chen. 2020. Romeo: A design tool for embedding transformable parts in 3d models to robotically augment default functionalities. In *Proceedings of the 33rd Annual Acm Symposium on User Interface Software and Technology*. 897–911.

[28] Jiahao Li, Jeeeun Kim, and Xiang'Anthony' Chen. 2019. Robiot: A design tool for actuating everyday objects with automatically generated 3D printable mechanisms. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 673–685.

[29] Jiahao Li, Alexis Samoylov, Jeeeun Kim, and Xiang 'Anthony' Chen. 2022. Roman: Making Everyday Objects Robotically Manipulable with 3D-Printable Add-on Mechanisms. 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 272, 17 pages. https://doi.org/10.1145/3491102.3501818

[30] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. 2020. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3706–3715.

[31] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. 2021. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14687–14697.

[32] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. 2022. Gen6D: Generalizable Model-Free 6-DoF Object Pose Estimation from RGB Images. *arXiv preprint arXiv:2204.10776* (2022).

[33] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. https://doi.org/10.48550/ARXIV.1906.08172

[34] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. 2019. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1048–1055.

[35] Pat Marion, Peter R Florence, Lucas Manuelli, and Russ Tedrake. 2018. Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3235–3242.

[36] Makoto Okabe, Kenshi Takayama, Takashi Ijiri, and Takeo Igarashi. 2007. Light shower: a poor man's light stage built with an off-the-shelf umbrella and projector. In *ACM SIGGRAPH 2007 sketches*. 62–es.

[37] Frank C Park and Bryan J Martin. 1994. Robot sensor calibration: solving AX= XB on the Euclidean group. *IEEE Transactions on Robotics and Automation* 10, 5 (1994), 717–721.

[38] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. 2019. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4561–4570.

[39] 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*. 1–14.

[40] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3234–3243.

[41] Johannes L Schonberger and Jan-Michael Frahm. 2016. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4104–4113.

[42] Chen Song, Jiaru Song, and Qixing Huang. 2020. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 431–440.

[43] Yongzhi Su, Jason Rambach, Alain Pagani, and Didier Stricker. 2021. Synponet—Accurate and fast CNN-based 6DoF object pose estimation using synthetic training. *Sensors* 21, 1 (2021), 300.

[44] Ryo Suzuki, Rubaiat Habib Kazi, Li-yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 166–181. https://doi.org/10.1145/3379337.3415892

[45] Stefan Thalhammer, Dominik Bauer, Peter Hönig, Jean-Baptiste Weibel, José García-Rodríguez, and Markus Vincze. 2023. Challenges for Monocular 6D Object Pose Estimation in Robotics. *arXiv preprint arXiv:2307.12172* (2023).

[46] Jiří Ulrich, Ahmad Alsayed, Farshad Arvin, and Tomáš Krajník. 2022. Towards Fast Fiducial Marker with Full 6 DOF Pose Estimation. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (Virtual Event) *(SAC '22)*. Association for Computing Machinery, New York, NY, USA, 723–730. https://doi.org/10.1145/3477314.3507043

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

[48] Hao Yang, Chen Shi, Yihong Chen, and Liwei Wang. 2022. Boosting 3D Object Detection via Object-Focused Image Fusion. *arXiv preprint arXiv:2207.10589* (2022).

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

[50] Jiawen Zhu, Zhenyu Chen, Zeqi Hao, Shijie Chang, Lu Zhang, Dong Wang, Huchuan Lu, Bin Luo, Jun-Yan He, Jin-Peng Lan, et al. 2023. Tracking Anything in High Quality. *arXiv preprint arXiv:2307.13974* (2023).

[51] Douglas E Zongker, Dawn M Werner, Brian Curless, and David H Salesin. 1999. Environment matting and compositing. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 205–214.