KPA-Tracker: Towards Robust and Real-Time Category-Level Articulated Object 6D Pose Tracking

Liu Liu[1]*, Anran Huang[1]*, Qi Wu[2], Dan Guo[1]†, Xun Yang[3], Meng Wang[1]

[1]Hefei University of Technology
[2]Shanghai Jiao Tong University
[3]University of Science and Technology of China
{liuliu, guodan, wangmeng}@hfut.edu.cn, 2459259637@qq.com, robotics_qi@sjtu.edu.cn, xyang21@ustc.edu.cn

Abstract

Our life is populated with articulated objects. Current category-level articulation estimation works largely focus on predicting part-level 6D poses on static point cloud observations. In this paper, we tackle the problem of category-level online robust and real-time 6D pose tracking of articulated objects, where we propose **KPA-Tracker**, a novel 3D **K**ey**P**oint based **A**rticulated object pose **Tracker**. Given an RGB-D image or a partial point cloud at the current frame as well as the estimated per-part 6D poses from the last frame, our KPA-Tracker can effectively update the poses with learned 3D keypoints between the adjacent frames. Specifically, we first canonicalize the input point cloud and formulate the pose tracking as an inter-frame pose increment estimation task. To learn consistent and separate 3D keypoints for every rigid part, we build **KPA-Gen** that outputs the high-quality ordered 3D keypoints in an unsupervised manner. During pose tracking on the whole video, we further propose a keypoint-based articulation tracking algorithm that mines keyframes as reference for accurate pose updating. We provide extensive experiments on validating our KPA-Tracker on various datasets ranging from synthetic point cloud observation to real-world scenarios, which demonstrates the superior performance and robustness of the KPA-Tracker. We believe that our work has the potential to be applied in many fields including robotics, embodied intelligence and augmented reality. All the datasets and codes are available at https://github.com/hhhhhar/KPA-Tracker.

Introduction

Articulated objects are very common in daily life. Accurately estimating and tracking 6D pose is crucial for a variety of computer vision and robotics applications, such as robot manipulation (Xiong et al. 2023; Geng et al. 2023), visual understanding (Li, Guo, and Wang 2021; Guo, Wang, and Wang 2021), human object interaction (Yang et al. 2022b,a; Li et al. 2023), embodied intelligence (Romero, Tzionas, and Black 2017; Fu et al. 2022) and VR/AR applications (Clark, Newman, and Dutta 2022). Unlike category-level articulated object pose estimation that predicts 6D poses from static point cloud or RGB-D image observations (Liu et al. 2022b,a, 2023), articulation pose tracking begins to attract attention from computer vision researchers in recent years. Given a sequence of point clouds for articulated object motion as well as the initialized per-part poses from the first frame, the pose tracker aims to update those for the rest frames (Weng et al. 2021).

Under this problem setting, some works attempt to transfer the static articulation pose estimation methods such as NOCS (Wang et al. 2019a) and A-NCSH (Li et al. 2020) into pose tracking task (Weng et al. 2021) but suffer from the following issues: (1) they rely on learning the object features from the visible points, which results in incomplete shape and kinematics modeling under the camera views with self-occlusion. (2) they require per-pixel representation learning that hinders the performance of tracking speed on the video. Thus, these limitations prevent the category-level articulated object trackers from achieving robust and real-time tracking performance.

In this paper, targeting at building a robust and real-time articulated object tracking approach, we propose **KPA-Tracker**, a novel 3D **K**ey**P**oint based **A**rticulated object pose **Track**ing framework. Given an initial articulated object pose at the first frame, our KPA-Tracker is to continuously track the 6D pose for each individual rigid part of an articulated object with learned 3D keypoints from the adjacent frames. The main motivation of our KPA-tacker is to track the per-part poses by registering a list of ordered 3D keypoints, which are learned to model the amodal shape explicitly from the point cloud observation. Exploiting these ordered 3D keypoints, KPA-Tracker might alleviate the effect of invisible parts and achieve more robust tracking performance. In addition, the pose tracker can more effectively compute the pose increment between two frames due to the sparsity of the keypoints.

In our KPA-Tracker, to relieve the difficulty of predicting the pose for the observed point cloud in camera space, we first canonicalize the input space at the current frame by transforming the point cloud using the inverse poses from the previous frame. This strategy formulates the articulated object tracking as an inter-frame pose increment estimation task. Next, we propose **KPA-Gen**, an unsupervised manner to automatically generate and train a sequence of ordered sparse 3D keypoints as articulation modeling for per-part representation, which can be used as supervision for KPA-

*These authors contributed equally.

†Corresponding author

Tracker learning without any human keypoint annotations. The learned 3D keypoints can be applied to jointly model the geometric shape and part motion, where the former ensures the generalization of unseen articulated objects and the latter contributes to accurate and fast per-part pose tracking. Finally, to achieve robust tracking performance in the video and avoid cumulative error, we also propose a keypoint-based articulation tracking algorithm that mines key frame as reference for tracking the whole video.

We evaluate our KPA-Tracker on both point clouds and RGB-D images, where the objects range from the synthetic dataset PartNet-Mobility (Xiang et al. 2020) to the semi-synthetic dataset ReArt-48 (Liu et al. 2022b). To further evaluate the generalization ability of our method to real-world scenarios, we test KPA-Tracker on a RobotArm dataset that contains much more diverse and complex scenes. We believe that the extensive experiments show the superior performance of the KPA-Tracker compared with state-of-the-arts on the category-level articulated object 6D pose tracking task.

Our contributions can be summarized as follows:

- KPA-Tracker is a novel framework proposed to solve the problem of category-level articulated object 6D pose tracking, where we introduce a list of 3D keypoints as articulation representation for per-part pose tracking.
- We propose an unsupervised learning method namely KPA-Gen to automatically generate the high-quality 3D keypoints on the complete point cloud, which can be used as supervision information for KPA-Tracker learning.
- The efficiency and robustness of the KPA-Tracker are demonstrated through the evaluation of the videos with either point clouds or RGB-D images for the articulated object pose tracking task, using various datasets ranging from synthetic to real-world scenarios.

Related Work

Category-level Articulation Pose Estimation

Category-level object pose estimation aims to aim at predicting the pose of previously unseen objects (Wang et al. 2019a; Manhardt et al. 2020; Di et al. 2022; Wang et al. 2019b; Liu et al. 2020). Beyond the definition of rigid object pose estimation, articulated objects hold a limited number of rigid parts that are connected by different types of joints. Thus, category-level articulation pose estimation requires per-part 6D pose as predicted results. A-NCSH extends the notation of normalized coordinates into articulation to estimate part-level poses (Li et al. 2020). Liu et al. further update the setting into the real-world articulated object analysis and propose part pair for investigating unseen instances (Liu et al. 2022b), as well as an integral pipeline to leverage articulation pose for robot manipulation (Liu et al. 2022a). Additionally, Xue et al. (Xue et al. 2021) propose using key-points as an articulation modeling to speed up the inference time for accurate pose estimation. Although the above works solve category-level articulation pose estimation well with satisfied performance, they are hard to apply as ready-to-use recipes into the pose tracking task since the dense prediction paradigm limits the robustness and inference speed.

Category-level Articulation Pose Tracking

To handle the problem of category-level online pose tracking of articulated objects, Weng et al. propose an end-to-end pipeline that learns to update the pose compared with those in the previous frames (Weng et al. 2021; Liu et al. 2022d). This solution takes point clouds as input and estimates pixel-level voting vectors for inter-frame pose change prediction. Otherwise, many researchers focus on keypoint-based object representation and modeling. Lin et al. track the rigid objects with predicted 2D keypoints in a RGB sequence (Lin et al. 2022). Considering depth information input, Heppert et al. introduce factor graphs into category-independent object pose tracking (Heppert et al. 2022) while Wen et al. exploit tracked articulated object pose for 3D reconstruction (Wen et al. 2023). Despite these achievements for pose tracking (Jain et al. 2021; Liu et al. 2022c), the keypoints extracted by these methods can only model the geometric information of the input object but ignore the kinematic motion when the part is rotating or translating along the corresponding joint. In this work, we introduce an unsupervised keypoint generation strategy (referred to 6-PACK (Wang et al. 2020) and Fernadez's work (Fernandez-Labrador et al. 2020)) into articulated objects, and model both geometry and part motion with the learned ordered 3D keypoints for per-part pose tracking.

Problem Statement

In this paper, we target at the problem of tracking the per-part 6D poses of articulated objects from known categories. We follow the category-level articulated object and part definition in A-NCSH (Li et al. 2020) and CAPTRA (Weng et al. 2021), and adopt the assumption that the number of rigid parts and kinematic structures is constant for all the objects in the same category. In this paper, the problem and notations are defined as follows: Given a live stream of point clouds $\{X_t\}_{t\geq 0}$ where in each t frame the point cloud contains K rigid parts and $S^{(k)}$ represents the points of k-th part, within the per-part pose $T_0^{(k)} = \{R_0^{(k)}, \mathbf{t}_0^{(k)}\}_{k=1}^K$ at 0-th frame, the outputs of the articulation tracker consist of category-level per-part 6D poses $T_t^{(k)} = \{R_t^{(k)}, \mathbf{t}_t^{(k)}\}_{k=1}^K$ at all the frames $t > 0$.

In our proposed paradigm of 3D keypoint-based articulation modeling for category-level articulated object pose tracking task, the tracking model aims to learn an ordered list of 3D keypoints $P = \{p_j \in \mathbb{R}^3\}_{j=1}^M$ with M keypoints from the point cloud $X^{(k)} = X \cdot S^{(k)}$ for k-th rigid part. In this way, we can transfer the per-part pose tracking task as a per-part 3D keypoints registration task. In other words, given the estimated articulated object pose $T_{t-1}^{(k)} = \{R_{t-1}^{(k)}, \mathbf{t}_{t-1}^{(k)}\}_{k=1}^K$ at $t-1$ frame, the tracking model would estimate the pose increment $\Delta T_t^{(k)} = (T_{t-1}^{(k)})^{-1} T_t^{(k)}$ by registering the ordered 3D keypoints of P_t and P_{t-1}.

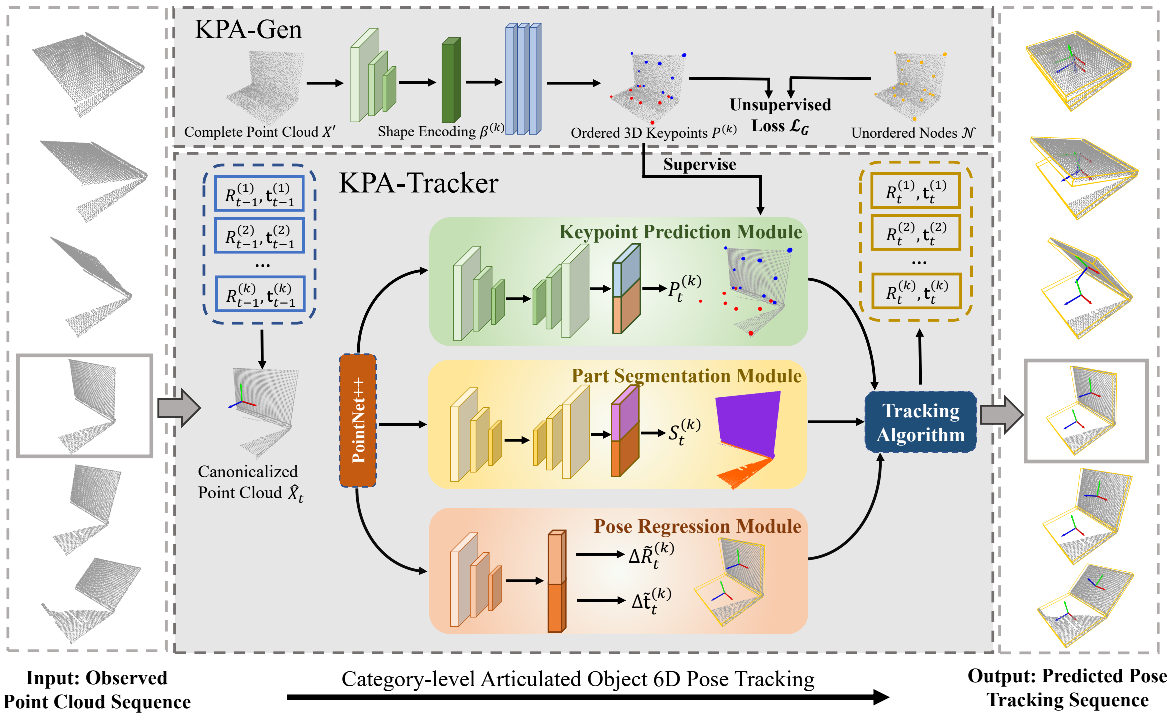
KPA-Gen
Complete Point Cloud X'
Shape Encoding $\beta^{(k)}$
Ordered 3D Keypoints $P^{(k)}$
Unsupervised Loss $\mathcal{L}_G$
Unordered Nodes $\mathcal{N}$
Supervise
KPA-Tracker
$R_{t-1}^{(1)}, \mathbf{t}_{t-1}^{(1)}$
$R_{t-1}^{(2)}, \mathbf{t}_{t-1}^{(2)}$
...
$R_{t-1}^{(k)}, \mathbf{t}_{t-1}^{(k)}$
Canonicalized Point Cloud $\hat{X}_t$
PointNet++
Keypoint Prediction Module
$P_t^{(k)}$
Part Segmentation Module
$S_t^{(k)}$
Pose Regression Module
$\Delta\tilde{R}_t^{(k)}$
$\Delta\tilde{\mathbf{t}}_t^{(k)}$
Tracking Algorithm
$R_t^{(1)}, \mathbf{t}_t^{(1)}$
$R_t^{(2)}, \mathbf{t}_t^{(2)}$
...
$R_t^{(k)}, \mathbf{t}_t^{(k)}$
Input: Observed Point Cloud Sequence
Category-level Articulated Object 6D Pose Tracking
Output: Predicted Pose Tracking Sequence

Figure 1: The overall pipeline of KPA-Tracker. Taking observed partial point cloud t frame as input, our KPA-Tracker first canonicalizes the point cloud with the estimated pose from the $t-1$ frame. Then it is processed by a PointNet++ architecture and predicts per-part 3D ordered keypoints, part segmentation and pose regression, in which the keypoints are supervised by those generated from KPA-Gen. The tracking algorithm refines and obtains the final poses by tracking the per-part keypoints.

KPA-Tracker Architecture

In this section, we introduce the KPA-Tracker in detail. The overall pipeline is illustrated in Fig. 1. Firstly, the input cloud point X_t at t frame is canonicalized by the estimated 6D pose $T_{t-1}^{(k)}$ of $t-1$ frame (Sec.). Next, we propose an unsupervised method that generates the ordered 3D keypoints from the complete point cloud (Sec.). Finally, we describe the learning pipeline of KPA-Tracker (Sec.) and the tracking algorithm (Sec. 13).

Articulation Pose Canonicalization

Inspired by (Weng et al. 2021), we canonicalize the per-part 6D pose $T_t^{(k)} = \{R_0^{(k)}, \mathbf{t}_t^{(k)}\}_{k=1}^{K}$ at t frame with the pose $T_{t-1}^{(k)} = \{R_{t-1}^{(k)}, \mathbf{t}_{t-1}^{(k)}\}_{k=1}^{K}$ at the previous frame. This operation has the following advantages: (1) the 6D pose estimation task in camera space can be transferred into an interframe delta pose estimation task in a "pseudo-canonical" space, which is friendly for neural network learning. (2) the canonicalized point cloud can effectively eliminate the effect of diverse joint states of each movable rigid part, and provide a shape and kinematic prior that largely contributes to 3D keypoints learning.

In articulation pose canonicalization, given a point cloud X_t, the canonicalized point cloud $\hat{X}_t$ can be computed as the product of the inverse transformation of $T_{t-1}^{(k)} = \{R_{t-1}^{(k)}, \mathbf{t}_{t-1}^{(k)}\}_{k=1}^{K}$ and X_t:

$$\hat{X}_t^{(k)} = (T_{t-1}^{(k)})^{-1} X_t^{(k)} = (R_{t-1}^{(k)})^{-1} (X_t^{(k)} - \mathbf{t}_{t-1}^{(k)}) \quad (1)$$

where $X_t^{(k)}$ is the point cloud that belongs to the k-th rigid part and can be computed by multiplying the estimated part mask $S_t^{(k)}$:

$$X_t^{(k)} = X_t \cdot \mathbb{1}(S_t = k) \quad (2)$$

By canonicalizing the input point cloud, the tracking model will only predict the per-part pose increment $\Delta T_t^{(k)} = \{\Delta R_t^{(k)}, \Delta \mathbf{t}_t^{(k)}\}$ between t and $t-1$ frame. Since $\Delta R_t^{(k)}$ is approximately a 3×3 identity matrix that $\Delta R_t^{(k)} \approx I$ and $\Delta \mathbf{t}_t^{(k)} \approx 0$ on the adjacent frames, the neural network would be more sensitive to the slight pose changes, and largely improve the tracking performance.

Articulation Modeling with 3D Keypoint

Taking the canonicalized point cloud $\hat{X}_t$ at t frame as input, a simple way to track the per-part 6D pose is to register the interframe point cloud like ICP (Zhou, Park, and Koltun 2018) or learn point correspondences for optimization like BundleTrack (Wen and Bekris 2021). In this paper, to boost the tracking speed, we propose to register the per-part pose increment $\Delta T_t^{(k)}$ by learning a list of ordered 3D keypoints for every rigid part. Ideally, given the learned ordered 3D keypoints $P_t^{(k)}$ and $P_{t-1}^{(k)}$ at t and $t-1$ frame, the pose increment $\Delta T_t^{(k)}$ can satisfy:

$$P_t^{(k)} = \Delta T_t^{(k)} P_{t-1}^{(k)} \tag{3}$$

In order to learn the 3D keypoints with high quality, the 3D keypoints need to meet the following requirements: (1) **Separateness.** All the keypoints need to be separately distributed on the surface of the rigid part, which demonstrates the ability to learn the geometric shape of articulated objects. (2) **Consistency.** The learned 3D keypoints are expected to be consistent among all the instances from the same category. In other words, the order of the keypoints remains the same on different objects. (3) **Symmetry.** In practice, we find many categories of articulated objects are symmetric, so the learned keypoints need to be distributed uniformly and symmetrically in these instances. Unfortunately, there are no keypoint annotations for articulated object datasets at the current stage that meet these requirements and are used for articulation pose tracking. Thus, we propose an automatic articulation keypoints generation method, namely **KPA-Gen**, to unsupervisely generate per-part 3D keypoints $P^{(k)}$ as annotations for KPA-Tracker training.

In KPA-Gen, the input is complete point cloud X' of the articulated object in the "pseudo-canonical" space where the X' is canonicalized by the pose $T_{t-1}^{(k)} = \{R_{t-1}^{(k)}, \mathbf{t}_{t-1}^{(k)}\}$ at $t-1$ frame. We apply PointNet++ (Qi et al. 2017) encoder-decoder architecture to obtain the pixel-level features and design two branches for node and keypoint learning respectively. The node branch predicts a sparse tuple of unordered nodes $\mathcal{N} = \{n_j \in \mathbb{R}^3\}_{j=1}^{M}$, which can be regarded as potential 3D keypoints but not consistent. These unordered nodes are initialized by Farthest Point Sampling (FPS) on the input X' and represented by grouped clusters where each point in the cluster is corresponding to an offset to the target node. Since these nodes are separated and floating on the surface, they can indicate the shape geometry and be adopted as keypoint constraints for training.

Based on these unordered nodes, we can design the keypoint branch that trains the part-level ordered 3D keypoints explicitly. These 3D keypoints are learned by the learnable shape parameters $\mathcal{B}^{(k)}$ that describe the category-specific shape information and encode the keypoint into the output of keypoint branch $\beta^{(k)}$. In this way, the 3D keypoints can be generated by:

$$P^{(k)} = F(\beta^{(k)}; \mathcal{B}^{(k)}) \tag{4}$$

where the shape parameters $\mathcal{B}^{(k)}$ are shared within the category so they ensure the category-level generalization. To train the KPA-Gen model, the loss functions are designed to guarantee separateness, consistency and symmetry for 3D keypoint learning. Firstly, the per-part chamfer loss $\mathcal{L}_{chf}^{(k)}$ is employed to make the distance between unordered nodes and order keypoints as small as possible, and defined as:

$$\mathcal{L}_{chf}^{(k)} = \sum_{n_i}^{\mathcal{N}^{(k)}} \min_{p_j \in P^{(k)}} \|n_i - p_j\|_2 + \sum_{p_j}^{P^{(k)}} \min_{n_i \in \mathcal{N}^{(k)}} \|n_i - p_j\|_2 \tag{5}$$

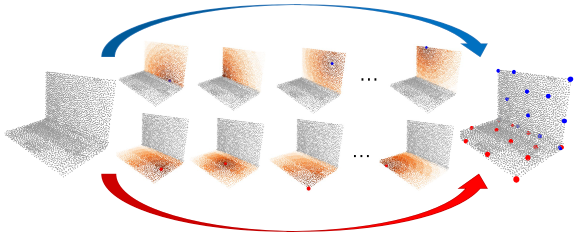

Figure 2: Heatmap and voting scheme for keypoint learning

Secondly, to avoid collapsing of the keypoints, the per-part separation loss $\mathcal{L}_{sep}$ is designed to prevent multiple nodes collaps to the same location but not keep these nodes too far away from each other:

$$\mathcal{L}_{sep}^{(k)} = \sum_{n_i}^{\mathcal{N}^{(k)}} \sum_{n_j}^{\mathcal{N}^{(k)}} \max(0, \delta^2 - \|n_i - n_j\|_2) \tag{6}$$

Finally, to ensure the learned keypoints can cover the whole shape geometry of the rigid part, the per-part coverage loss $\mathcal{L}_{cov}$ is designed by the difference between the volume of keypoints $P^{(k)}$ and that of the input point cloud $X'^{(k)}$, and also penalize those keypoints that are far away from $X'^{(k)}$:

$$\mathcal{L}_{cov}^{(k)} = \|\mathrm{vol}(P^{(k)}) - \mathrm{vol}(X'^{(k)})\|_2 + \sum_{n_i}^{\mathcal{N}^{(k)}} \|n_i - X'^{(k)}\|_2 \tag{7}$$

The total loss $\mathcal{L}_G$ for KPA-Gen that generate unsupervised per-part 3D ordered keypoints is the weighted sum of losses from chamfer loss $\mathcal{L}_{chf}$, separation loss $\mathcal{L}_{sep}$ and coverage loss $\mathcal{L}_{cov}$ with λ_{chf}, λ_{sep} and λ_{cov} where:

$$\mathcal{L}_G = \lambda_{chf}\mathcal{L}_{chf} + \lambda_{sep}\mathcal{L}_{sep} + \lambda_{cov}\mathcal{L}_{cov} \tag{8}$$

Network Modules

Given the generated 3D keypoints $P_t^{(k)}$ from KPA-Gen as supervision for every canonicalized point cloud X_t, we train the KPA-Tracker for category-level articulated object pose tracking task. The input of KPA-Tracker is canonicalized partial point cloud $\hat{X}_t$ at t frame, which is processed by a PointNet++ architecture (Qi et al. 2017) as feature extractor to obtain per-pixel feature vectors $f_i \in \mathbb{R}^{128}$. Then we build three parallel modules at the end of the feature extractor: part segmentation module, keypoint prediction module and pose regression module.

Part Segmentation Module. For each canonicalized point $\hat{x}_i$, we build three multi-layer perceptions (MLPs) with ReLU activation function that outputs K channels for part segmentation $s_i^{(k)}$. We use cross-entropy loss to train the part segmentation module.

Keypoint Prediction Module. We use an offset-voting mechanism with heatmap of X_t' to predict 3D keypoints $P^{(k)} = \{p_j^{(k)}\}_{j=1}^{M}$ for each part as shown in Fig. 2. Specifically, we also build three MLPs and output $4KM$ channels, where KM channels indicate the heatmap of the i-th

Algorithm 1: Tracking algorithm on the whole video with the learned 3D keypoints

```
input  : Observed point cloud sequence $\{X_t\}_{t\geq 0}$
         Per-part 3D keypoints $\{P_t^{(k)}\}_{t\geq 0}$
         Per-part 6D pose $T_0^{(k)}$ at the first frame.
output: Per-part 6D pose $T_t^{(k)}$ at all the $t > 0$ frames
Initialize keyframe pool $B$
Add frame $t = 0$ into keyframe pool $B$
for $t > 0$ frames do
    if $t\%N == 0$ then
        Add $t$-th frame into key frame pool $B$
    end
    Obtain the nearest keyframe $t'$ for $t$-th frame
      from keyframe pool $B$
    for $K$ parts do
        Align $P_t^{(k)}$ to $P_{t'}^{(k)}$ for $k$-th part
        Compute the delta pose $\Delta T_t^k$ for $k$-th part
        Compute the pose $T_t^{(k)}$ at current $t$-th frame
    end
end
```

point voting for which keypoint and $3KM$ channels indicate the offset between this point and the target keypoint. The heatmap for j-th keypoint H_i^j is defined as:

$$H_i^j = 1 - \|x_i - p_j^*\|_2/\sigma \tag{9}$$

where σ is the distance threshold and we only consider the points for voting whose distance to the keypoint p_j is smaller than σ. * indicates the ground truth. The offset V_i^j is defined as:

$$V_i^j = (x_i - p_j^*)/H_i^j \tag{10}$$

Therefore, within the predicted heatmap H_i^j, offset V_i^j as well as the segmentation $s_i^{(k)}$, the predicted j-th keypoint for k-th part can be obtained by:

$$p_j^{(k)} = \frac{1}{N}\sum_{i=1}^{N} s_i^{(k)} H_i^j (x_i + V_i^j) \tag{11}$$

Pose Regression Module. We build a pose regression module to provide an initial per-part delta pose $\Delta\widetilde{T}_t^{(k)} = \{\Delta\widetilde{R}_t^{(k)}, \Delta\widetilde{\mathbf{t}}_t^{(k)}\}$ for t frame by direct regression scheme. To be specific, for rotation $\Delta\widetilde{R}_t^{(k)}$, we regress the quaternion vector as pose replacement. For translation $\Delta\widetilde{\mathbf{t}}_t^{(k)}$, we directly regress it with L2 loss function. We use a joint-centric articulation pose modeling strategy that uses joint state θ as pose representation. Please refer to supplementary materials for more details.

Articulation Pose Tracking Algorithm

Taking the per-part keypoints $P_{t-1}^{(k)}$ and pose $T_{t-1}^{(k)}$ at $t-1$ frame, and keypoints $P_t^{(k)}$ and initial delta pose $\Delta\widetilde{T}_t^{(k)}$ at t frame as input, KPA-Tracker can output the per-part pose $T_t^{(k)}$. To be specific, we build a tracking energy function E to calculate $T_t^{(k)}$ with optimization scheme:

$$E = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Var}(\|\Delta\widetilde{T}_t^{(k)} \cdot p_{j,t-1}^{(k)} - p_{j,t}^{(k)}\|_2) \tag{12}$$

Thus, we can minimize the energy function E to refine the $\Delta\widetilde{T}_t^{(k)}$.

$$\Delta T_t^{(k)} = \underset{\Delta\widetilde{T}_t^{(k)}}{\arg\min}\, E \tag{13}$$

The pose $T_t^{(k)}$ at t frame can be recovered by:

$$T_t^{(k)} = \Delta T_t^{(k)} T_{t-1}^{(k)} \tag{14}$$

In this way, we can track the articulated object pose $T_t^{(k)}$ from $T_{t-1}^{(k)}$. In addition, to achieve robust pose tracking in the whole video, we propose a simple but effective tracking algorithm that mines the keyframes at N intervals and tracks each frame into the nearest keyframe. The overall articulation tracking procedure with the learned 3D keypoints for the video is summarized in Algorithm 1.

Experiments

Experimental Settings

Datasets. To train the KPA-Tracker as well as the KPA-Gen network, we generate the corresponding datasets for training and validation. Firstly, we build a synthetic articulated object tracking dataset with the objects from PartNet-Mobility (Xiang et al. 2020). We select five categories of laptop, dishwasher, eyeglasses, scissors and drawer referred by ArtImage (Xue et al. 2021), where there are 30K frames and 300 videos for each category. Next, we generate a semi-synthetic dataset for articulated object tracking task from ReArt-48 repository using SAMERT technique (Liu et al. 2022b), which generates more than 50K frames for each category. To validate the performance of the KPA-Tracker, we adopt degree error for 3D rotation, distance error for 3D translation and tracking speed for real-time analysis. We also test the cumulative tracking error on the whole video.

Implementation Details. During the data pre-processing, input point clouds are sampled into 2,048 points and the objects in RGB-D images are also cropped and projected into the point cloud as the network inputs. The initial learning rate is 0.001 and we adopt cosine learning rate decay during training. The total training epoch is 100. The hyperparameters are: $\lambda_{chf} = 1.0$, $\lambda_{sep} = 2.0$, $\lambda_{cov} = 1.0$, $\sigma = 0.1$. All the experiments are implemented on four NVIDIA GeForce RTX 4090 GPUs with 24GB memory.

Articulated Object Pose Tracking

We report the results of KPA-Tracker evaluated on the synthetic dataset containing the articulated objects from PartNet-Mobility (Xiang et al. 2020) in Table 1. As it can be

Category	Method	Per-part 6D Pose		Inference Time (s)
		Rotation Error (°)	Translation Error (m)	
Laptop	A-NCSH (Li et al. 2020)	8.5, 9.2	0.084, 0.103	1.67
	OMAD (Xue et al. 2021)	8.7, 8.9	0.092, 0.096	0.34
	Oracle ICP (Zhou, Park, and Koltun 2018)	10.8, 16.2	0.131, 0.174	0.72
	CAPTRA* (Weng et al. 2021)	5.9, **5.3**	0.080, **0.063**	0.10
	KPA-Tracker (Ours)	**5.0**, 7.8	**0.076**, 0.084	**0.05**
Eyeglasses	A-NCSH (Li et al. 2020)	7.6, 24.8, 26.6	0.079, 0.324, 0.319	2.59
	OMAD (Xue et al. 2021)	8.5, 9.3, 9.6	0.105, 0.123, 0.118	0.84
	Oracle ICP (Zhou, Park, and Koltun 2018)	14.3, 26.5, 29.6	0.154, 0.198, 0.196	0.96
	CAPTRA* (Weng et al. 2021)	4.5, 12.6, 13.1	0.054, 0.097, 0.084	0.14
	KPA-Tracker (Ours)	**2.7**, **4.3**, **4.4**	**0.031**, **0.050**, **0.053**	**0.08**
Dishwasher	A-NCSH (Li et al. 2020)	5.0, 5.7	0.074, 0.119	1.70
	OMAD (Xue et al. 2021)	6.2, 7.0	0.126, 0.207	0.36
	Oracle ICP (Zhou, Park, and Koltun 2018)	7.8, 12.4	0.196, 0.234	0.67
	CAPTRA* (Weng et al. 2021)	4.6, 5.4	**0.055**, 0.089	0.11
	KPA-Tracker (Ours)	**3.7**, **4.9**	0.061, **0.087**	**0.06**
Scissors	A-NCSH (Li et al. 2020)	5.0, 5.7	0.041, 0.057	1.21
	OMAD (Xue et al. 2021)	6.1, 6.6	0.055, 0.069	0.29
	Oracle ICP (Zhou, Park, and Koltun 2018)	16.8, 14.5	0.185, 0.167	0.49
	CAPTRA* (Weng et al. 2021)	4.1, **4.7**	0.032, 0.039	0.12
	KPA-Tracker (Ours)	**3.9**, 5.2	**0.028**, **0.035**	**0.06**
Drawer	A-NCSH (Li et al. 2020)	8.6, 9.8, 11.5, 8.5	0.088, 0.255, 0.257, 0.175	3.64
	OMAD (Xue et al. 2021)	6.5, 6.5, 6.5, 6.5	0.168, 0.242, 0.243, 0.239	0.62
	Oracle ICP (Zhou, Park, and Koltun 2018)	12.5, 19.8, 19.6, 20.1	0.234, 0.342, 0.338, 0.337	1.03
	CAPTRA* (Weng et al. 2021)	**4.8**, 6.5, 6.3, **6.0**	**0.112**, 0.185, 0.177, **0.156**	0.25
	KPA-Tracker (Ours)	6.1, **6.1**, **6.1**, 6.1	0.145, **0.178**, **0.150**, 0.167	**0.12**

Table 1: Comparison with state-of-the-art on the synthetic dataset with the articulated objects from PartNet-Mobility. The training and validation data are generated by the technique from ArtImage. * indicates the re-implementation on our datasets.

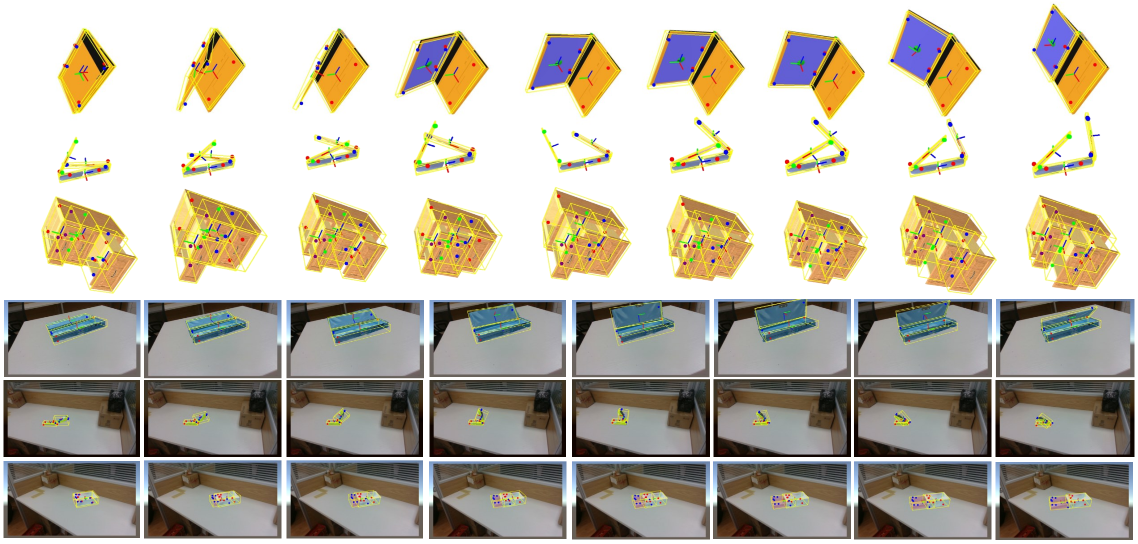

Figure 3: Qualitative results on synthetic dataset of articulated objects from PartNet-Mobility (top) and semi-synthetic dataset of the objects from ReArt-48 (bottom).

seen, compared with the static articulation pose estimation methods such as A-NCSH and OMAD, KPA-Tracker shows a big margin in per-part 6D pose tracking performance with only **3.9**° and **5.2**° on rotation error of category Scissors. For translation errors, our KPA-Tracker can achieve state-of-the-art performance on the five categories with average **6** centimeters. This can be explained by that the learned 3D keypoints can well model the geometry of rigid parts and contributes to better registration. In terms of inference time, KPA-Tracker also obtains the fastest speed compared to A-NCSH and OMAD with only an average **0.07**s per frame. Although our method shows a slight improvement in pose tracking compared to CAPTRA, it holds a better real-time performance. Therefore, we can conclude that KPA-Tracker can well learn 3D keypoints and fully utilize them in pose tracking task. Qualitative results are shown in Fig. 3.

#KP per-part	Rotation Error (°)	Translation Error (m)
4	5.0, 7.8	0.076, 0.084
8	5.8, 8.8	0.084, 0.082
12	6.2, 9.8	0.092, 0.109
16	5.2, 11.3	0.096, 0.115

Table 2: The effect of keypoint numbers on pose tracking

Method	A-NCSH	OMAD	CAPTRA	Ours
FPS	0.8	3.3	10.7	**14.2**

Table 3: Comparison of tracking FPS

Ablation Study

Number of 3D Keypoints. We investigate the effect of the keypoint number, where we experiment with PartNet-Mobility objects from the category Laptop and Table 2 illustrates the results. As it can be seen, We can see that the performance becomes worser when the keypoint number is very large. Intuitively, we believe that too many keypoints make the target locations predicted by keypoint branch less accurate. Thus, it is crucial for the neural network to distinguish between different keypoints when the keypoints are too dense, which determines the upper bound on the performance of the KPA-Tracer.

Real-time Analysis. To further discuss the real-time analysis of KPA-Tracker, we report the FPS performance compared with A-NCSH, OMAD and CAPTRA. In Table 3. we can see that our method achieves the state-of-the-art tracking speed with **14.2** FPS performance, which is dramatically better than A-NCSH and OMAD. Besides, due to keypoint-based pose tracking paradigm adopted in KPA-Tracker, we also obtain a faster tracking performance than CAPTRA, which relies on per-pixel NOCS coordinates prediction.

Generalization Capacity

Experiments on Semi-Synthetic Scenarios. We evaluate the articulated object pose tracking on the dataset generated with ReArt-48 (Liu et al. 2022b) with semi-synthetic scenarios. The tracking results are shown in Table. 4. We can see that the accurate and robust tracking performance with only **5.6**°, **5.9**° on rotation error and **0.010m**, **0.009m** on translation error for category Box. Qualitative results in Fig. 3 also show the high-quality 3D keypoints learning.

Category	Per-part 6D Pose	
	Rotation Error (°)	Translation Error (m)
Box	5.6, 5.9	0.010, 0.009
Stapler	7.6, 8.1	0.010, 0.008
Cutter	3.6, 3.6	0.010, 0.010
Scissors	11.3, 8.5	0.006, 0.006
Drawer	8.4, 8.4	0.027, 0.019

Table 4: Pose tracking results on articulated objects from ReArt-48 dataset.

Per-part Rotation Error (°)						
0.1	0.8	2.5	9.8	13.8	18.6	20.0
Per-part Translation Error (m)						
0.002	0.017	0.025	0.071	0.076	0.139	0.161

Table 5: Pose tracking results on RobotArm dataset

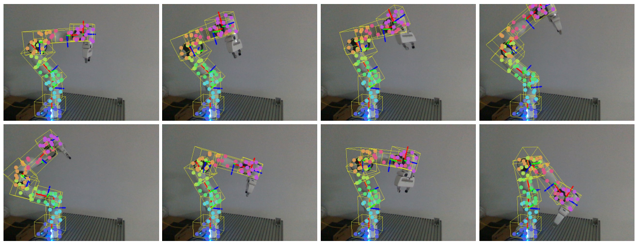

Figure 4: Qualitative results on RobotArm dataset

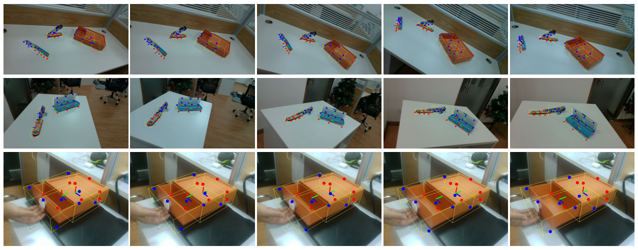

Figure 5: Demonstrations on real-world articulated objects

Experiments on Real-world Scenarios. To investigate the tracking performance in real-world scenarios, we train and evaluate KPA-Tracker on the 7-part RobotArm dataset (Liu et al. 2022b). Table 5 shows the tracking errors and we can observe the passable 6D pose tracking performance on rotation and translation. It is undeniable to suffer from the effect of the multi-depth structure of the robot arm instance. Qualitative results are shown in Fig. 4. Furthermore, we also test the KPA-Tracker in real-world videos and we illustrate the demonstrations in Fig. 5.

Conclusion

In this work, we formulate the category-level articulated object 6D pose tracking as a 3D keypoint registration problem and introduce KPA-Tracker to tackle this issue. Our method designs KPA-Gen to automatically generate 3D ordered keypoints by in unsupervised manner for training KPA-Tracker. During inference, we propose a keyframe-based tracking algorithm that boosts the robustness and real-time performance of the whole video. Experiments demonstrate that KPA-Tracker is able to obtain state-of-the-art tracking performance on various datasets and scenarios.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant 62302143 62272144 and U20A20183, Anhui Provincial Natural Science Foundation under Grant 2308085QF207, and National Key R&D Program of China under Grant 2022ZD0118201.

References

Clark, M.; Newman, M. W.; and Dutta, P. 2022. ARticulate: One-Shot Interactions with Intelligent Assistants in Unfamiliar Smart Spaces Using Augmented Reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1): 1–24.

Di, Y.; Zhang, R.; Lou, Z.; Manhardt, F.; Ji, X.; Navab, N.; and Tombari, F. 2022. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6781–6791.

Fernandez-Labrador, C.; Chhatkuli, A.; Paudel, D. P.; Guerrero, J. J.; Demonceaux, C.; and Gool, L. V. 2020. Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 546–563. Springer.

Fu, H.; Xu, W.; Xue, H.; Yang, H.; Ye, R.; Huang, Y.; Xue, Z.; Wang, Y.; and Lu, C. 2022. Rfuniverse: A physics-based action-centric interactive environment for everyday household tasks. *arXiv preprint arXiv:2202.00199*.

Geng, H.; Xu, H.; Zhao, C.; Xu, C.; Yi, L.; Huang, S.; and Wang, H. 2023. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7081–7091.

Guo, D.; Wang, H.; and Wang, M. 2021. Context-aware graph inference with knowledge distillation for visual dialog. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6056–6073.

Heppert, N.; Migimatsu, T.; Yi, B.; Chen, C.; and Bohg, J. 2022. Category-independent articulated object tracking with factor graphs. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3800–3807. IEEE.

Jain, A.; Lioutikov, R.; Chuck, C.; and Niekum, S. 2021. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13670–13677. IEEE.

Li, K.; Guo, D.; Chen, G.; Liu, F.; and Wang, M. 2023. Data Augmentation for Human Behavior Analysis in Multi-Person Conversations. In *Proceedings of the 31st ACM International Conference on Multimedia*, 9516–9520.

Li, K.; Guo, D.; and Wang, M. 2021. Proposal-free video grounding with contextual pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1902–1910.

Li, X.; Wang, H.; Yi, L.; Guibas, L. J.; Abbott, A. L.; and Song, S. 2020. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3706–3715.

Lin, Y.; Tremblay, J.; Tyree, S.; Vela, P. A.; and Birchfield, S. 2022. Keypoint-based category-level object pose tracking from an RGB sequence with uncertainty estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, 1258–1264. IEEE.

Liu, L.; Du, J.; Wu, H.; Yang, X.; Liu, Z.; Hong, R.; and Wang, M. 2023. Category-Level Articulated Object 9D Pose Estimation via Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, 728–736.

Liu, L.; Xu, W.; Fu, H.; Qian, S.; Yu, Q.; Han, Y.; and Lu, C. 2022a. AKB-48: a real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14809–14818.

Liu, L.; Xue, H.; Xu, W.; Fu, H.; and Lu, C. 2022b. Toward real-world category-level articulation pose estimation. *IEEE Transactions on Image Processing*, 31: 1072–1083.

Liu, Q.; Qiu, W.; Wang, W.; Hager, G. D.; and Yuille, A. L. 2020. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*.

Liu, X.; Wang, G.; Li, Y.; and Ji, X. 2022c. Catre: Iterative point clouds alignment for category-level object pose refinement. In *European Conference on Computer Vision*, 499–516. Springer.

Liu, Y.; Liu, Y.; Jiang, C.; Lyu, K.; Wan, W.; Shen, H.; Liang, B.; Fu, Z.; Wang, H.; and Yi, L. 2022d. HOI4D: A 4D egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21013–21022.

Manhardt, F.; Wang, G.; Busam, B.; Nickel, M.; Meier, S.; Minciullo, L.; Ji, X.; and Navab, N. 2020. CPS++: Improving class-level 6D pose and shape estimation from monocular images with self-supervised learning. *arXiv preprint arXiv:2003.05848*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Romero, J.; Tzionas, D.; and Black, M. J. 2017. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6): 1–17.

Wang, C.; Martín-Martín, R.; Xu, D.; Lv, J.; Lu, C.; Fei-Fei, L.; Savarese, S.; and Zhu, Y. 2020. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 10059–10066. IEEE.

Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; and Guibas, L. J. 2019a. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2642–2651.

Wang, X.; Zhou, B.; Shi, Y.; Chen, X.; Zhao, Q.; and Xu, K. 2019b. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8876–8884.

Wen, B.; and Bekris, K. 2021. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8067–8074. IEEE.

Wen, B.; Tremblay, J.; Blukis, V.; Tyree, S.; Müller, T.; Evans, A.; Fox, D.; Kautz, J.; and Birchfield, S. 2023. BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 606–617.

Weng, Y.; Wang, H.; Zhou, Q.; Qin, Y.; Duan, Y.; Fan, Q.; Chen, B.; Su, H.; and Guibas, L. J. 2021. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13209–13218.

Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; et al. 2020. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11097–11107.

Xiong, H.; Fu, H.; Zhang, J.; Bao, C.; Zhang, Q.; Huang, Y.; Xu, W.; Garg, A.; and Lu, C. 2023. RoboTube: Learning Household Manipulation from Human Videos with Simulated Twin Environments. In *Conference on Robot Learning*, 1–10. PMLR.

Xue, H.; Liu, L.; Xu, W.; Fu, H.; and Lu, C. 2021. OMAD: Object Model with Articulated Deformations for Pose Estimation and Retrieval.

Yang, L.; Li, K.; Zhan, X.; Lv, J.; Xu, W.; Li, J.; and Lu, C. 2022a. ArtiBoost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2750–2760.

Yang, L.; Li, K.; Zhan, X.; Wu, F.; Xu, A.; Liu, L.; and Lu, C. 2022b. OakInk: A Large-scale Knowledge Repository for Understanding Hand-Object Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20953–20962.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2018. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847*.