ICK-Track: A Category-Level 6-DoF Pose Tracker Using Inter-Frame Consistent Keypoints for Aerial Manipulation

Jingtao Sun, Yaonan Wang, Mingtao Feng, Danwei Wang, Jiawen Zhao, Cyrill Stachniss, Xieyuanli Chen*

***Abstract*— Robots that are supposed to interact with or manipulate objects in the world must be able to track the poses of objects in their sensor data. Thus, Detecting and tracking the 6-DoF poses of targeted objects is important for aerial manipulation and is still in the early stage due to the high dynamics and limited onboard capacity of such systems. In this paper, we propose ICK-Track, a novel method for onboard category-level object 6-DoF pose tracking that can be applied to aerial manipulation without using any pre-defined object CAD models. It first utilizes a semi-supervised video segmentation to detect objects in the eye-in-hand RGB-D camera stream to segment the 3D points of objects. Then, canonical keypoints are extracted using iterative farthest point sampling. We propose a novel inter-frame consistent keypoints generation network to generate the corresponding keypoint pairs, which are used together with ICP to estimate the pose changes of objects for tracking. Experimental results show that our method is more robust to viewpoint changes and runs faster than the state-of-the-art methods on category-level pose tracking. We further test our proposed method on a real aerial manipulator. A demo video showing the use of our method on a real aerial manipulator and the implementation of our method are available at: https://github.com/S-JingTao/ICK-Track.**

I. Introduction

Estimating accurate 6 degree-of-freedom (6-DoF) poses of objects in the 3D space plays a central role in many applications over the past decades, such as human-robot interaction [11], autonomous driving [31], augmented reality [29], and manipulation [16]. Previous works have already explored instance-level 6-DoF pose estimation, often requiring corresponding CAD models of objects with shape and size information [21], [23]. Despite good tracking results, the dependence on 3D models hinders this type of method for flexible applications, e.g., using objects that have never been seen before or objects with no CAD model available. Category-level 6-DoF object pose estimation [9], [20] constantly gains more attention. It leverages temporal consistency to dynamically estimate and track 6-DoF object poses over image sequences without using predefined models.

Although a large number of works have been done, the online object 6-DoF pose tracking for aerial manipulation is still challenging due to the limited onboard resources and high-speed requirements of UAVs. Recently, RGB-D sensors have been used more frequently than before for aerial manipulation [12], [34] due to their favorable size and weight for UAVs, and furthermore providing both color and depth information. In this paper, we aim at developing a lightweight and robust 6-DoF pose tracker on RGB-D data that generalizes to new objects without using predefined 3D models. Existing RGB-D based object 6-DoF pose tracking methods [26], [32] do not perform well for aerial manipulation when deployed on real UAVs due to the limited onboard resources and fast-changing views hindering the tracking performance.

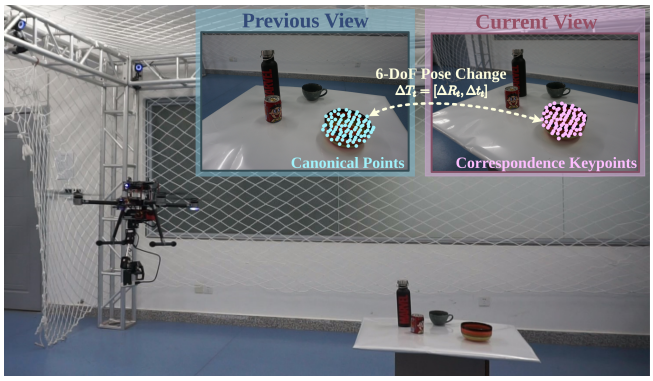

Fig. 1: Applying our proposed novel category-level 6-DoF object pose tracking method on an aerial manipulator. Our method exploits neural networks to generate inter-frame consistent keypoints from RGB-D frames to track the object pose changes for the aerial manipulation task.

The main contribution of this paper is a novel category-level 6-DoF object pose tracker for aerial manipulators. To deal with the above-mentioned problems, our proposed method tracks a set of corresponding keypoints of inter-frames and estimates the changes in the object pose observed by an RGB-D image stream during the aerial operations, as shown in Fig.1. Our approach first exploits a semi-supervised video segmentation network to semantically segment the objects in the RGB-D data and generates point clouds of target objects in the video stream. Then, it extracts canonical keypoints using the iterative farthest point sampling algorithm in the previous frame, and feeds together with the corresponding object points in the current frame to the proposed inter-frame consistent keypoints generation network to construct keypoint pairs. In the end, those keypoint pairs are used together with the iterative closest point (ICP) algorithm to estimate the pose changes of objects between two frames.

J. Sun, Y. Wang, M. Feng, J. Zhao are with the College of Electrical and Information Engineering, and the National Engineering Laboratory for Robot Visual Perception and Control (RVC-NATIONAL ENGNEERING LAB), Hunan University, Changsha 410082, China.

D. Wang is with School of Electrical and Electrical Engineering, Nanyang Technological University, Singapore.

C. Stachniss and X. Chen are with University of Bonn, Germany.

* corresponding author: xieyuanli.chen@igg.uni-bonn.de

Research was supported in part by the National Natural Science Foundation of China under Grant 61903135, Grant 61803089, and Grant 61733004, in part by Natural Science Foundation of Hunan Province under Grant 2020JJ5090.

By repeating this procedure, our method achieves category-level object 6-DoF pose tracking for aerial manipulation.

In sum, we make the following three key claims. Our approach is able to: (i) track the 6-DoF pose of objects without using predefined 3D CAD models, and generalize better to different category objects than the state-of-the-art methods, (ii) operate online on a real aerial manipulator, and (iii) be more robust to fast view changes with a higher success rate of object pose tracking. All claims are backed up by the paper and our experimental evaluation.

II. Related Work

There are many works have been done for 6-DoF object pose tracking, which can be divided into two groups, instance-level and category-level approaches [22]. The difference between instance-level and category-level is that instance-based methods estimate 6-DoF poses of *seen* objects, which usually works with the known 3D CAD model of the object, while category-level focus on estimating the 6-DoF poses of *unseen* objects, whose models are not available during operation.

A large number of scientific work have been done for instance-level approaches [2], [8], [13], [27], [30], [33], where the target object instances are known, and pre-defined 3D CAD models are needed in the process of network training and testing. Conventional methods also treat this task as template matching or object positions regression techniques, where they align the target point cloud to a 3D CAD model using registration approaches such as the iterative closest point (ICP) [8] algorithm or based on hand-crafted feature descriptors [2]. Recently, deep neural networks have been widely used for this task. For example, Wang et al. [27] propose DenseFusion, which employs CNN to extract features from RGB-D data to estimate the 6D pose of a set of known objects. Xiang et al. [33] propose PoseCNN estimating the 3D translation of an object by localizing its center in the RGB image, and the 3D rotation is estimated by regressing to a quaternion representation. Li et al. [13] propose a deep neural network for 6-DoF object pose matching named DeepIM, which iteratively refines the pose by matching the rendered image against the observed image. Similarly, Wen et al. [30] also network to track the poses of objects in the RGB-D data. Their network is trained only with synthetic data and can work over real images.

In contrast to the instance-level methods, category-level approaches estimate the pose of unseen object instances in the scene without relying on pre-defined CAD models. One pioneer work is proposed by Wang et al. [28], which uses a normalized object coordinate space (NOCS) feature as the shared canonical representation used for the 6-DoF pose tracking of different objects in a particular category. Several works have also been proposed to improve NOCS. For example, Li et al. [14] further develop an articulation-aware normalized coordinate space hierarchy to achieve 6-DoF pose tracking specifically for articulated objects. Chen et al. [3] propose the canonical shape space feature, a unified representation for a variety of instances of a certain object category to improve the estimation results. Nonetheless, the above-discussed methods use only single frame information to extract features and cannot leverage temporal information from the continuous stream or previous frames. Weng et al. [31] propose a multi-object tracking method on point cloud data, which uses the Kalman filter to exploit the temporal information. Based on that, Chen et al. [5] propose an online motion detection method with a deep network [4] using object tracking for autonomous driving. These methods achieve online object tracking, however, with the assumption that there is no fast change in pitch and roll of the objects on the road, which does not apply to aerial manipulation.

Recently, Wang et al. [26] propose an RGB-D video-based 6-DoF pose tracker named 6-PACK, which uses an unsupervised learning approach that detects the set of 3D keypoints for tracking. Similarly, Weng et al. [32] propose CAPTRA that handles both 6-DoF pose and corresponding bounding box tracking for rigid-body object instances exploiting sequential information. Our method also exploits temporal information between two continuous frames to generate keypoint pairs. However, different from 6-PACK using general keypoints for all instances, our method generates for each instance specific keypoint pairs using our proposed ICK network. CAPTRA produces pose-canonicalized point clouds to achieve pose regression between different scales, while our method uses spherical normalization to handle different sizes and scales of points.

Although both 6-PACK and CAPTRA achieve 6-DoF object pose tracking, they are not designed for aerial manipulation. Both methods focus more on ground-based robots and do not work so well when run on a UAV. Very few works in the literature focus on the 6-DoF object pose tracking for aerial manipulators. Most existing works are either for tracking the UAV's trajectory [7] or tracking the end effector of the aerial manipulators. The most related work to our method is the one by Kumar et al. [10], which enables an aerial manipulator to perform multiple complicated tasks using deep learning networks including the tracking of the objects observed in the camera stream. However, it only achieves object tracking in the image level but not the 6-DoF pose tracking of the objects. Our work presents a novel deep neural network-based method and demonstrates that our method achieves category-level 6-DoF object pose tracking on a real aerial manipulator.

III. Our Approach

A. Problem Formulation

Our work aims to continuously detect and track the 6-DoF poses of target objects from known categories during the flight of an aerial manipulator, where the onboard camera is set to eye-in-hand mode. We assume that the object instances are within visual range. Taking one object as an example, with the initial pose in 3D space $\mathcal{T}_0 \in \mathbf{SE}(3)$, our problem can be defined as: given the stream of RGB-D frames $\{I_t\}_{t\geq 1}$, the goal is to track the poses of the targeted object within the frames in an online manner, as shown in Fig.1. More specifically, given two consecutive

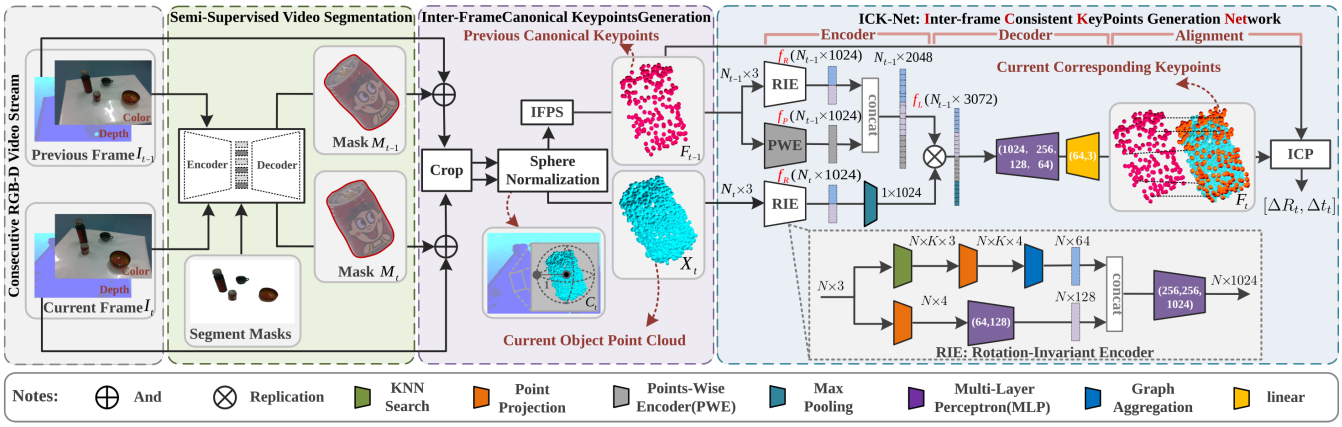

Fig. 2: Overview of our proposed ICK-Tracker. It takes the consecutive RGB-D video stream as the inputs and firstly uses a semi-supervised video segmentation to produce pixel-wise object masks per frame to generate object point clouds. Then, the sphere normalization and IFPS are exploited to extract the canonical keypoints and normalized current object point cloud, which are fed to the proposed ICK-Net. Our ICK-Net then directly generates the corresponding inter-frame consistent keypoint pairs, and uses them with ICP to calculate the inter-frame object pose change.

frames I_{t-1} and I_t, we estimate the change of pose of object $\Delta T_t = [\Delta R_t, \Delta \mathbf{t}_t] \in \mathbf{SE}(3)$, where $\Delta R_t \in \mathbf{SO}(3)$ represents the change in rotation, and $\Delta \mathbf{t}_t \in R^3$ is the change of translation. The absolute pose can be calculated by applying recursively the last change of pose of each object:

$$T_t = \Delta T_t \cdot T_{t-1} = \Delta T_t \cdot \Delta T_{t-1} \cdot \ldots \cdot T_0. \quad (1)$$

B. Overview of the Proposed Method

An overview of our proposed category-level 6-DoF pose tracking framework is depicted in Fig. 2. It consists of three main modules, video segmentation, canonical feature generation, and inter-frame consistent keypoint generation. Following the initialization introduced in [26], our method also assumes the targeted objects together with their initial poses are given. Taking one object as an example, it starts at the location indicated by a given initial pose T_0. For the next every two continuous RGB-D frames I_t and I_{t-1}, our method firstly feeds them to a semi-supervised video segmentation network to obtain pixel-wise object masks M_t and M_{t-1}, for both the current and previous frames. Using the predicted object masks, we filter out the background points generated from the RGB-D sensor and only use the points belonging to the targeted objects as the input for the following procedures. Based on the points of the objects, our method then utilizes the iterative farthest point sampling (IFPS) [6] algorithm to generate canonical keypoints F_{t-1} in the previous frame I_{t-1}. They are generated based on geometric information and used to guide the following keypoint extraction network yielding more robust keypoint pairs. In the third step, the canonical keypoints F_{t-1} of the previous frame and the point cloud of the objects X_t in the current frame are fed into our proposed ICK network for generating the corresponding keypoints F_t in the current frame. For each keypoint in F_{t-1} of the previous frame, our network generates one corresponding keypoint in F_t. In the end, the keypoint matches of the current and previous frames are passed to the iterative closest point [1] algorithm, a point-based registration method, to calculate the inter-frame object 6-DoF pose changes.

C. Semi-Supervised Video Object Segmentation

The first step is to use an object segmentation network to separate the objects from the background in the RGB-D image stream. Unlike ground robots, the observations of the onboard camera in the eye-in-hand mode change rapidly during the flight of the aerial robot and the operation of the manipulator. This challenge breaks the assumption of most existing video object segmentation approaches, which rely on temporal consistency, assuming that objects do not change too much between one frame and the next. To overcome this, we adopt a per-frame video object segmentation method by Maninis et al. [15], which separates an object from the background in the video in a continuous manner given the object mask in the first frame and is able to overcome the occlusions or missing frames caused by the camera motion in a certain range. To automatically generate the object mask for the first frame, we use an RGB-D based instance segmentation method by Papon et al. [17]. It uses a seeding methodology based on 3D space and a flow-constrained local iterative clustering using color and geometric features.

D. Inter-Frame Canonical Feature Generation

Given the object mask from the video object segmentation network, we then separate the raw object points out of the whole point cloud generated by the RGB-D sensor by checking the pixel-point correspondences. Based on object points, the goal of this module is to extract a set of canonical keypoints that represents the geometric features of the object at the previous frame. We denote the original object points as $C_t^i = \{p_j \in \mathbb{R}^3\}_{j=1}^{N_P}$, where N_P is the number of the points, and i represents the i-th object in the frame I_t. We omit the index i from now on for notation simplicity. To make the keypoints robust against different scales for all object points, we normalize them into a unit sphere space for the

same objects in different frames, and use X_t to represent the normalized object points by:

$$\begin{cases} X_t = (C_t - \bar{\mathbf{p}})/\sigma \\ \bar{\mathbf{p}} = \frac{1}{N_P} \sum_{\mathbf{p}_j \in C_t} \mathbf{p}_j \\ \sigma = \max\limits_{\mathbf{p}_j \in C_t} \{\|\mathbf{p}_j - \bar{\mathbf{p}}\|_2\} \end{cases}, \tag{2}$$

where $\bar{\mathbf{p}}$ is the centroid of the raw object points C_t, and σ is the scale factor that transforms the original points into the unit scale.

After the normalization, we then use the IFPS algorithm to sample the object points X_t and extract a sparse set of keypoints F_t. IFPS [6] is a classical method to extract canonical keypoints from a point cloud based on the idea of repeatedly placing the next sample point in the middle of the least-known area of the sampling domain. We refer more details to the original paper [6]. Here, we only extract canonical keypoints on the previous frame and use them together with the raw object points in the current frame as the input to our ICK network to generate corresponding keypoints in the current frame.

E. Inter-Frame Consistent Keypoints Generation Network

For estimating the pose changes of objects between two frames, the previous approaches [25], [26] usually have two steps. They first extract keypoints in each frame independently and then match them for the pose estimation. In this way, however, the network does not fully exploit the inter-frame information in keypoint generation and matching. Different from the two-step methods, we treat the 6-DoF object pose tracking as an inter-frame corresponding keypoints generation and registration problem. Our approach directly generates the keypoint pairs without explicitly matching them, making the proposed method more robust to wrong matches.

As mentioned before, a set of canonical keypoints is generated for the previous frame by using IFPS. Based on that, we propose an inter-frame consistent keypoints generation network, called ICK-Net. It learns the correspondence among intra-class instances and generates for each canonical keypoint in the previous frame a corresponding keypoint from the raw object points in the current frame. Unlike prior work on keypoint detection individually on each frame, our proposed method maintains canonical keypoints from the previous frame as a precondition, which provides references for current corresponding keypoints reconstruction so as to enable adjacent data association together with the latest observation.

As depicted in the right part of Fig. 2, we take both the canonical keypoints F_{t-1} of the previous frame together with the normalized object points of the current frame X_t as the input to our ICK-Net. To encode the temporal-spatial information from both parts, we propose a novel rotation invariant encoder (RIE). Inspired by Sun et al. [24], we design the RIE as two branches, as shown in Fig. 2. In the main backbone (lower branch), it first uses a point projection operation (PPO) to map the 3D coordinates of keypoints into a 4-dimensional features Ω by:

$$\Omega = \{f(\alpha, x_1), f(\alpha, x_2), \dots f(\alpha, x_N)\} \in \mathbb{R}^{N \times 4}, \tag{3}$$

where, α represents the three new axes $(\alpha_1, \alpha_2, \alpha_3)$, and $f(\alpha, x_i)$ denotes the point projection as:

$$f(\alpha, x_i) = (\cos(\alpha_1, x_i), \cos(\alpha_2, x_i), \cos(\alpha_3, x_i), |x_i|). \tag{4}$$

After generating rotation-invariant 4D features for each keypoint, RIE uses a multiLayer perceptron (MLP) layer to further abstract pointwise features.

The upper side branch aims to extract features from local regions. It exploits the K nearest neighbor points of a keypoint and uses a PPO and graph aggregation operation (GAO) to form a local feature. GAO consists of a graph convolutional layer and a max-pooling layer to update features and encode a descriptor that contains the local neighbor information. We refer the reader to [24] for more details of GAO. In the end, the features from two branches are concatenated to generate rotation-invariant features with size of 1024. For the point-wise encoder, we use the one from PointNet by Qi et al. [19] to generate features for each keypoint.

The proposed ICK-Net first applies an RIE and a pointwise encoder (PWE) separately on the canonical keypoints F_{t-1} of the previous frame to obtain rotation-invariant features f_{t-1}^R with size of $N_{t-1} \times 1024$ and pointwise features f_{t-1}^P with the same size respectively. It then concatenates the f_{t-1}^R and f_{t-1}^P to generate a global feature vector for each canonical keypoint in the previous frame. In parallel, another RIE is also applied to the object points X_t in the current frame and generates rotation-invariant features f_t^R, followed by a max-pooling layer to obtain a global representation for the whole current object points. Then we replicate this representation N_{t-1} times and concatenate it with the features from the previous keypoints to form a global latent vector f_L in size of $N_{t-1} \times 3072$.

After all encodings, we get the feature f_L. It combines both the current and previous, local and global features, and contains rich inter-frame information. We then use a decoder consisting of several MLPs and a linear layer to integrate the latent feature map into an ordered set of correspondence keypoints F_t; thus, the generated keypoints in the current frame are one-to-one matched with the canonical keypoints in the previous frame. Our method then calculates the relative pose between them via the ICP algorithm. Conducting this procedure for every incoming two frames, our method achieves tracking objects through the RGB-D image stream.

F. Loss Function

There are two parts in our loss function: inter-frame keypoint consistency loss and inter-frame pose estimation loss. The inter-frame keypoint consistency loss measures the difference between the keypoints generated in consecutive frames. The inter-frame pose estimation loss directly takes the difference in poses between the ground truth and estimated ones as the loss.

For the inter-frame keypoint consistency loss L_K, the objective is to minimize the residuals between the generated keypoints of the current frame to the corresponding canonical keypoints in the previous view after transforming them into the same viewpoint using the ground truth relative pose. We take the average residual over all pairs as the final loss, which can be formalized as:

$$L_K = \frac{1}{N_{t-1}} \sum_{k=1}^{N_{t-1}} \left\| \Delta \hat{T}_t \cdot \mathbf{p}_{t-1}^k - \mathbf{p}_t^k \right\|, \tag{5}$$

where $\Delta \hat{T}_t = [\Delta \hat{R}_t | \Delta \hat{\mathbf{t}}_t]$ is the ground truth relative pose between $t-1$ frame and t frame. N_{t-1} is the number of keypoints extracted by our network.

To guide the network learning to estimate the change of pose, we further add an inter-frame pose estimation loss, which consists of both the translation loss L_t and the rotation loss L_R:

$$L_t = \left\| \Delta \hat{\mathbf{t}}_t - \Delta \mathbf{t}_t \right\|, \tag{6}$$

$$L_R = 2 \arcsin \left(\frac{1}{2\sqrt{2}} \left\| \Delta \hat{R}_t - \Delta R_t \right\|_F \right), \tag{7}$$

where ΔR_t and $\Delta \mathbf{t}_t$ are the estimated inter-frame relative pose based on the corresponding keypoint sets using ICP. $\| \cdot \|_F$ is the Frobenius norm to calculate the difference between two rotation matrices. To avoid the ambiguities brought by symmetric instances, we adopt the strategy introduced by Wang et al. [26] to redefine the inter-frame consistency loss and rotation loss for symmetric categories.

In summary, the overall loss function is:

$$L = \mu L_K + (1 - \mu)(L_t + L_R), \tag{8}$$

where $0 \leq \mu \leq 1$ is a weight factor.

G. Implementation Details

We implement our network based on the PyTorch library [18]. All the building blocks are trained using an ADAM optimizer with an initial learning rate of 10^{-3} and a batch size of 20. The training epoch number is set as 100. We employ batch normalization and RELU activation units in every MLP. We fine-tune the OSVOS network [15] for the video segmentation and the inputs size of canonical keypoints F_{t-1} from previous frame is set to $N_{t-1} = 500$.

The experiments on the public datasets are conducted on a desktop computer with an Intel Xeon Gold 6226R@2.90GHz processor and a single NVIDIA RTX A6000 GPU. We furthermore deploy the proposed method on a real aerial manipulator equipped with a Pixhawk-v2 flight controller, an NVIDIA Jetson Xavier as the core controller. We design a series structured 4-DoF manipulator using the ROBOTIS Dynamixel MX-28 servo motors. For the eye-in-hand configuration, we use an Intel RealSense D435i camera. An OptiTrack Motion Capture System measures the position and yaw orientation of the robot, and detailed data flow and configuration can be seen in Fig. 4.

IV. Experimental Evaluation

We present our experiments to show the capabilities of our method and to support the claims that our approach is able to: (i) track the 6-DoF pose of objects without using predefined 3D CAD models and generalize better to different category objects than the state-of-the-art methods, (ii) operate online on a real aerial manipulator, and, (iii) be more robust to quick-changing view with a higher success rate of object pose tracking.

A. Experimental Setup

Dataset. We evaluate different methods using both a public dataset and also on the data collected by our own using a real aerial manipulator. There are not many real-world benchmark datasets for category-level object 6-DoF pose tracking, and we use the NOCS-REAL275 dataset by Wang et al. [28]. It contains six categories: bottle, bowl, camera, can, laptop and mug. This dataset consists of two parts: The real RGB-D videos and synthetic rendering objects. The real videos are with ground truth object poses depicting in total three instances of objects of each category. Following Wang et al. [26], we use seven real videos together with all synthetic data for training and six real videos for testing, which contain three different unseen instances for each object category with 3,200 frames in total. Besides the public dataset, we also deploy our tracking method to a real aerial manipulator and evaluate the performance of different methods using the RGB-D data collected by our aerial manipulator. An OptiTrack Motion Capture System provides the ground-truth poses.

Baseline methods. In all experiments, we compare our method to two state-of-the-art category-level 6-DoF pose tracking methods, 6-PACK by Wang et al. [26] and CAPTRA by Weng et al. [32]. 6-PACK uses the anchor-based keypoints for category-level 6-DoF pose tracking. CAPTRA directly predicts pose changes using the point cloud at the current frame and the estimated pose from the last frame. For comparison on the public dataset, we directly use the results reported in the original papers of the baseline methods. For experiments on our real aerial manipulator, we use their open-sourced implementations with default parameters to generate the 6-DoF pose tracking results.

Evaluation Metrics. In line with the baseline methods, we use the following metrics for fair comparisons: (1) $5°5\,cm$ refers to the percentage of estimating and tracking results with orientation error $< 5°$ and translation error $< 5\,cm$; (2) IoU25 represents the percentage of volume overlap between the prediction and ground-truth 3D bounding box that is larger than 25%; (3) R_{err}, means of the orientation error in degrees, and (4) T_{err}, means of the translation error in centimeters.

B. Evaluation Results on the NOCS-REAL275 Dataset

The first experiment evaluates our method on the public NOCS-REAL275 dataset. The results support our first claim that our method tracks the 6-DoF pose of objects without

TABLE I: Quantitative results on the NOCS-REAL275 dataset. With the metrics of $5°5\,cm$ and IoU25, the higher value the better. On contrary, the lower the better for the R_{err} and T_{err} metrics. The baseline results are taken from their original papers.

Methods	Metrics	Bottle	Bowl	Camera	Can	Laptop	Mug	Average
6-PACK [26]	$5°5\,cm\uparrow$	24.5	55.0	10.1	22.6	63.5	24.1	33.3
	IoU25 $\uparrow$	**91.1**	**100.0**	87.6	**92.6**	**98.1**	**95.2**	**94.2**
	$R_{err}\downarrow$	15.6	5.2	35.7	13.9	4.7	21.3	16.0
	$T_{err}\downarrow$	4.0	1.7	5.6	4.8	2.5	2.3	3.5
CAPTRA [32]	$5°5\,cm\uparrow$	**79.5**	**79.2**	0.41	64.7	94.0	55.1	62.2
	IoU25 $\uparrow$	72.1	79.6	2.5	62.5	87.2	80.7	64.1
	$R_{err}\downarrow$	**3.3**	**3.5**	17.8	3.4	**2.2**	**5.4**	5.9
	$T_{err}\downarrow$	**2.6**	**1.4**	35.5	5.7	**1.5**	**0.8**	7.9
Ours	$5°5\,cm\uparrow$	78.2	73.4	**89.5**	**79.3**	**96.1**	**89.9**	**84.4**
	IoU25 $\uparrow$	73.6	66.3	**87.8**	67.4	85.6	78.9	76.6
	$R_{err}\downarrow$	4.7	6.2	**1.9**	**2.3**	5.7	6.0	**4.5**
	$T_{err}\downarrow$	3.4	4.6	**1.4**	**3.8**	2.7	2.5	**3.1**

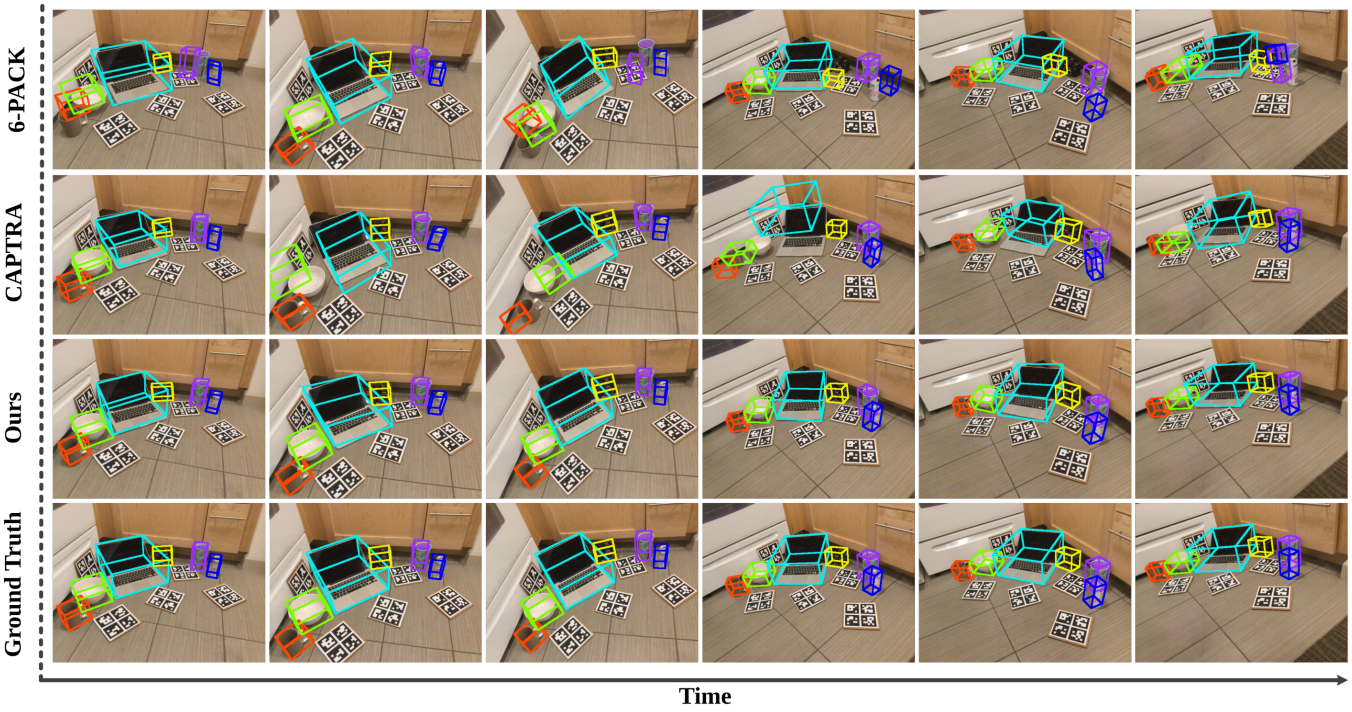

Fig. 3: Qualitative visualization sample of our ICK-Track and representative comparison baselines on NOCS-REAL275 dataset.

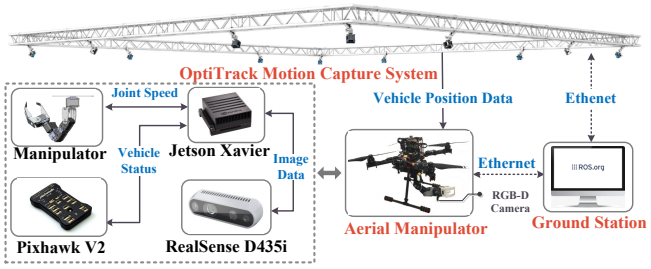

Fig. 4: Illustration of our real aerial manipulator platform and experiment setup for online pose tracking for aerial manipulation.

using predefined 3D CAD models, and generalize better to different category objects than the state-of-the-art methods.

Quantitative results and qualitative visualization on the testing set of the NOCS-REAL275 dataset are depicted in Tab. I and Fig. 3 respectively. As shown in Tab. I, our method outperforms the state-of-art methods in terms of $5°5\,cm$, R_{err} and T_{err}. Note that, the baseline methods sometimes fail in tracking certain types of objects. In contrast, our method successfully tracks all different category objects, which shows the robustness and generalization ability of our proposed method. Furthermore, our method performs generally well in all metrics indicating that our method can both estimate good pose changes of objects as well as track the shape of the objects. The qualitative results are shown in Fig. 5, which also illustrates that our method is more robust than baseline methods and does not lose track of objects caused by rapid changes in view.

C. Experiments on an Aerial Manipulator

The results of the second experiment support our last two claims that our method can operate online on a real aerial manipulator and is more robust to fast view changes with a higher success rate of tracking than baseline methods. In

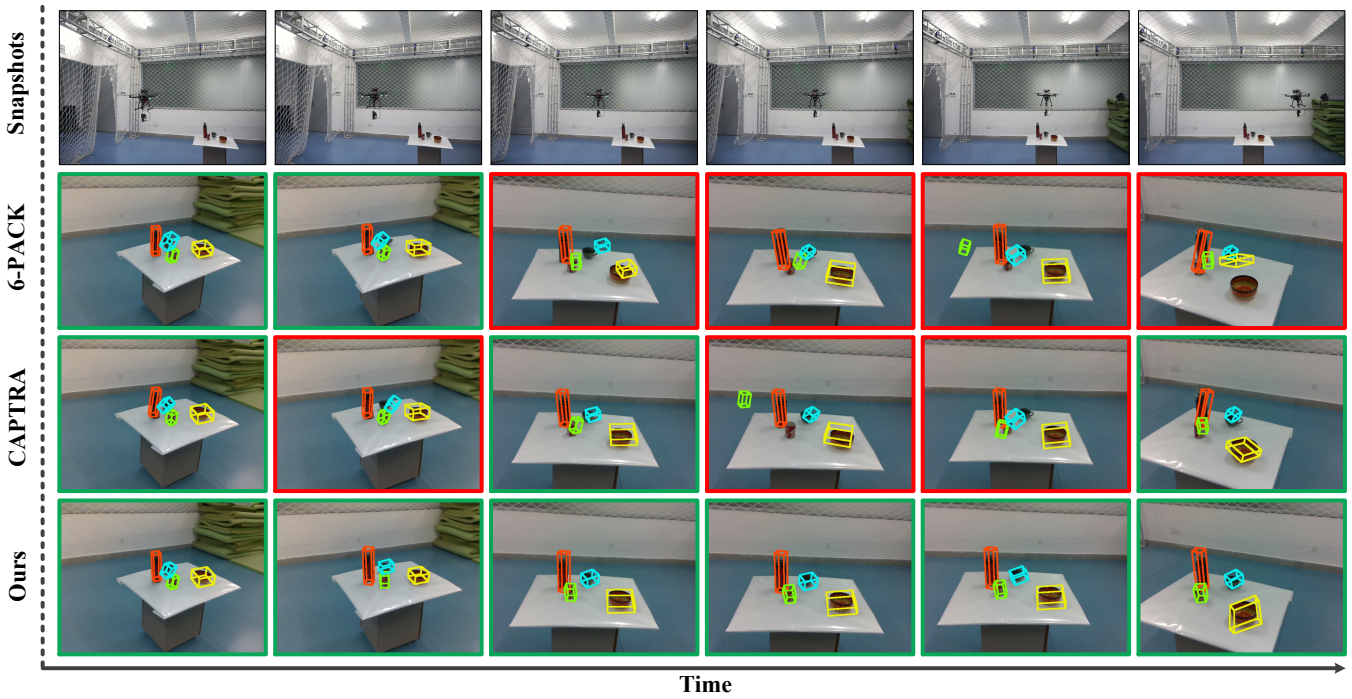

Fig. 5: Evaluation results on object pose tracking with a real aerial manipulator. The upper row illustrates the real flying status. The lower three rows show the pose tracking results of different methods. The green boundary represents a successful tracking frame, while the red boundary represents a failure case.

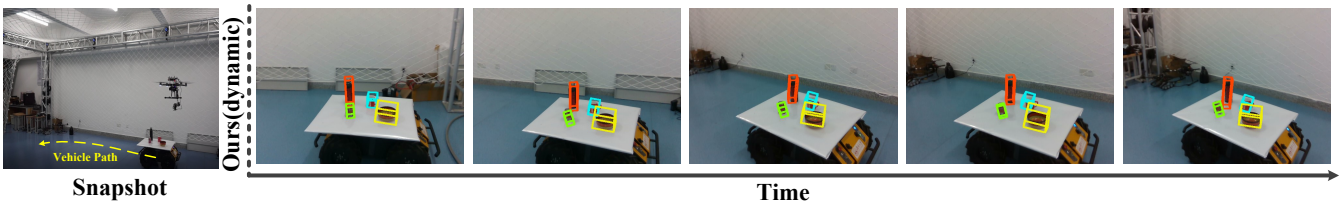

Fig. 6: Qualitative results on dynamic object pose tracking in an air-ground robots collaborative scenario, where the aerial manipulator follows a ground vehicle and tracks the onboard objects. It is a highly dynamic scenario, and only our method works.

TABLE II: Pose tracking runtime and success rate evaluated on the onboard platform

Methods	6-PACK	CAPTRA	Ours
FPS	5.22	9.67	**11.38**
Success Rate [%]	70.32	75.46	**96.87**

this experiment, we deploy our method trained on the public dataset directly to a real-world aerial manipulator platform, as shown in Fig. 4. A scenario for aerial 6-DoF pose tracking task is designed as shown in the top row of Fig. 5, where the task consists of two phases: the aerial manipulator takes off and hovers where the target object can be detected, and then it flies around the object to find a good picking up position, while the 6-DoF pose tracking method works online. We compare the tracking performance on several unseen instances of known categories, such as bowl, bottle, can, and mug.

Visualization results in Fig. 5 demonstrate that our proposed tracker successfully generalizes to unseen instances in real scenarios with a high success rate. For each object in each RGB-D frame, if the difference between the estimated center and the ground truth center is smaller than its radius, we count it as correct tracking. When all the objects inside one frame are correctly tracked, we count that frame as a success tracking frame. In Fig. 5, the success tracked frames are labeled with *green* boundary and failed one with *red* boundary. As can be seen, our method achieves a higher success rate than the baseline methods when applied to a real aerial manipulator.

The same conclusion can also be drawn from statistical results reported in Tab. II. We calculate the success tracking rate using 1500 frames collected from one real flight, and the success rate of our method is much higher (more than absolute 20%) than other methods. Meanwhile, we test the runtime of all methods on the real UAV platform as shown in II. Our method takes on average 11.38 Hz with less than 12.5% of the GPU storage (16 GB in total), while the actual speed of both 6-PACK and CAPTRA is slower than 10 FPS.

We further test our method in a more challenging air-ground robots collaborative scenario shown in Fig. 6, where the aerial manipulator follows a ground vehicle and tracks the onboard objects. It is a highly dynamic scenario where only our method works while the other two baselines fail

to track the objects. A video of our real flight demo can be found here: https://github.com/S-JingTao/ICK-Track.

V. Conclusion

In this work, we presented our proposed ICK-Tracker, a category-level 6-DoF object pose tracking method for aerial manipulation. Our tracker is based on a novel inter-frame consistent keypoints generation network that generates consistent keypoint pairs for different instances of the same category between consecutive frames to estimate the inter-frame pose changes. Experiments on both, a public dataset and a real aerial manipulator, demonstrate that our method achieved comparable performance to state-of-art baselines and successfully tracked all different category objects. We furthermore test our method on an aerial manipulator platform, and the experimental results show that our method is more robust than the state-of-the-art methods in tracking unseen objects under fast-changing views with faster runtime and a much higher success tracking rate.

References

[1] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, PAMI-9(5):698–700, 1987.

[2] P. Azad, D. Münch, T. Asfour, and R. Dillmann. 6-dof model-based tracking of arbitrarily shaped 3d objects. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[3] D. Chen, J. Li, Z. Wang, and K. Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss. Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)*, 6:6529–6536, 2021.

[5] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss. Automatic Labeling to Generate Training Data for Online LiDAR-based Moving Object Segmentation. *arXiv preprint*, 2022.

[6] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Trans. on Image Processing*, 6(9):1305–1315, 1997.

[7] S.A. Emami and A. Banazadeh. Simultaneous trajectory tracking and aerial manipulation using a multi-stage model predictive control. *Aerospace Science and Technology*, 112, 2021.

[8] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[9] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[10] A. Kumar, M. Vohra, R. Prakash, and L. Behera. Towards deep learning assisted autonomous uavs for manipulation tasks in gps-denied environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[11] S. Lemaignan, F. Garcia, A. Jacq, and P. Dillenbourg. From real-time attention assessment to "with-me-ness" in human-robot interaction. In *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, pages 157–164, 2016.

[12] L. Li, T. Zhang, H. Zhong, H. Li, H. Zhang, S. Fan, and Y. Cao. Autonomous removing foreign objects for power transmission line by using a vision-guided unmanned aerial manipulator. *Journal of Intelligent and Robotic Systems (JIRS)*, 103(2):1–14, 2021.

[13] P. Li, T. Qin, and S. Shen. Stereo Vision-based Semantic 3D Object and Ego-motion Tracking for Autonomous Driving. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[14] X. Li, H. Wang, L. Yi, L.J. Guibas, A.L. Abbott, and S. Song. Category-level articulated object pose estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] K.K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 41(6):1515–1530, 2018.

[16] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi. Past, present, and future of aerial robotic manipulators. *IEEE Trans. on Robotics (TRO)*, 38(1):626–645.

[17] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 32, 2019.

[19] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] C.R. Qi, W. Liu, C. Wu, H. Su, and L.J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2013.

[22] C. Sahin, G. Garcia-Hernando, J. Sock, and T.K. Kim. Instance-and category-level 6d object pose estimation. In *RGB-D Image Analysis and Processing*, pages 243–265. 2019.

[23] C. Sahin and T.K. Kim. Recovering 6d object pose: a review and multi-modal analysis. In *Proc. of the Europ. Conf. on Computer Vision (ECCV) Workshops*, 2018.

[24] X. Sun, Z. Lian, and J. Xiao. Srinet: Learning strictly rotation-invariant representations for point cloud classification and segmentation. In *Proc. of the ACM Intl. Conf. on Multimedia*, 2019.

[25] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2018.

[26] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 10059–10066, 2020.

[27] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L.J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] J. Wei, G. Ye, T.R. Mullen, M. Grundmann, A. Ahmadyan, and T. Hou. Instant motion tracking and its applications to augmented reality. In *Proc. of the CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, 2019.

[30] B. Wen, C. Mitash, B. Ren, and K.E. Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[31] X. Weng, J. Wang, D. Held, and K. Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 10359–10366, 2020.

[32] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L.J. Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2021.

[33] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[34] X. Zhang, Y. Zhang, P. Liu, and S. Zhao. Robust localization of occluded targets in aerial manipulation via range-only mapping. *IEEE Robotics and Automation Letters (RA-L)*, 2022.