Uncertainty-Aware Semi-Supervised Semantic Key Point Detection via Bundle Adjustment

Kai Li[1,2], Yin Zhang[2], and Shiyu Zhao[2]

***Abstract*— Visual relative localization is widely used in multi-robot systems. While semantic key points offer a promising solution for 6DoF pose estimation, manual data labeling for network training remains unavoidable. In this paper, we introduce a novel method that jointly estimates the semantic key point detection model and 6DoF camera pose. Our key idea is to leverage the 3D-2D projection to produce pseudo labels for detection model training while taking the key point predictions as landmarks for 6DoF camera pose estimation. Compared with state-of-the-art works, our method eliminates the need for calibration and time synchronization of multi-camera systems, requiring only a handful of manually labeled data, which significantly improves the training efficiency. The experiment validates the effectiveness and practicality of our method in public datasets and real-world robotic applications. Code and data are made available[3].**

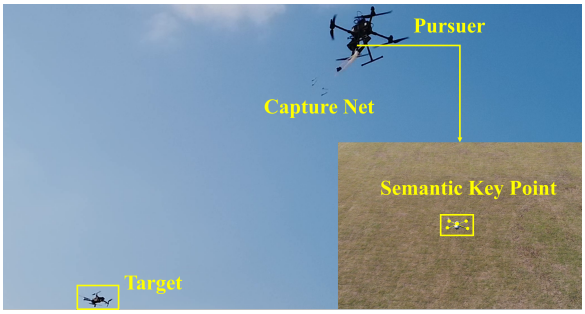


Fig. 1: Illustration of aerial pursuit. Semantic key points of the target drone are detected from the pursuer's gimbal camera image and used for relative pose estimation.

I. Introduction

Visual relative localization and pose estimation have been research topics with high interest in many autonomous robot applications such as drone detection [1]–[3], drone flocking [4], drone racing [5], multi-target tracking [6] [7], and swarm state estimation [8]. Semantic key point detection provides a promising solution to the 6DoF pose estimation problem in robotics [5], [7]–[10]. Semantic key point detection with deep neural networks classifies different types of key points on the image. Combined with a geometric pose solver, semantic key points prove to be a powerful tool for relative pose estimation in multi-robot systems.

While deep neural networks for semantic key points detection have achieved impressive results in computer vision research [11], labeling large amounts of training data requires time-consuming and error-prone manual operations. Unlike general-purpose key point detection tasks such as human pose estimation or facial landmark localization, acquiring pre-labeled public datasets for a specific type of robot is challenging. Researchers seeking to apply the semantic key point detection model to real-world robot applications must manually label data for a robot of a specific type. Therefore, how to train the key point detection model *at a low cost of labeling* is important and worth considering.

Multiple view constraints can be used to triangulate and reproject 3D key points, producing pseudo labels for both 3D and 2D key points. Many works have been proposed to leverage multiple view constraints for supervision [12]–[18]. However, most existing works of semi-supervision from multiple views assume that camera poses are known [12] [18], or are incapable of estimating the entire 6DoF camera poses [16] [17]. In addition, most previous works also require using multiple synchronized cameras to observe the target [16] [15]. Yet in practical situations, fast and convenient 6DoF camera calibration or multi-camera synchronization is not an easy task. How to jointly optimize the key point detection model and 6DoF camera poses from a time series of frames in one single camera remains an open problem.

In this paper, we present a novel method to train a semantic key point detection model in a semi-supervised manner. We fully utilize multiple view constraints and jointly optimize the detection model, 3D points, and 6DoF camera poses. We start to train the detection network from a small fraction of the training set and use reprojection to generate pseudo labels for the unsupervised data. Through an iterative training process, we finally gain the trained detection model, 6DoF camera poses, and 3D points of the target. The contributions and novelties of this paper are summarized as follows:

1) We propose a *network-and-pose* semi-supervised training strategy that optimizes both the detection model and 6DoF camera pose. This method significantly reduces the workload of pose calibration and data annotation.

2) The proposed method utilizes *only one moving* camera and takes the well-associated detection results fixed on the rigid target as visual landmarks for 6DoF pose estimation. Our method does not rely on multi-camera synchronization, calibration, or video flow tracking.

3) We apply the proposed method to real-world robot tasks, i.e. aerial pursuit and mobile robot formation, to prove

[1]College of Computer Science and Technology at Zhejiang University, Hangzhou, China.

[2]WINDY Lab, School of Engineering at Westlake University, Hangzhou, China. {likai,zhangyin,zhaoshiyu}@westlake.edu.cn

[3] Project homepage: https://github.com/WindyLab/semi-super-skp. This work was supported by the STI 2030—Major Projects (Grant No. 2022ZD0208800).

TABLE I: Comparison of features in representative works for semantic key point detection training from multiple views.

Method	Object type	3D label	2D label	Calibration	Projection model	#Cameras	Time sync.	Video tracking
MetaPose [14]	Flexible	✗	✓	✗	Perspective	Multiple	✓	✓
Anipose [13]	Flexible	✗	✓	✓	Perspective	Multiple	✓	✓
BKinD-3D [15]	Flexible	✗	✗	✓	Perspective	Multiple	✓	✓
MBW [16]	Flexible	✗	Small amount	✗	Orthogonal	Multiple	✓	✓
EpipolarPose [19]	Flexible	✗	✓	Optional	Perspective	Multiple	✓	✓
Isakov et al. [20]	Flexible	✓	✓	✓	Perspective	Multiple	✓	✓
Takahashi et al. [21]	Flexible	✗	✓	✗	Perspective	Multiple	✗	✗
S3K [12]	Rigid	✗	Small amount	✓	Perspective	Single	✗	✗
Ours	Rigid	✗	Small amount	✗	Perspective	Single	✗	✗

the effectiveness and practicality of our method.

II. Related Work

A. Visual Relative Localization and Data Labeling

Vision-based relative localization and pose estimation are extensively employed in multi-robot systems. In aerial systems, deep neural networks are utilized for 2D object detection and semantic key point detection for localization [1]–[4] and pose estimation [5], [7], [8]. Other robotic tasks also adopt visual measurements for localization and pose estimation, such as collaborative perception [9], manipulation [12], [22], and underwater robot tracking [23].

A key challenge in applying deep neural network detectors to real-world robotic tasks is data labeling. Some works that adopt deep neural networks in multi-robot systems [6]–[8] did not report the data labeling procedure for their custom robot targets. The work in [4] uses a foreground mask to generate training labels automatically. While this method is convenient for labeling 2D bounding boxes, it can not be applied to the labeling task for semantic key point detection. A simulation environment can also be used to generate data annotation automatically [9] [22]. However, performance is likely to decline due to the domain difference between the simulation and the real-world environment.

Apart from the above algorithm-based methods, an alternative way to facilitate data annotation is by adding an extra hardware system. Some works [1] [24] use ultra-wide band (UWB) systems to facilitate visual data annotation, which requires extra hardware development. In addition to UWB, the Vicon motion tracking system is also used for providing ground truth pose and building datasets for visual target tracking [25] [26]. However, the motion tracking system can not be used in an outdoor environment.

B. Supervision from Multiple Views

In the context of robotics, S3K [12] uses one camera to observe a static scene and takes multiple-view consistency to achieve self-supervised 2D key point training. S3K does not rely on a pre-calibrated synchronized multi-camera system. However, camera poses for 3D triangulation in S3K are assumed to be known.

In the context of computer vision, attempts have been made to employ multiple-view supervision [13]–[16], [18], [27] or epipolar constraints [28] [19] to realize a semi-supervised training scheme for semantic key points. All these works use multiple cameras to observe the target, and camera poses and time stamps must be precisely calibrated and synchronized. The work in [21] uses human target key points as landmarks for multi-camera pose calibration and synchronization. It relies on a pre-trained 2D detection model, which still requires manual labeling of 2D points back to its source. Other works [16], [17], [27] jointly estimate 3D semantic key points and camera rotations. However, as they use the orthogonal projection model, only 3DoF camera rotation can be estimated. Some works [16] [19] also use video frames as input and optical flow tracking to generate pseudo labels. Optical flow tracking is likely to fail when large inter-frame overlap exists and tends to drift in the long run.

Table I compares the features of different methods. The novelties of our method are as follows: First, we utilize only one moving camera to observe the target for data collection, eliminating the need for multi-camera spatial calibration or time synchronization. Additionally, we optimize 6DoF camera poses concurrently with the detection model based on the perspective model during the data collection process. Finally, our method does not rely on video flow tracking and hence performs well even when inter-frame overlap is small.

III. Proposed Method

A. Overview

In this section, we introduce the overview of our method, as shown in Fig. 2. The whole process consists of two optimization modules, 1) optimization of the semantic key point detection model and 2) optimization of camera poses and 3D points.

We define the key point detection module as

$$f(\mathbf{I}) \rightarrow \{\mathbf{H}_i\}_{i=0}^{n}, \tag{1}$$

where f refers to the detection network and $\mathbf{I} \in \mathbb{R}^{w \times h}$ denotes the input image. $\mathbf{H}_i$ refers to the heat map of an individual class of key points produced by the network, and n denotes the number of classes of the key points to be detected. k denotes the number of total frames. The initial training set is defined as

$$\mathcal{T}^{(0)} := \{\{\mathbf{I}_i, \mathbf{H}_i^n\}_{i=0}^{m}\}, \tag{2}$$

where $\mathbf{H}_i^n$ denotes the label for all key points and m denotes the number of the initial training set. In the following part, we use superscript in the bracket to represent the iteration round.

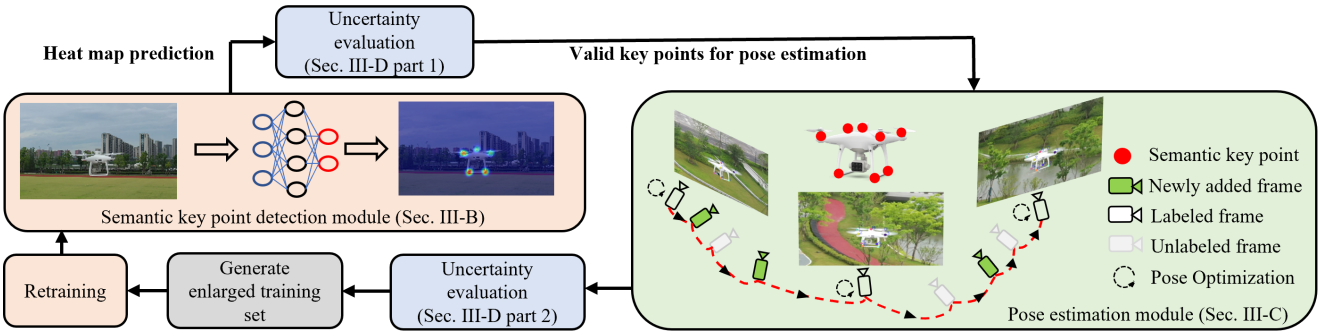


Fig. 2: Overview of our semi-supervised network-and-pose training process.

The initial training set $\mathcal{T}^{(0)}$ is used to train the initial weak detection model $f^{(0)}$. Note that $\mathcal{T}^{(0)}$ is labeled manually.

When the training of the initial weak detection model $f^{(0)}$ is completed, we take it as the detection model and perform inference on the whole training set $\mathcal{T}^{(all)}$ to produce a set of heat map predictions ${}^{(0)}\{\mathbf{H}_i\}_{i=0}^{n}$. 2D key points locations ${}^{(0)}\{\mathbf{p}_j\}_{j=0}^{n}$ are extracted from ${}^{(0)}\{\mathbf{H}_i\}_{i=0}^{n}$. Then the pose estimation module uses these points to estimate camera poses $\mathbf{T}_i \in \mathbf{SE(3)}$ in the target frame and 3D points ${}^{(0)}\{\mathbf{P}_j\}_{j=0}^{n}$:

$$\mathcal{PS}({}^{(0)}\{\mathbf{p}_j\}_{j=0}^{n}) \rightarrow \{{}^{(1)}\{\mathbf{T}_i\}_{i=0}^{k}, {}^{(1)}\{\mathbf{P}_j\}_{j=0}^{n}\}. \quad (3)$$

With the estimated camera poses and 3D points, new key points annotations in the training set $\mathcal{T}^1$ can be generated by projections,

$${}^{(1)}\{\mathbf{H}_i^n\}_{i=0}^{m} = \mathbf{\Pi}({}^{(1)}\{\mathbf{T}_i\}_{i=0}^{k}, {}^{(1)}\{\mathbf{P}_j\}_{j=0}^{n}, \mathbf{K}), \quad (4)$$

where $\mathbf{\Pi}$ represents the 3D-2D projection and $\mathbf{K}$ denotes the camera intrinsics. Then we re-train the detection model on the enlarged set $\mathcal{T}^{(1)} \cup \mathcal{T}^{(0)}$ and yield an improved model $f^{(1)}$. During re-training, the loss function is

$$\mathcal{L}(\theta) = \mathcal{L}_s(\theta) + \lambda \mathcal{L}_u(\theta), \quad (5)$$

where $\mathcal{L}_s(\theta)$ denotes supervised losses on the initial labeled set and $\mathcal{L}_u(\theta)$ denotes the unsupervised losses on the labels generated by Eq. (5). λ is a weight factor and is determined by the uncertainty of the projection-generated pseudo labels. When the re-training is finished, the detection model $f^{(1)}$ is further improved as the total training set is enlarged and covers more viewpoints. We use the refined detection model $f^{(1)}$ on $\mathcal{T}^{(all)}$ and yield a new set of 2D key points ${}^{(1)}\{\mathbf{p}_j\}_{j=0}^{n}$. As $f^{(1)}$ is improved, more accurate camera poses are expected to be estimated from the improved 2D observations ${}^{(1)}\{\mathbf{p}_j\}_{j=0}^{n}$. Note that not all 2D observations should be used for pose estimation. We select valid key points according to their covariance, which is addressed in detail in the first part of Section D.

The arrow cycle in Fig. 2 illustrates the iteration process. By repeating the network-and-pose dual optimization cycle, the detection model $f^{(r)}$, the camera poses ${}^{(r)}\{\mathbf{T}_i\}_{i=0}^{k}$ and 3D structures ${}^{(r)}\{\mathbf{P}_j\}_{j=0}^{n}$ at iteration round r can be stepwise improved, even though we only have a handful of ground-truth labels and no prior information about the camera poses at the beginning. To ensure the detection model $f^{(r)}$ is improved after every step, newly added labels from projection should be carefully scored according to their uncertainty. This is controlled by the factor λ in Eq.(5). In the second part of Section D, we will discuss how to determine this factor in detail. Algorithm I describes the whole optimization process step by step. For the sake of simplicity, we omit the set symbol for heat maps, camera poses, and 3D points in the pseudo code.

Algorithm 1 Network-and-pose semi-supervised training

Input:
Labeled initial training data $\mathcal{T}^{(0)}$;
Output: Refined detection model $f^{(q)}$, camera poses ${}^{(q)}\mathbf{T}_i$ and 3D points ${}^{(q)}\mathbf{P}_j$

for iteration r in 0 to q **do**
  if $r == 0$ **then**
    $f^{(0)} \leftarrow train(\mathcal{T}^{(0)})$;
  else
    Run prediction with $f^{(r)}$ to get 2D key points;
    Run pose estimation to get ${}^{(r)}\mathbf{T}_i$ and ${}^{(r)}\mathbf{P}_i$ (Eq.(3));
    Generate labels $\mathcal{T}^{(r)}$ by projection (Eq.(4));
    Evaluate valid labels with uncertainty metric;
    Retrain detection model with the enlarged set;
    $f^{(r+1)} \leftarrow train(\mathcal{T}^{(r)} \cup \mathcal{T}^{(0)})$;
  end if
end for

B. Semantic Key Point Detection Module

Many different network architectures have been proposed for semantic key point detection [11] [29]. Our semi-supervised method does not rely on a particular kind of network architecture. We choose U-net [30] as the basic architecture and use a top-down detection scheme. U-net is a light-weighted network for pixel-wise classification. Heat map regression is performed for the training and prediction of the U-net. Kullback–Leibler divergence is used as training loss. In practice, KL divergence tends to produce singular loss values due to zero entries in label heat maps. We address this issue by adding a logarithm softmax operation at the end of the network.

C. Camera Pose and 3D Points Estimation Module

Since the key point detection model already provides 2D observations which are well-associated between frames, we take them as landmarks for camera pose estimation. We set the camera pose in the *target frame*. As landmarks are fixed on the target, pose estimation is not influenced when the target is moving in the world frame. This is particularly advantageous in the case of dynamic target data collection. To realize an efficient, robust, and initialization-free estimation module, we leverage the recent advance in neural-fashioned structure from motion, i.e. the DPESFM [31] for initial pose estimation. Given key point locations, the DPESFM produces a good initial guess for poses and points to be estimated. With a least-square bundle adjustment for refinement, accurate camera poses and 3D points can be recovered efficiently and precisely. Note that using DPESFM for initialization is *not a must*, conventional 3D initialization methods like least-square triangulation are also viable alternatives.

D. Uncertainty Propagation for the Unsupervised Data

In this section, we aim to address the following question: How to properly evaluate heat map predictions and pseudo labels to select valid ones for the next turn of optimization? We first derive the uncertainty representation of heat map prediction, then we propagate it to the uncertainty of bundle adjustment to get the uncertainty of pseudo-labels.

1) Uncertainty of the Heat Map Prediction: This part aims to extract the predicted key point location μ and observation covariance $\mathbf{C}$ for uncertainty evaluation. $\mathbf{C}$ is used for the selection of valid key point predictions and uncertainty derivation of pose estimation. Provided that the detection model is sufficiently trained, heat map predictions from the network should be consistent with its supervision labels, i.e. in a 2D Gaussian distribution as well. The predicted heat map can be represented as

$$\mathcal{P}(\mathbf{x};\mu,\mathbf{C}) = \frac{1}{2\pi|\mathbf{C}|^{1/2}}\exp(-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\mu)), \tag{6}$$

where x is the image coordinate, μ is the mean of the Gaussian and $\mathbf{C}$ is the covariance of the 2D Gaussian, i.e. $\mathbf{C} = \mathrm{diag}(\sigma_{\mathrm{u}},\sigma_{\mathrm{v}})$. A straightforward way of obtaining μ is to set μ at the maximum activation location $\mu = \mathbf{x}_{\max}$ and $\mathbf{C}$ as the negative inversion of the Hessian at $\mathbf{x}_{\max}$. In practice, due to noises in the predicted heat map, $\mathbf{x}_{\max}$ is unlikely to be the true mean location. To further refine the key point location, we add the following operations. We first apply the logarithmic transformation to the predicted heat map $\mathcal{M}(x) = \ln(\mathcal{P}(\mathrm{x}))$. The gradient and Hessian, i.e. the first and second order derivative of the logarithm-transformed heat map are

$$\mathcal{M}'(\mathbf{x}) = \frac{\partial\mathcal{M}(\mathbf{x})}{\partial\mathbf{x}} = -\mathbf{C}^{-1}(\mathbf{x}-\mu), \tag{7a}$$

$$\mathcal{H} = \mathcal{M}''(\mathbf{x}) = \frac{\partial^2\mathcal{M}(\mathbf{x})}{\partial\mathbf{x}^2} = -\mathbf{C}^{-1}, \tag{7b}$$

where $\mathcal{H}(\mathbf{x})$ represents the Hessian matrix. By computing $\mathcal{H}(\mathbf{x}_{\max})$ on $\mathcal{M}$, covariance of the heat map prediction is recovered,

$$\mathbf{C}^{-1} = -\mathcal{H}(\mathbf{x})|_{\mathbf{x}_{\max}}, \tag{8}$$

where u, v denotes the image axis. As image gradient $\mathcal{M}'(\mathbf{x}_{\max})$ at $\mathbf{x}_{\max}$ can be easily computed, by combining Eq. (7a), Eq. (7b), and Eq. (8), the refined mean location μ can be recomputed as

$$\mu = \mathbf{x}_{\max} + \mathbf{C}\mathcal{M}'(\mathbf{x}_{\max}). \tag{9}$$

2) Uncertainty of Reprojection Pseudo Labels: This part aims to derive the uncertainty of reprojection and set it as the weight factor λ in Eq.(5) for each individual pseudo label. The classic bundle adjustment problem can be formulated as the maximum likelihood estimation of poses $\mathbf{T} = \{\mathbf{T}_0, \mathbf{T}_1 ... \mathbf{T}_k\}$ and 3D points $\mathbf{P} = \{\mathbf{P}_0, \mathbf{P}_1 ... \mathbf{P}_n\}$. We use $\mathbf{x}$ to represent estimation parameters $\mathbf{x} = \{\mathbf{T}_0, \mathbf{T}_1 ... \mathbf{T}_k, \mathbf{P}_0, \mathbf{P}_1 ... \mathbf{P}_n\}$. The estimation problem can be written as

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\mathrm{argmin}}\,\frac{1}{2}\sum_{i=0}^{w}\mathbf{r}_{uv}^{\mathrm{T}}\mathbf{C}_{uv}^{-1}\mathbf{r}_{uv}, \tag{10}$$

where the cost function is the sum of squared reprojection error $\mathbf{r}_{uv}$. $\mathbf{C}_{uv}$ has been derived in Eq.(8) and w denotes the number of total observations. Uncertainty of camera poses and 3D points are measured through the covariance of parameter $\mathbf{x}$:

$$\mathrm{Cov}[\mathbf{x}^\star] = (\mathbf{J}^{\mathrm{T}}\mathbf{C}_{\mathrm{all}}^{-1}\mathbf{J})^{-1}, \tag{11}$$

where $\mathbf{J} = \partial\mathbf{r}/\partial\mathbf{x}^{\mathrm{T}}|_{\mathbf{x}^\star}$ is the Jacobian matrix of the total reprojection residual $\mathbf{r}$ relative to $\mathbf{x}$. $\mathbf{C}_{\mathrm{all}} \in \mathbb{R}^{2w\times 2w}$ is a block diagonal matrix with w observation covariance on the diagonal. In practice, the Hessian $\mathbf{H} \approx \mathbf{J}^{\mathrm{T}}\mathbf{C}_{\mathrm{all}}^{-1}\mathbf{J}$ is calculated for each iteration. To avoid inversion of the Hessian, we take the Fisher information matrix (FIM) $\mathbf{I}_{\mathbf{x}} = \mathbf{J}_x^{\mathrm{T}}\mathbf{C}_{\mathrm{all}}^{-1}\mathbf{J}_x$ to measure the uncertainty. We choose the trace of the FIM and average residual as the metric for uncertainty,

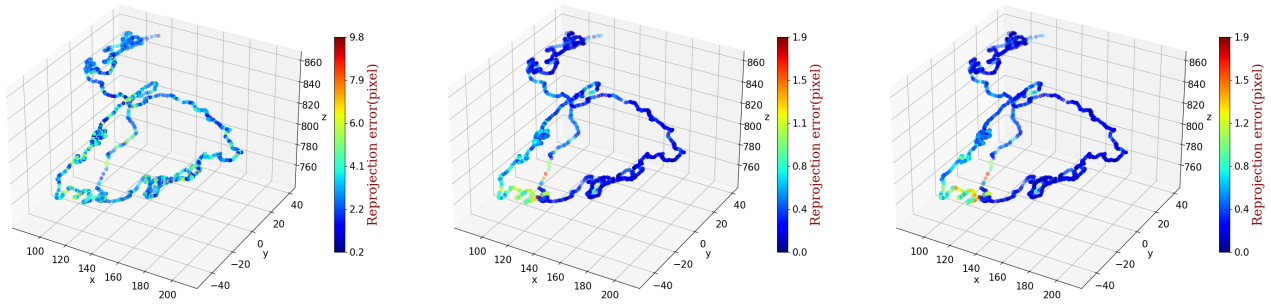
$$\lambda_i = \alpha\ \mathrm{Trace}(\mathbf{I}_{\mathbf{x}^\star}^i), \tag{12}$$

where $\mathbf{I}_{\mathbf{x}^\star}^i$ denotes the ith camera pose block in the FIM. α is a positive scaling factor. We treat α as a hyperparameter. The trace of FIM reflects how informative the current detection results are. Eq.(12) indicates that we assign lower weights to frames which is less observable, and higher weights to frames of which the observations are more informative.

IV. Experimental Evaluations

A. Experiment on the YCB Video Dataset

In this section, we conduct experiments on the YCB video dataset [32] to verify the effectiveness of our method. We select several objects and skip ones lacking in remarkable corner key points. For each selected object, we choose 5 scenes for training and validation, and another 2 scenes for testing. The detection model takes an input of 320×240 pixels. We train the network on an Nvidia GeForce RTX



(a) *Iteration round 0* (b) *Iteration round 1* (c) *Iteration round 2*

Fig. 3: Camera trajectory of scene #50 in the YCB video dataset. The color of each dot reflects the average reprojection error of that frame. Note that at round 0, camera poses are initialized with the DPESFM, and the average error is around 5 pixels. After several iterations with our method, reprojection errors of most images narrow down.

Fig. 4: Key points detection results and heat map predictions from the detection network on the YCB video dataset.

4090 GPU with the RMSprop optimizer and a learning rate of 1e-05. Fig. 3 shows the 3D camera trajectory of scene #50 in the YCB video dataset. The target object is the mustard bottle which contains 7 key points. We can see that at iteration round 0, the average reprojection error is around 5 pixels, which is not accurate enough to produce pseudo labels. This is because poses are initialized by the DPESFM without bundle adjustment refinement. After two rounds of iteration, the reprojection error at most frames narrowed down to less than 1 pixel.

Determining the ratio of initially manually labeled data is a critical challenge. We must strike a balance between the performance of the initial detection model and the workload for human labeling. Factors like illumination conditions and viewpoint distribution may influence model performance, making it intractable to determine an optimal initial ratio analytically. To find a reasonable initial ratio, we test the relation between pixel error and labeled training data ratio, as shown in Fig. 5. By projecting the intersection point of the reprojection threshold and reprojection-ratio curve onto the X-axis, as shown by the red dotted line in Fig. 5, we get a reasonable label ratio. In practice, we choose 4 pixels as the threshold, since the radius of the heat map kernel is 2 pixels. From Fig. 5 we also find that training with an extremely small fraction of labels (1% to 2%) fails to produce a qualified detection model for pose estimation, as the model is poorly trained and most detected landmarks are invalid. When starting training with more than 70% of total data, the improvement from pose estimation and pseudo labels becomes marginal, as the performance of the initial model closely matches that of the model trained with pseudo labels.

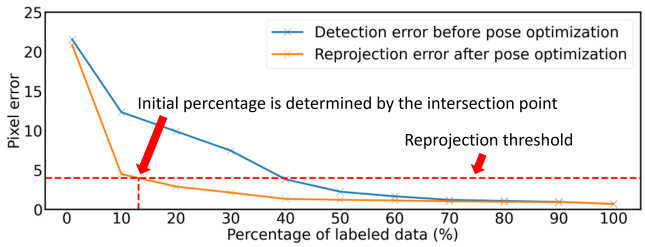


Fig. 5: Relation between average pixel error and labeled training data ratio. The red dotted line represents the reprojection threshold and its intersection with the curve is treated as the empirical initial ratio.

TABLE II: Average detection errors in pixels of different training conditions. The percentage in the first row denotes the training data ratio. FS refers to the fully-supervised training. SS refers to our semi-supervised training strategy.

Object	1%-FS	10%-FS	SS	100%-FS
003_cracker_box	19.98	14.25	1.27	0.87
004_sugar_box	18.78	12.61	0.90	0.80
006_mustard_bottle	17.59	11.95	0.79	0.75
008_pudding_box	18.96	9.62	0.98	0.54
009_gelatin_box	20.68	8.54	0.92	0.72
010_potted_meat_can	24.11	18.32	0.58	0.43
019_pitcher_base	27.54	12.17	0.91	0.61
025_mug	19.21	11.15	0.79	0.45
035_power_drill	18.60	12.28	1.11	0.98
Average	20.61	12.32	0.92	0.68

As depicted in Table II, the detection error of our semi-supervised method closely matches that of the 100% fully-supervised training method by 0.24 pixels, despite beginning training with only 10%-20% labeled data. Fully-supervised training with 1% or 10% of total data produces sub-par results. The key reason for the performance improvement is that the semi-supervised method greatly increases the number of valid training data. This proves the effectiveness and accuracy of our semi-supervised method.

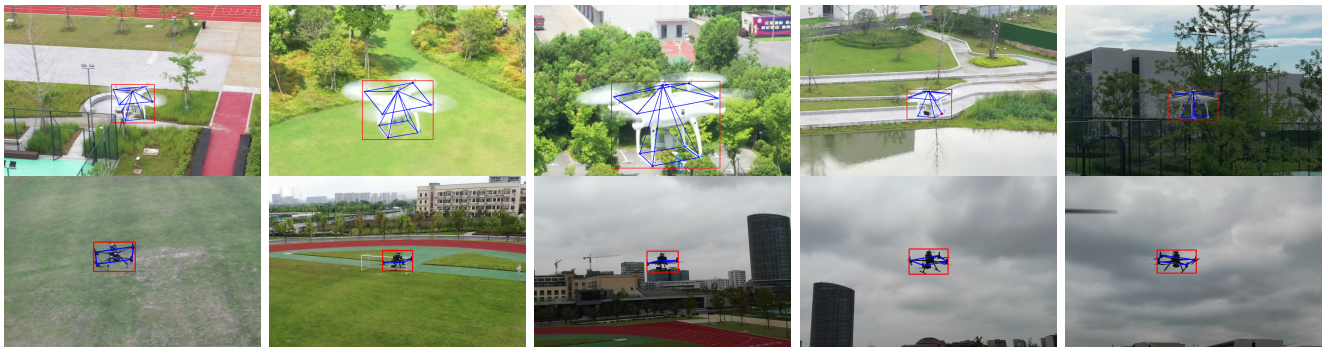


Fig. 6: Detection results of the outdoor aerial pursuit experiments.

Finally, we compare the performance of our method with that of the state-of-the-art semi-supervised method for semantic key point detection. As discussed in Table I, among all the state-of-the-art methods, S3K [12] has equivalent experimental settings to our method, so we choose it as a baseline method. Other methods require multiple synchronized cameras, which is different from our setting. Compared with S3K, our method can recover camera poses for triangulation and train the detection model in a semi-supervised manner. We use the term "ours" to represent our basic semi-supervised method. In addition, as discussed in Sec.III-D, our method adds three more techniques regarding uncertainty modeling to improve the detection performance: 1) Modeling the covariance of the predicted heat map (Eq. 8), denoted by "UH". 2) Refinement for key point location (Eq. 9), denoted by "Ref". 3) Modeling of pseudo label weights (Eq. 12), denoted by "UL".

Ground truth camera poses from the YCB-video dataset are used for S3K. Table III shows the comparison and ablation study results. Detection errors are averaged from different objects. Comparative and ablation results show that given ground truth camera poses, S3K and our basic method produce key point detection accuracy of the same level. The results of S3K are 0.2 pixels more accurate than ours since it uses ground truth camera poses for triangulation. While S3K can not recover camera poses, ours recovers poses with an accuracy level of 1.6 deg. and 29 mm. The uncertainty modeling and point location refinement improve the overall performance of detection accuracy. In conclusion, our method produces a nearly on-par performance with S3K while recovering 6DoF camera poses.

TABLE III: Comparison with the state-of-the-art work and ablation study in terms of detection and pose estimation errors. Our method achieves a nearly on-par performance while recovering 6DoF camera poses.

Method	Detect. (Pixel)	Rot. (Deg)	Trans.(mm)
S3K [12]	1.29	N/A	N/A
S3K [12]+UH.	1.12	N/A	N/A
S3K [12]+UH.+Ref.	0.77	N/A	N/A
S3K [12]+UH.+Ref.+UL.	0.71	N/A	N/A
Ours (basic semi-supervision)	1.54	3.98	31.2
Ours+UH.	1.43	3.63	30.4
Ours+UH.+Ref.	0.99	1.59	29.1
Ours+UH.+Ref.+UL.	0.92	1.51	29.1

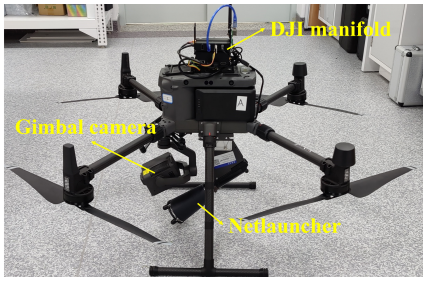


Fig. 7: Experimental platform for aerial pursuit.

B. Real World Experiment 1: MAVs Pursuit

In this part, we conduct experiments on outdoor MAV detection and pursuit. During the pursuit, the pursuer relies on visual key points to estimate the target's pose, facilitating tracking and capture. Fig. 7 shows the experimental setup. We use a DJI M300 as the carrier and the DJI H20 gimbal camera to capture images. The platform is also equipped with a DJI manifold with the Nvidia Jetson TX2 as the onboard computation unit. The pursuer is also equipped with a net launcher for capturing the target drone.

Images are extracted from videos and cropped at the center. The cropped images are resized to a size of 320×180 pixels. We convert the model to TensorRT with FP16 precision for inference acceleration and achieve 14 FPS onboard rate. Distance between the pursuer and target drone varies from 15 m to 2.5 m. We keep the target drone still in the world frame during the collection, and the pursuer approaches the target from far to near. Fig. 6 shows the detection results. We increase the inter-frame distance for each scene and perform collection in scenes with various backgrounds to avoid overfitting. As the inter-frame distance is enlarged, optical flow label propagation [28] [16] tends to fail. An extremely small fraction of labels, as 1% to 2% in [16], is far from being sufficient. We manually label all images as ground truth key point locations. Using the technique introduced by Fig. 5, we take 11% as the initial labeling ratio. Ground truth pose is recovered by using PnP as in [7] with ground truth key points.

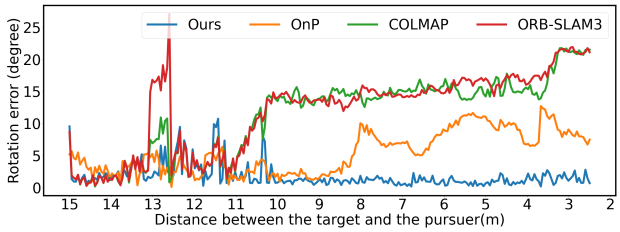


Fig. 8: Comparison with state-of-the-art camera pose estimation methods during the aerial pursuit in terms of rotation error. With the pursuer approaching the target, errors of other methods increase while ours stay stable.

Since our method can train the detection model together with 6DoF camera pose estimation, we compare our method with the state-of-the-art camera pose estimation methods, i.e. ORB-SLAM3 [33], COLMAP [34] and Orthogonal-n-Point [35] [36] in terms of reprojection and pose estimation accuracy. Fig. 8 shows the camera-target rotation error curve. The pursuer approaches the target from far to near hence the x-axis is inversed. At the beginning stage, of which the distance is larger than 10 m, the pose estimation accuracy of all methods is at the same level. With the pursuer further approaching the target, rotation errors of COLMAP, ORB-SLAM3, and OnP become larger. The core reason for the inferior performance of COLMAP or ORB-SLAM3 is as follows. Although we keep the target drone motionless throughout the data collection process, hovering drift is unavoidable in practice. COLMAP and ORB-SLAM3 take static feature points for pose estimation, producing less accurate poses in the drifting target frame in the long run. OnP is an approximation method often used when precise camera parameters are unknown, which produces subpar estimation results, especially at close range as the depth variation of 3D points is neglected.

Table IV summarizes the median of errors and computation time among different scenes of different methods. Our method outperforms others in terms of reprojection and pose estimation accuracy. Besides, thanks to the limited number of visual landmarks fixed on the target and the simplified estimation scheme, our method takes much less time. In conclusion, our method produces reprojection-generated pseudo labels with higher fidelity and efficiency, which is particularly favorable for dynamic robotic applications such as aerial pursuit.

TABLE IV: Reprojection and pose estimation comparison.

Method	Repro. (pixel)	Rot. (degree)	Trans. (mm)	Time (s)
DPESFM [31]	1.64	25.80	280	**0.6**
Ours (DPESFM+BA)	**1.43**	**2.82**	**43**	4.0
OnP [36]	1.88	8.54	N/A	207.1
COLMAP [34]	7.03	11.04	251	18,741.8
ORB-SLAM3 [33]	9.92	12.12	305	551.9

Limitations: In the data collection process, the method estimates camera pose in the target frame. Therefore, only one target object with semantic key points should be visible at a time. Detecting key points from multiple moving targets in a single frame can lead to incorrect pose estimation results. However, during the online detection process, multiple targets can be detected simultaneously.

C. Real World Experiment 2: Mobile Robot Formation

In addition to aerial pursuit, we also apply the proposed method to a mobile robot system. Each robot in the fleet is equipped with an omnidirectional camera set and an Nvidia Jetson NX. Semantic key points are used by each robot to estimate its relative pose to neighboring robots, enabling leader-follower formation motion based on the relative pose. We collected 3,762 images of the robot in an indoor environment and applied our semi-supervised strategy to train the detection model. Given the effectiveness of the U-net in the constrained indoor environment, only 127 manually labeled images were needed, significantly reducing the labeling workload. Fig. 9 illustrates the vision-based mobile robot system and the online key point detection results during the collective motion experiment, demonstrating the versatility of our method in robotic applications.

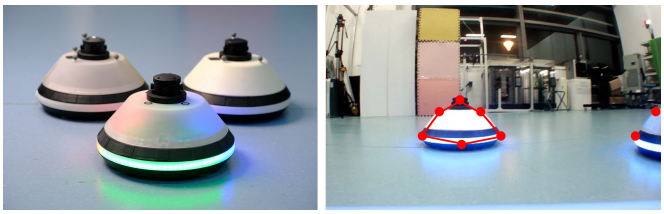


Fig. 9: The multi-robot system and online image detection results. Semantic key points are detected and used for relative pose estimation in the formation experiment.

V. Conclusions

This paper introduces a novel semi-supervised method for training a semantic key point detection model. Our approach requires only a small fraction of manually labeled data while estimating full 6DoF camera poses. It proves particularly beneficial for multi-robot systems where pre-labeled datasets of custom robot types are hard to obtain. We conduct experiments on both real-world robotic applications and public datasets of everyday objects, demonstrating the effectiveness and accuracy of our method.

Acknowledgment The authors would like to thank Canlun Zheng, Huaben Chen, Shiliang Guo, Jiachen Liang, and Zian Ning for the data collection and experiments.

References

[1] S. Bonato, S. C. Lambertenghi, E. Cereda, A. Giusti, and D. Palossi, "Ultra-low power deep learning-based monocular relative localization onboard nano-quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3411–3417, IEEE, 2023.

[2] Y. Zheng, C. Zheng, X. Zhang, F. Chen, Z. Chen, and S. Zhao, "Detection, localization, and tracking of multiple mavs with panoramic stereo camera networks," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 1226–1243, 2023.

[3] X. Oh, R. Lim, L. Loh, C. H. Tan, S. Foong, and U.-X. Tan, "Monocular uav localisation with deep learning and uncertainty propagation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7998–8005, 2022.

[4] F. Schilling, F. Schiano, and D. Floreano, "Vision-based drone flocking in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2954–2961, 2021.

[5] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[6] R. Ge, M. Lee, V. Radhakrishnan, Y. Zhou, G. Li, and G. Loianno, "Vision-based relative detection and tracking for teams of micro aerial vehicles," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 380–387, IEEE, 2022.

[7] M. Pavliv, F. Schiano, C. Reardon, D. Floreano, and G. Loianno, "Tracking and relative localization of drone swarms with a vision-based headset," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1455–1462, 2021.

[8] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.

[9] S. Bultmann, R. Memmesheimer, and S. Behnke, "External camera-based mobile robot pose estimation for collaborative perception with smart edge sensors," in *International Conference on Robotics and Automation (ICRA)*, pp. 1251–1257, IEEE, 2023.

[10] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2011–2018, IEEE, 2017.

[11] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[12] M. Vecerik, J.-B. Regli, O. Sushkov, D. Barker, R. Pevceviciute, T. Rothörl, C. Schuster, R. Hadsell, L. Agapito, and J. Scholz, "S3K: Self-supervised semantic keypoints for robotic manipulation via multi-view consistency," in *Conference on Robot Learning (CoRL)*, 2020.

[13] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: a toolkit for robust markerless 3d pose estimation," *Cell reports*, vol. 36, no. 13, p. 109730, 2021.

[14] B. Usman, A. Tagliasacchi, K. Saenko, and A. Sud, "Metapose: Fast 3D pose from multiple views without 3D supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6759–6770.

[15] J. J. Sun, L. Karashchuk, A. Dravid, S. Ryou, S. Fereidooni, J. C. Tuthill, A. Katsaggelos, B. W. Brunton, G. Gkioxari, A. Kennedy, *et al.*, "Bkind-3d: Self-supervised 3d keypoint discovery from multi-view videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9001–9010, 2023.

[16] M. Dabhi, C. Wang, T. Clifford, L. Jeni, I. Fasel, and S. Lucey, "MBW: Multi-view bootstrapping in the wild," *Advances in Neural Information Processing Systems (NIPS)*, vol. 35, pp. 3039–3051, 2022.

[17] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning monocular 3D human pose estimation from multi-view images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8437–8446, 2018.

[18] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1145–1153, 2017.

[19] Y. Zhang and H. S. Park, "Multiview supervision by registration," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 420–428, 2020.

[20] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7718–7727, 2019.

[21] K. Takahashi, D. Mikami, M. Isogawa, and H. Kimata, "Human pose as calibration pattern: 3D human pose estimation with multiple unsynchronized and uncalibrated cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1775–1782, 2018.

[22] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432, IEEE, 2020.

[23] F. Shkurti, W.-D. Chang, P. Henderson, M. J. Islam, J. C. G. Higuera, J. Li, T. Manderson, A. Xu, G. Dudek, and J. Sattar, "Underwater multi-robot convoying using visual tracking by detection," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4189–4196, IEEE, 2017.

[24] S. Li, C. De Wagter, and G. C. De Croon, "Self-supervised monocular multi-robot relative localization with efficient deep neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9689–9695, IEEE, 2022.

[25] K. M. Judd and J. D. Gammell, "The Oxford multimotion dataset: Multiple SE(3) motions with ground truth," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 800–807, 2019.

[26] H. Naik, A. H. H. Chan, J. Yang, M. Delacoux, I. D. Couzin, F. Kano, and M. Nagy, "3D-POP:An automated annotation approach to facilitate markerless 2D-3D tracking of freely moving birds with marker-based motion capture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21274–21284, 2023.

[27] M. Dabhi, C. Wang, K. Saluja, L. A. Jeni, I. Fasel, and S. Lucey, "High fidelity 3D reconstructions with limited physical views," in *2021 International Conference on 3D Vision (3DV)*, pp. 1301–1311, IEEE, 2021.

[28] Y. Yao, Y. Jafarian, and H. S. Park, "Monet: Multiview semi-supervised keypoint detection via epipolar divergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 753–762, 2019.

[29] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–499, Springer, 2016.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, Springer, 2015.

[31] D. Moran, H. Koslowsky, Y. Kasten, H. Maron, M. Galun, and R. Basri, "Deep permutation equivariant structure from motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5976–5986, 2021.

[32] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[33] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[34] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4104–4113, 2016.

[35] C. Steger, "Algorithms for the orthographic-n-point problem," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 2, pp. 246–266, 2018.

[36] I. Kissos, L. Fritz, M. Goldman, O. Meir, E. Oks, and M. Kliger, "Beyond weak perspective for monocular 3D human pose estimation," in *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pp. 541–554, Springer, 2020.