# MVP: One-Shot Object Pose Estimation by Matching with Visible Points

Wentao Cheng[1], Minxing Luo[1]

***Abstract*— We introduce a novel method for one-shot object pose estimation. Recent detector-free one-shot methods have achieved promising results for challenging low-textured objects. The features in a query image are directly matched with all features in an object point cloud reconstructed via Structure-from-Motion (SfM) techniques. Rejecting invisible 3D points, as well as associated features, is performed implicitly using a deep neural network that is trained specifically for feature matching. This tightly-coupled strategy is prone to preserve 3D points that are rarely visible from the query view. In contrast, we propose to prune such erroneous points using the explicit image-point relational graph, which is a lightweight by-product of the SfM reconstruction. By injecting the graph-based pruning into stacked feature transformers, our method is able to obtain high quality 2D-3D correspondences through matching with visible points in an early stage. The experiments demonstrate that our method outperforms state-of-the-art one-shot methods with faster speed. The source code is available at https://github.com/wtchengcv/MVP.**

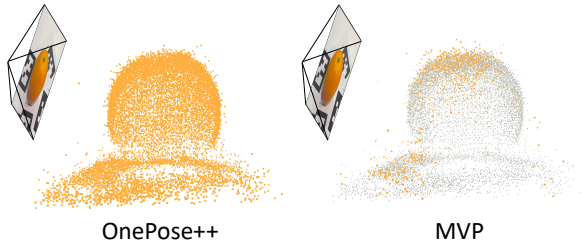


Fig. 1: Comparison between our method MVP and the state-of-the-art one-shot object pose estimation method OnePose++ [6]. OnePose++ uses the complete object point cloud (in orange) to establish 2D-3D matches with a query image. While our method prunes invisible 3D points (in grey) in an early stage. Best viewed in color.

## I. Introduction

Estimating the six degree-of-freedom (6-DOF) object poses from a single RGB image is a fundamental problem in augmented reality. It is also crucial for embodied agents (i.e. robots), as various interactive tasks require precise knowledge of the position and orientation of objects in 3D space. While substantial attention has been paid to instance-level object pose estimation [1, 2], the requirement of high-quality CAD models for each instance greatly limits their scope of use in practical scenarios. Category-level methods [3, 4] reduce the need for CAD models by learning a general representation over different instances in the same category. Yet, it remains challenging when facing objects of unseen category.

In order to achieve better generalization capability, recent one-shot object pose estimation methods [5, 6] revisit the traditional visual localization pipeline with generalizable feature matching neural networks. OnePose [5] works in a setting that a sequence of images with annotated 3D bounding boxes for each object are given in advance, and the object point cloud with local features is reconstructed using Structure-from-Motion (SfM) techniques [7]. The features extracted from a query image directly match with the object point cloud to enable efficient object pose estimation. To go a step further, OnePose++ [6] proposes a detector-free pipeline from point cloud generation to pose estimation, which shows superiority in handling low-textured objects. Leaving aside the type of local features, OnePose and OnePose++ both adopt the attention mechanism with all positional encoded features in an object point cloud, so that globally-consented 2D-3D matches can established.

However, a key common sense is that only part of an object can be observed from a viewpoint in most cases. The 3D points and associated features lying on the back of an object are invisible, deemed as unnecessary noise. Their survival may affect the effectiveness of feature matching, by either introducing wrong attention or harassing the mutual nearest neighbor search. In OnePose and OnePose++, such invisible points are implicitly handled by assuming that the neural network learns to reject them during feature matching. We argue that a fundamental misalignment easily break this assumption: the feature matching neural network is trained to be 3D position- and feature-appearance-dependent, while determining the visibility of a 3D point is a 6D view-dependent and feature-appearance-independent task.

A natural question arises: *can we efficiently and effectively prune invisible points for one-shot object pose estimation?* To this end, we introduce a novel 6D object pose estimation paradigm that ultimately Match with Visible Points (MVP). The key ingredient of MVP is the disentanglement of feature matching and invisible point pruning. In our proposed paradigm, a transformer-based neural network is utilized purely for feature matching, while invisible points are pruned with the guidance of explicit visibility priors. We integrate these two modules into a *look-around and gaze* pipeline, to mimic how human find 2D-3D matches for objects. Specifically, we first feed all features in an object point cloud to the feature matching network to capture the coarse global

[1]Wentao Cheng (corresponding author) and Minxing Luo are with the VCIP Lab from Nankai University, wentaocheng@nankai.edu.cn

context. We encourage the early attention layers to possess the capability of match prediction, so that features that are more distinguishable can be selected quickly. This *look-around* step allows us to narrow down the search space and only *gaze* from one viewpoint. Invisible points and features are safely pruned by exploiting the bipartite visibility graph, which is a "free" and lightweight prior from SfM. Fig. 1 gives a demonstration about the key difference between our method and OnePose++.

We evaluate the proposed method on several object pose estimation datasets. The experimental results show that MVP achieves the state-of-the-art performance in a one-shot and CAD-model-free manner. In addition, as much fewer 3D points are engaged to obtain higher quality 2D-3D matches, our method achieves near real-time efficiency with only 75 *ms* to process one query image on GPU.

## II. Related Work

**Generalizable Object Pose Estimation.** Instance-level object pose estimation methods [1, 8–11] need to train a neural network individually for each instance, which usually require high-fidelity CAD models to render redundant training samples. Category-level methods [3, 12–15] learn a shape prior for each category, thus alleviating the requirement for CAD models. However, category-level methods still struggle in generalizing to unseen categories.

We categorize generalizable object pose estimation approaches into two types: CAD-model-based and CAD-model-free. In order to generate sufficient rendered views for template matching, recent CAD-model-based approaches [16–20] rely on CAD models of objects during the training phase. While CAD-model-free methods [5, 6, 21, 22] leverage other easily available proxies, such as unlabelled or labelled RGB images, to facilitate zero-shot or one-shot object pose estimation. Gen6D [21] first detect the object and compute image-wise similarity for view selection, thus is prone to fail in occluded scenarios. OnePose [5] and OnePose++ [6] train a generalizable feature matching model to establish 2D-3D matches with respect to a pre-computed object point cloud.

**Visual Localization.** Visual localization aims to estimate the camera poses of query images with respect to a reconstructed scene model. Structure-from-Motion techniques [7] are usually used to build the scene model, which contains either traditional hand-crafted [23] or learned features [24, 25]. The 2D-3D correspondences between the 2D query image and 3D scene model thus can be established by either nearest neighbor search [26] or bag-of-word methods [27].

To handle the scale curse brought by large scene models, graph-based methods [28–33] rely on the bipartite visibility graph in the scene model, to prioritize the candidates that are more likely to find correct matches. Other approaches [34, 35] assume that the gravity prior is known to narrow the searching space. HLoc [36] proposes a coarse-to-fine localization paradigm. The final 2D-3D correspondences are obtained by first using image retrieval to establish 2D-2D correspondences with several database images. Moreover, the above-mentioned methods adopt a fixed criteria to evaluate the local feature appearance individually, making the match disambiguation step unstable.

Recent learning-based matching methods [37, 38] have achieved better performance by considering the global spatial and visual context. The message passing is accomplished by staggered self and cross attention layers with positional encoding. To reduce the computational cost, which is $O(N^2)$ with respect to the length of input tokens, LoFTR [38] replaces traditional full attention with linear kernelized attention [39]. LightGLUE [40] discards features that are likely to be unmatchable at early attention layers, such as points from indistinguishable background areas. IMP [41] iteratively sample potential correct matches guided by a predicted epipolar line. Yet, those pruning strategies are specially designed for 2D-2D matching. Instead, our proposed approach takes the 3D scene structure into consideration, and effciently prunes invisible 3D points and features.

## III. Method

An overview of the proposed method is given in Fig. 2. We propose a graph-based pruning module that communicates with the coarse matching block, to discard invisible 3D points and points dynamically. The sub-pixel 2D-3D matches are obtained with subsequent fine matching block to facilitate the final object pose estimation.

### A. Preliminaries

**Detector-free SfM.** Given a set of captured images $\{\mathbf{I}_i\}$ that describe the same object, we use the detector-free SfM [6] to reconstruct the semi-dense object point cloud. Simply put, coarse feature maps $\{\tilde{\mathbf{F}}_i\}$ are extracted from the images and 2D-2D feature matching is performed using LoFTR [38]. A coarse object point cloud is then reconstructed using the popular SfM toolkit COLMAP [7]. The final accurate point cloud $\{\mathbf{P}_j\}$ is built by refining the sub-pixel locations with a separately extracted fine-level feature maps $\{\hat{\mathbf{F}}_i\}$. Each 3D point $\mathbf{P}_j$ is triangulated from a set of 2D keypoints with associated features $\{\tilde{\mathbf{F}}_i^k\}$ and $\{\hat{\mathbf{F}}_i^k\}$, where $k$ is the feature index in image $i$. For the later object pose estimation, each 3D point stores a coarse feature $\{\tilde{\mathbf{F}}_j^{3D}\}$ and a fine feature $\{\hat{\mathbf{F}}_j^{3D}\}$ by averaging its associated features. There is also a bipartite visibility graph $\mathcal{G} = \{\mathcal{P}, \mathcal{I}, \mathcal{E}\}$ that is naturely encoded after SfM reconstruction. Here $\mathcal{P}$ represents the indices of all 3D points, and $\mathcal{I}$ represents the indices of images. An edge $\mathcal{E}(i, j) = 1$ if one of the keypoints from the image $\mathbf{I}_i$ is used to triangulate the 3D point $\mathbf{P}_j$, otherwise 0.

**Object Pose Estimation.** We briefly describe the state-of-the-art one-shot object pose estimation method [6] using the reconstructed object point cloud. In the localization stage, the coarse-level and fine-level feature maps ($\{\tilde{\mathbf{F}}_q^{2D}\}$, $\{\hat{\mathbf{F}}_q^{2D}\}$) are extracted from a query image $\mathbf{I}_q$. With a stack of self and cross attention layers, $\{\tilde{\mathbf{F}}_q^{2D}\}$ interact with coarse features $\{\tilde{\mathbf{F}}_j^{3D}\}$ stored in the point cloud to reveal the global context. A set of coarse 2D-3D matches are found by measuring the similarity of transformed coarse features. Subsequently,

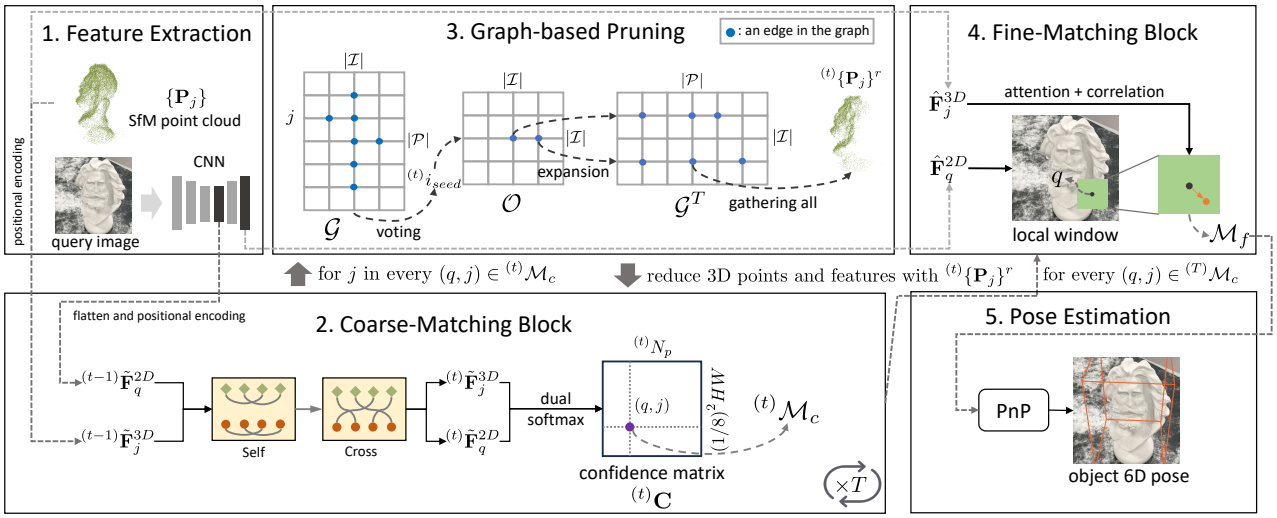


Fig. 2: Overview of MVP. There are five modules in MVP: 1, coarse and fine level features from a query image are extracted via CNN; 2, coarse features with positional encoding are passed to the coarse matching block to generate intermediate 2D-3D matches ${}^{(t)}\mathcal{M}_c$ after $t$ blocks; 3, ${}^{(t)}\mathcal{M}_c$ is used to vote a seed camera index with the visibility graph $\mathcal{G}$, expand on the image overlap graph $\mathcal{O}$, and gather all visible 3D points and features using the transpose of $\mathcal{G}$; 4, fine-level 2D-3D matches ${}^{(T)}\mathcal{M}_f$ are computed with correlation after traversing all $T$ coarse matching blocks; 5, final object pose is estimated using Perspective-n-Point algorithm and RANSAC.

the corresponding fine-level features are augmented with attention, to refine the locations of 2D keypoints in a local window. The final object pose $\boldsymbol{\xi}_q$ thus can be computed with the refined 2D-3D matches. However, a large portion of invisible points for the query view are involved in the complete localization stage, which severely decreases the feature matching accuracy. In the next section, we present how to efficiently handle this problem.

### *B. Matching with Visible Points*

**Early Prediction.** We first extract the coarse-level feature map $\{\tilde{\mathbf{F}}_q^{2D}\} \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times\tilde{c}}$, where $H$ and $W$ are the height and width of query image respectively, and $\tilde{c}$ is the dimension of coarse feature. We use the same feature transformer architecture with [6, 38], which includes several multi-head attention (MA) operations as following:

$$\begin{aligned}
\Phi_q^{2D} &= \text{MA}({}^{(t-1)}\tilde{\mathbf{F}}_q^{2D}, {}^{(t-1)}\tilde{\mathbf{F}}_q^{2D}) \\
\Phi_j^{3D} &= \text{MA}({}^{(t-1)}\tilde{\mathbf{F}}_j^{3D}, {}^{(t-1)}\tilde{\mathbf{F}}_j^{3D}) \\
{}^{(t)}\tilde{\mathbf{F}}_q^{2D} &= \text{MA}(\Phi_q^{2D}, \Phi_j^{3D}) \\
{}^{(t)}\tilde{\mathbf{F}}_j^{3D} &= \text{MA}(\Phi_j^{3D}, \Phi_q^{2D}),
\end{aligned} \tag{1}$$

where $t$ is the index of coarse matching block. The first element in MA is used to obtain the query vector $Q$ with linear projection, and the second element is used to obtain the key vector $K$ and the value vector $V$. Usually the semi-dense object point cloud may contain tens of thousands of 3D points, the traditional dot product attention with the point cloud will be slow. To this end, we follow [6, 38] to adopt the fast linear attention [39] as following:

$$\text{Attention}(Q, K, V) = \phi(Q)\left(\phi\left(K^T\right)V\right) \tag{2}$$

, where $\phi(\cdot) = \text{elu}(\cdot) + 1$. As the depth of coarse matching blocks increases, the resultant features gradually become more discriminative. The features that are discriminative themselves shuold easily be distinguished at early blocks. Our key objective is to gather sufficient correct 2D-3D matches as early as possible. To this end, after a coarse matching block $t$, we measure the feature similarity between ${}^{(t)}\tilde{\mathbf{F}}_q^{2D}$ and ${}^{(t)}\tilde{\mathbf{F}}_j^{3D}$, and apply dual-softmax to obtain the coarse match confidence matrix ${}^{(t)}\mathbf{C}$. The mutual nearest matches whose confidences are above the threshold $\delta$ are collected as ${}^{(t)}\mathcal{M}_c$. Similar to [5, 6], we adopt the focal loss ($FL$) on the confidence matrix. In order to make each block possess the capability of prediction, we apply the focal loss on the confidence matrix returned from each block. The overall loss for coarse match prediction $\mathcal{L}_c$ is summarized as following:

$$\mathcal{L}_c = \frac{1}{T}\sum_t \frac{1}{\left|{}^{(t)}\mathcal{M}_c\right|} \sum_{(q,j)\in{}^{(t)}\mathcal{M}_c} FL({}^{(t)}\mathbf{C}(q,j)), \tag{3}$$

where $T$ is number of all coarse matching blocks, $q$ and $j$ represent the index of 2D and 3D features respectively.

**Graph-based Pruning.** Assuming that a relatively strict threshold is set, the intermediate coarse 2D-3D matches ${}^{(t)}\mathcal{M}_c$ should contain a preliminary set of correct matches. A straightforward way to prune invisible 3D points is to utilize the back-face culling algorithm [42]. By computing a coarse object pose using ${}^{(t)}\mathcal{M}_c$, the 3D points and associated features are discarded if the dot product of surface normal and camera-to-point vector is greater than or equal to zero. In SfM point clouds, each 3D point $\mathbf{P}_j$ also includes a priori feature orientation, by averaging the camera-to-point vectors

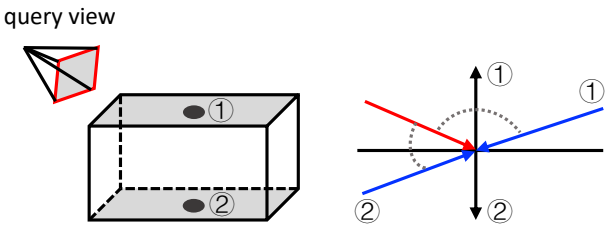


Fig. 3: An illustration of the weakness when pruning with the back-face culling algorithm or view-angle difference. The red arrow represents the query view direction; The black arrow represents the surface normal. The blue arrow represents the average feature direction of a 3D point (best viewed in color).

of all images from $\{\mathbf{I}_i|\mathcal{E}(i,j)=1\}$. Subject to limited rotational invariance of features, the feature of a 3D point is rarely observed by query views that are significantly different from its priori orientation. The back-face culling algorithm is difficult to reject reject invisible 3D points (the first type in Fig. 3) that locate on the front plane of a query view frustum. In addition, computing a coarse object pose also introduces an inevitable computational overhead. Pruning 3D points by measuring the difference between the camera-to-point vector and the priori orientation of features is also not adequate, since it is prone to preserve 3D points that lie on the back of objects (the second type in Fig. 3).

To recap, the inherent bipartite visibility graph $\mathcal{G}$ is a lightweight summarization of the visibility of features with respect to a discrete set of camera views. Here we show how to exploit this graph for clean point pruning. The first step is to identify a seed image node in $\mathcal{G}$ whose object pose is similar to the query object pose. Instead of defining thresholds in the euclidean space, we propose a more generalized method by graph-based voting. Ideally, features in a

Ideally, if two object poses are similar in both orientation and position, they should observe a similar set of features. Assuming that the features of 3D points in ${}^{(t)}\mathcal{M}_c$ is the query view can observe. We select the seed image node ${}^{(t)}i_{\text{seed}}$, that co-observe most with the query view

$$ {}^{(t)}i_{\text{seed}} = \arg\max_{i} \sum_{(q,j)\in {}^{(t)}\mathcal{M}_c, i\in\mathcal{I}} \mathcal{E}(i,j). \tag{4} $$

Preserving 3D points with only the seed image ${}^{(t)}i_{\text{seed}}$ is prone to discard useful features. In order to safely expand the 3D points, we build an undirected overlap graph $\mathcal{O}$. The overlap ratio $\mathcal{O}(i,k)$ between image $\mathbf{I}_i$ and $\mathbf{I}_k$ is measured as following:

$$ \mathcal{O}(i,k) = \frac{\sum_{j\in\mathcal{P}} \mathcal{E}(i,j)\times\mathcal{E}(k,j)}{\sum_{j\in\mathcal{P}} \mathcal{E}(i,j)}. \tag{5} $$

A small overlap ratio indicates that two images may captured from significantly different views. We first select from the neighbors of node ${}^{(t)}i_{\text{seed}}$ in the overlap graph $\mathcal{O}$ whose overlap ratios are above the threshold $\theta$. The final point pruning is performed by discarding the 3D points that are not observed by any of the selected images, and the reduced point set ${}^{(t)}\{\mathbf{P}_j\}$ is formulated as following:

$$ {}^{(t)}\{\mathbf{P}_j\} = \left\{\mathbf{P}_j|\mathcal{E}(k,j)=1 \wedge \mathcal{O}(k, {}^{(t)}i_{\text{seed}}) > \theta\right\} \tag{6} $$

**Fine Matching.**

TABLE I: **Comparison with One-shot Approaches.**

| | OnePose | | |
|---|---|---|---|
| | 1cm-1deg | 3cm-3deg | 5cm-5deg |
| HLoc (SPP + SPG) | 51.1 | 75.9 | 82.0 |
| HLoc (LoFTR) | 39.2 | 72.3 | 80.4 |
| OnePose | 49.7 | 77.5 | 84.1 |
| †OnePose++ | 51.5 | 80.5 | 87.4 |
| **MVP** | **52.1** | **80.8** | **87.6** |
| | OnePose-LowTexture | | |
| | 1cm-1deg | 3cm-3deg | 5cm-5deg |
| HLoc (SPP + SPG) | 13.8 | 36.1 | 42.2 |
| HLoc (LoFTR) | 13.2 | 41.3 | 52.3 |
| OnePose | 12.4 | 35.7 | 45.4 |
| †OnePose++ | 17.2 | 58.7 | 73.1 |
| **MVP** | **18.4** | **62.2** | **77.1** |

## IV. EXPERIMENTS

### A. Experiment Settings

**Datasets.** We evaluate our method on three object 6D pose estimation datasets. The OnePose dataset [5] consists of more than 450 video sequences of 150 rich-textured objects. In the validation set, the object point cloud is reconstructed using one video sequence with bundle adjustment to reduce the drift error. Other video sequences of an object are used for evaluation. Note that each frame is accompanied with the 3D bounding box annotation and the ground truth camera pose. The OnePose-LowTexture dataset [6] is a supplement of the original OnePose dataset, which contains 40 low-textured household objects. Additionally, eight of the 40 objects are provided with high-fidelity CAD models. The LINEMOD dataset [43] is a widely used benchmark for object 6D pose estimation. It consists of 13 low-textured objects and relatively low-resolution images, and corresponding ground truth CAD models.

**Metrics and Baselines.** For the datasets containing CAD models, we adopt the commonly used ADD(s)-0.1d and 2D projection metric. supplement For the OnePose and OnePose-LowTexture datasets, we adopt the $cm$-$degree$ metric that is widely used in visual localization tasks. The predicted object pose is regarded as correct if the rotation and translation error are less than the defined thresholds respectively. We follow [5, 6] to evaluate the recall rates under $1cm$-$1deg$, $3cm$-$3deg$ and $5cm$-$5deg$.

We conduct a comprehensive comparison between our method and recent advanced object 6D pose estimation methods. In the one-shot setting, our method is most relevant to OnePose [5] and OnePose++ [6]. For OnePose++, we compare with the experiment results (denoted with †) by running the provided code, which are better than the results

TABLE II: **Experimental results on objects with CAD models.**

| Obj. ID | 0700 | 0706 | 0714 | 0721 | 0727 | 0732 | 0736 | 0740 | Avg |
|---|---|---|---|---|---|---|---|---|---|
| PVNet | 12.3 | 90.0 | 68.1 | 67.6 | 95.6 | 57.3 | 49.6 | 61.3 | 62.7 |
| †OnePose++ | 89.5 | 99.1 | 97.2 | **92.6** | 98.5 | 79.5 | 97.2 | 57.6 | 88.9 |
| MVP | **93.1** | **99.3** | **97.7** | 91.8 | **100.0** | **89.3** | **99.8** | **64.7** | **92.0** |

reported in the original paper. We also compare with the visual localization framework HLoc [36] with two settings: Superpoint [24] for description and SuperGLUE [37] for matching (SPP+SPG), detector-free 2D-2D feature matching method LoFTR [38]. Our method is also compared with Gen6D [21] that is also CAD-Model-free. In the instance-level setting, CDPN [9] and PVNet [10] are compared, in the scenarios when CAD models are available.

**Implementation Details.** We train our model on the same training set with [5, 6], which consists of 49 objects with rich texture. The depth of overall feature transform blocks $T$ is set to 4. The image overlap ratio threshold $\theta$ is set to $0.1$. The inlier threshold $\tau = 0.15$, the inlier ratio threshold $\rho = 0.2$. The feature confidence threshold $\delta$ is set to $0.1$. For intermediate pose estimation, we set reprojection error threshold to $12$ pixels. To fairly compare with one-shot baselines, the final object pose estimation setting is the same with [5, 6].

A crucial choice about where to use the graph-based pruning has to be made. The graph-based pruning should start as early as possible, so that only a small subset of 3D points are transferred to the subsequent feature transform blocks. We conduct the graph-based pruning only after the first feature transform block. For efficient training, we adopt a two-stage training strategy. By setting $t = 4$ in the coarse matching loss $\mathcal{L}_c$, we first train the complete model with fine matching loss $\mathcal{L}_f$. In the second stage, we only train a single feature transform block with $t = 1$ in $\mathcal{L}_c$ with other model parameters fixed.

### *B. Experiment Results*

**Results on OnePose and OnePose-LowTexture.** In these two datasets, we mainly focus on comparing with one-shot baselines. As shown in Table. I, our proposed method MVP consistently outperforms other methods on both two datasets. Since the OnePose dataset consists of many rich-textured objects, traditional detector-based HLoc (SPP+SPG) and OnePose can achieve satisfactory performance with sufficient keypoints and 2D-3D matches. In the OnePose-LowTexture dataset, our method shows considerable superiority comparing with †OnePose++, which overtakes other methods by a large margin. This is mainly due to our early elimination of irrelevant invisible 3D points and features, allowing more correct 2D-3D matches to be established. On average, our method takes 13363 3D points as input, and preserves 3448 (26%) 3D points before passing to the second coarse matching block. For the final object pose estimation, our method collects 415 2D-3D matches, in which 256 (52%) are inliers after RANSAC. Correspondingly, †OnePose++ obtains 451 2D-3D matches, in which only 220 (40%) are inliers. Fig. 4 gives a detailed qualitative comparison between our method and †OnePose++.

For the objects with CAD models in the OnePose-LowTexture dataset, we also compare with the instance-level method PVNet. As presented in Table. II, our method significantly outperforms PVNet using the ADD(S)-0.1d metric. As PVNet trains an individual model for each object using synthetic images from massive views, the performance bottleneck may be due to the statistic variance between the synthetic and real-world images. Under the ADD(S)-0.1d metric, our method still outperforms †OnePose++ in the subset, which further proves the robustness of our method.

**Results on LINEMOD.** We compare our method with both instance-level and one-shot methods on ADD(S)-0.1d and Proj2D metrics. As presented in Table. III, our method achieves the best performance under the one-shot paradigm. Gen6D requires an accurate object bounding box to start the pose refinement process. However, images of the LINEMOD dataset are of low resolution, making Gen6D prone to fail at the beginning. Due to the low-textured properties of objects in the LINEMOD dataset, OnePose performs worse than our method since the feature detector is difficult to find sufficient keypoints for description. Comparing with †OnePose++, our method rejects invisible 3D points from coarse matching and thus obtains more correct 2D-3D matches.

**Ablation Study and Runtime Analysis.** To fully evaluate different components of our method, we conduct four variants on the OnePose-LowTexture dataset and the results are given in Table. IV. In detail: a) Without the graph-based pruning module, the recall rate significantly drops. b) Only gathering 3D points that are visible in the seed camera without expansion also reduces the performance. c) We apply an additional geometric verification step with PnP+RANSAC before the pruning module, it brings a large computational overhead and a slight performance drop. d) Changing the feature confidence threshold $\delta$ in pruning from 0.1 to 0.2 barely changes the results. Our method is faster than †OnePose++ for two reasons. First, RANSAC stops earlier in the final object pose estimation due to fewer but higher quality 2D-3D matches.

## REFERENCES


[1] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[2] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7668–7677.

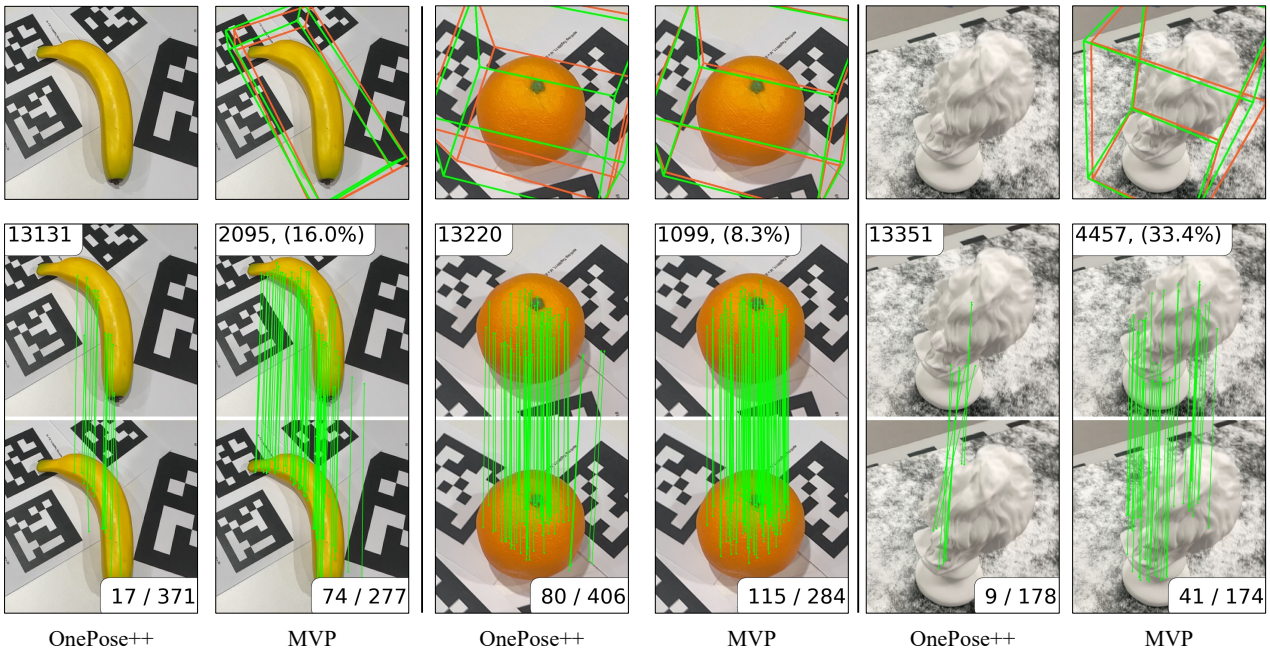


Fig. 4: where (→) means feature direction

TABLE III: **Experimental results on LINEMOD dataset.**

| Type | Name | Object Name | | | | | | | | | | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ape | benchwise | cam | can | cat | driller | duck | eggbox* | glue* | holepuncher | iron | lamp | phone | |
| | | *ADD(S)-0.1d* | | | | | | | | | | | | | |
| Instance-level | CDPN | 67.3 | 98.8 | 92.8 | 96.6 | 86.6 | 95.1 | 75.2 | 99.6 | 99.6 | 89.7 | 97.9 | 97.8 | 80.7 | 91.4 |
| | PVNet | 43.6 | 99.9 | 86.9 | 95.5 | 79.3 | 96.4 | 52.6 | 99.2 | 95.7 | 81.9 | 98.9 | 99.3 | 92.4 | 86.3 |
| One-shot | Gen6D | - | 77.0 | 66.1 | - | 60.7 | 67.4 | 40.5 | 95.7 | 87.2 | - | - | - | - | - |
| | OnePose | 11.8 | 92.6 | 88.1 | 77.2 | 47.9 | 74.5 | 34.2 | 71.3 | 37.5 | 54.9 | 89.2 | 87.6 | 60.6 | 63.6 |
| | †OnePose++ | 30.5 | 98.5 | 90.1 | 89.0 | **72.6** | 92.6 | 49.4 | 99.6 | 68.1 | 67.8 | 97.3 | 97.9 | 79.1 | 79.4 |
| | Ours | **34.2** | **98.8** | **92.8** | **90.9** | 71.6 | **95.3** | **53.4** | **99.8** | **71.4** | **72.0** | **98.1** | **98.4** | **82.1** | **81.5** |
| | | *Proj2D* | | | | | | | | | | | | | |
| Instance-level | CDPN | 97.5 | 98.8 | 98.6 | 99.6 | 99.3 | 94.9 | 98.4 | 99.1 | 98.4 | 99.5 | 97.9 | 95.7 | 96.8 | 98.0 |
| | PVNet | 99.2 | 99.8 | 99.2 | 99.9 | 99.3 | 96.9 | 98.0 | 99.3 | 98.5 | 100.0 | 99.2 | 98.3 | 99.4 | 99.0 |
| One-shot | OnePose | 35.2 | 94.4 | 96.8 | 87.4 | 77.2 | 76.0 | 73.0 | 89.9 | 55.1 | 79.1 | 92.4 | 88.9 | 69.4 | 78.1 |
| | †OnePose++ | 97.5 | **99.6** | 99.6 | 99.6 | 97.9 | 94.3 | 97.5 | 99.1 | 76.7 | 99.1 | 99.2 | 98.8 | 95.4 | 96.5 |
| | Ours | **97.8** | 99.5 | **99.7** | **99.8** | **98.5** | **95.0** | **97.9** | **99.2** | **77.5** | **99.2** | **99.4** | **99.0** | **96.0** | **96.8** |

TABLE IV: Ablation study on the OnePose-LowTexture dataset. a) without graph-based pruning module. b) without expansion when pruning. c) apply geometric verification (gv) before pruning. d) change the feature confidence threshold $\delta$ from 0.1 to 0.2 when pruning.

| Method | recall (%) | | | time (ms) |
|---|---|---|---|---|
| | 1cm-1deg | 3cm-3deg | 5cm-5deg | |
| †OnePose++ | 17.2 | 58.7 | 73.1 | 98.4 |
| MVP | **18.4** | **62.2** | **77.1** | 75.2 |
| ↳a) w/o pruning | 17.7 | 56.1 | 73.4 | 110.8 |
| ↳b) w/o expansion | 17.7 | 60.6 | 75.5 | **66.0** |
| ↳c) w/ gv | 18.2 | 61.7 | 76.5 | 103.2 |
| ↳d) $\delta = 0.2$ | 18.2 | 62.1 | **77.1** | 72.7 |


[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2642–2651.

[4] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7822–7831.

[5] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou, "Onepose: One-shot object pose estimation without cad models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6825–6834.

[6] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou, "Onepose++: Keypoint-free one-shot object pose estimation without cad models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 103–35 115, 2022.

[7] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[9] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7678–7687.

[10] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4561–4570.

[11] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *Proceedings of the IEEE/CVF international conference*

*on computer vision*, 2019, pp. 1941–1950.

[12] M. Tian, M. H. Ang, and G. H. Lee, "Shape prior deformation for categorical 6d object pose and size estimation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 530–546.

[13] J. Wang, K. Chen, and Q. Dou, "Category-level 6d object pose estimation via cascaded relation and recurrent reconstruction networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4807–4814.

[14] T. Lee, B.-U. Lee, M. Kim, and I. S. Kweon, "Category-level metric scale object shape and pose estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8575–8582, 2021.

[15] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1323–1330.

[16] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose guided rgbd feature learning for 3d object pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3856–3864.

[17] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[18] M. Sundermeyer, M. Durner, E. Y. Puang, Z.-C. Marton, N. Vaskevicius, K. O. Arras, and R. Triebel, "Multi-path learning for object pose estimation across domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 916–13 925.

[19] I. Shugurov, F. Li, B. Busam, and S. Ilic, "Osop: A multi-stage one shot object pose estimation framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6835–6844.

[20] M. Cai and I. Reid, "Reconstruct locally, localize globally: A model free method for object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3153–3163.

[21] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang, "Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images," in *European Conference on Computer Vision*. Springer, 2022, pp. 298–315.

[22] Z. Fan, P. Pan, P. Wang, Y. Jiang, D. Xu, H. Jiang, and Z. Wang, "Pope: 6-dof promptable pose estimation of any object, in any scene, with one reference," *arXiv preprint arXiv:2305.15727*, 2023.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[25] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[26] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[27] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. Ieee, 2006, pp. 2161–2168.

[28] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*. Springer, 2010, pp. 791–804.

[29] S. Choudhary and P. Narayanan, "Visibility probability structure from sfm datasets and applications," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*. Springer, 2012, pp. 130–143.

[30] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *European conference on computer vision*. Springer, 2012, pp. 15–29.

[31] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.

[32] L. Liu, H. Li, and Y. Dai, "Efficient global 2d-3d matching for camera localization in a large-scale 3d map," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2372–2381.

[33] W. Cheng, W. Lin, K. Chen, and X. Zhang, "Cascaded parallel filtering for memory-efficient image-based localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1032–1041.

[34] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2704–2712.

[35] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1455–1461, 2016.

[36] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[37] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[38] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8922–8931.

[39] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International conference on machine learning*. PMLR, 2020, pp. 5156–5165.

[40] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," *arXiv preprint arXiv:2306.13643*, 2023.

[41] F. Xue, I. Budvytis, and R. Cipolla, "Imp: Iterative matching and pose estimation with adaptive pooling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 317–21 326.

[42] H. Zhang and K. E. Hoff III, "Fast backface culling using normal masks," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, 1997, pp. 103–ff.

[43] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*. Springer, 2013, pp. 548–562.