iG-6DoF: Model-free 6DoF Pose Estimation for Unseen Object via Iterative 3D Gaussian Splatting

Tuo Cao[1], Fei Luo[1], Jiongming Qin[1], Yu Jiang[1], Yusen Wang[1], and Chunxia Xiao[1*]
[1]School of Computer Science, Wuhan University, Wuhan, Hubei, China
{maplect,luofei,jiongming,jiangyu1181,wangyusen,cxxiao}@whu.edu.cn
http://graphvision.whu.edu.cn/

Abstract

Traditional methods in pose estimation often rely on precise 3D models or additional data such as depth and normals, limiting their generalization, especially when objects undergo large translations or rotations. We propose iG-6DoF, a novel model-free 6D pose estimation method using iterative 3D Gaussian Splatting to estimate the pose of unseen objects. We first estimates an initial pose by leveraging multi-scale data augmentation and the rotation-equivariant features to create a better pose hypothesis from a set of candidates. Then, we propose an iterative 3DGS approach through iteratively rendering and comparing the rendered image with the input image to further progressively improve pose estimation accuracy. The proposed method consists of an object detector, a multi-scale rotation-equivariant feature based initial pose estimator, and a coarse-to-fine pose refiner. Such combination allows our method to focus on the target object in a complex scene dealing with large movement and weak textures. Our method achieves state-of-the-art results on the LINEMOD, OnePose-LowTexture, GenMOP datasets and our self-captured data, demonstrating its strong generalization to unseen objects and robustness across various scenes.

1. Introduction

Estimating the rotation and translation parameters of objects within images has been a longstanding and widely studied problem in computer vision. It has extensive applications in virtual reality, robotic manipulation, and autonomous driving. Early pose estimation methods [10, 11, 26, 48, 61, 62, 71] primarily focused on pose estimation at instance-level, requiring the target object to be included in the training set. They often lack generalization capabilities and hinder the estimation of unseen objects. Subsequently, researchers introduced category-level pose estimation methods [15, 19, 63, 73, 77], which can estimate the pose parameters of objects within the same category, even if the specific instance is not present in the training set. They demonstrate a degree of generalization.

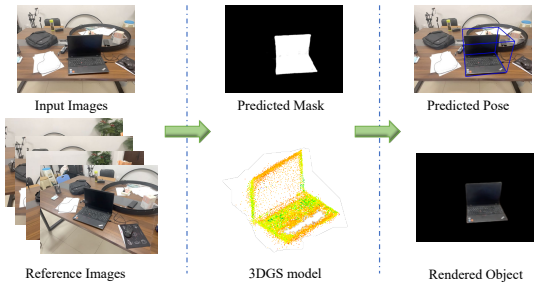

Figure 1. Given a set of reference images and an input image, our method outputs the object mask, constructs a 3D Gaussian model, and estimates its 6D pose.

Recently, research has increasingly focused on generalizable pose estimation, aiming to develop a universal model to estimate an object's pose using only its CAD model or a few specific-view images [41]. Existing generalizable pose estimation methods can be primarily categorized into two types. The first type is CAD model-based. These methods [14, 21, 22, 36, 50] typically utilize the 3D or texture information of a precise CAD model as prior knowledge. They often employ feature-matching techniques to obtain 2D-3D correspondences between the query image and the CAD model. Then, they calculate pose parameters using traditional numerical algorithms such as PnP [20] or ICP [6]. The second type is model-free object pose estimation. These methods [8, 13, 25, 27, 42, 59] do not require precise CAD models but rely on a set of annotated reference images of the object. Multi-view stereo geometry provides geometric information about the object as prior knowledge. Compared to CAD-based methods, model-free methods offer greater potential for practical applications without the need to acquire accurate CAD models.

*Chunxia Xiao is corresponding author.

However, current model-free methods have certain limitations. For instance, FS6D [27] requires additional depth information for supervision, Gen6D [42] relies solely on 2D representations and struggles with large object movements and rotations. OnePose [59] necessitates establishing 2D-3D correspondences, which can lead to suboptimal performance in weak-texture regions. To address these issues, we propose a pose estimation network based on the multi-scale rotation-equivariant feature and the 3D Gaussian Splatting (3DGS). The core idea is to utilize multi-scale information to tackle challenges posed by large-scale movements and leverage the high-quality rendering capabilities of 3DGS to handle pose estimation for weak textures.

As illustrated in Figure 1, our method takes a set of reference images and an input image to output the object's mask, construct a 3D Gaussian model, and determine the object's 6D pose. Unlike traditional methods that match the query image to the closest reference image, which often results in inaccurate initial poses due to sparse reference data, our approach employs multi-scale data augmentation of reference images and builds a feature vector space on the icosahedral group to estimate the initial pose. Then, we refine this pose by iteratively searching the surrounding neighborhood, utilizing the high-quality rendering capabilities of 3DGS [33]. The key contributions of this work can be summarized as follows:

- We propose a novel end-to-end object pose estimation method that enables direct pose estimation of unseen objects without retraining.
- To enhance initialization accuracy, we introduce a multi-scale icosahedral group feature matching module, improving initial pose estimation precision.
- Finally, we incorporate a 3DGS-based rendering-and-comparison module for fast and accurate iterative pose optimization.

2. Related works

2.1. Model-based Unseen Object Pose Estimation

CAD model-based methods incorporate detailed 3D object models as prior knowledge to accurately determine the position and orientation of previously unseen instances within a scene. Pitteri *et al*. pioneered using CAD models for 3DoF pose estimation by approximating object geometry with corner points [50]. However, this approach was limited to objects with distinct corners. To address this, they subsequently introduced an embedding method to capture local 3D geometry, enabling 2D-3D correspondence establishment and PnP+RANSAC-based pose estimation [49]. However, both methods were confined to estimating only three degrees of freedom.

Building upon point cloud registration techniques for unseen objects, Zhao *et al*. [75] introduced a geometry correspondence-based approach using generic, object-agnostic features to establish robust 3D-3D correspondences. However, this method required external methods like Mask-RCNN [24] for object class and segmentation mask determination. To address this limitation, Chen *et al*. [14] presented ZeroPose, a framework for joint instance segmentation and pose estimation of unseen objects. Leveraging SAM [34], they generated object proposals and employed template matching for instance segmentation. A hierarchical geometric feature matching network based on GeoTransformer [53] was used to establish correspondences. Expanding on ZeroPose, Lin *et al*. [40] introduced a refined matching score considering semantics, appearance, and geometry for improved segmentation. For pose estimation, they developed a two-stage partial-to-partial point matching model to effectively construct dense 3D-3D correspondences. FoundPose [46] put forward a rapid template retrieval approach which founded on visual words created from DINOv2 [45] patch descriptors. As a result, it reduces the dependence on large amounts of data and boosts the matching speed. Freeze [12] represents the initial technique that harnesses the synergy between geometric and vision foundation models to estimate the pose of unseen objects.

2.2. Model-free Unseen Object Pose Estimation

In contrast to CAD model-based approaches, manual reference view-based methods bypass the need for object CAD models by relying on manually labeled reference images. These methods primarily establish correspondences between the query image and reference views, either in 3D-3D or 2D-3D space, to determine object pose. He *et al*. [27] introduced a pioneering few-shot 6DoF pose estimation method using a transformer-based dense RGBD prototype matching framework to correlate query and reference views without additional training. Corsetti *et al*. [32] employed textual prompts for object segmentation and reformulated the problem as relative pose estimation between scenes, solved through point cloud registration.

Sun *et al*. [59] adapted visual localization techniques for pose estimation by constructing a Structure from Motion (SfM) model of the unseen object using reference view RGB sequences. A graph attention network matched 2D query image keypoints with 3D points in the SfM model. However, this approach suffered from poor performance on low-textured objects due to reliance on repeatable keypoints. He *et al*. [25] addressed this limitation by introducing a keypoint-free SfM method to reconstruct semi-dense point cloud models of low-textured objects using the detector-free feature matching method LoFTR [58]. Recognizing the suboptimal performance of pre-trained feature matching models [54, 58] for pose estimation, Castro *et al*. [13] redesigned the training pipeline using a three-view system for one-shot object-to-image matching. In ad-

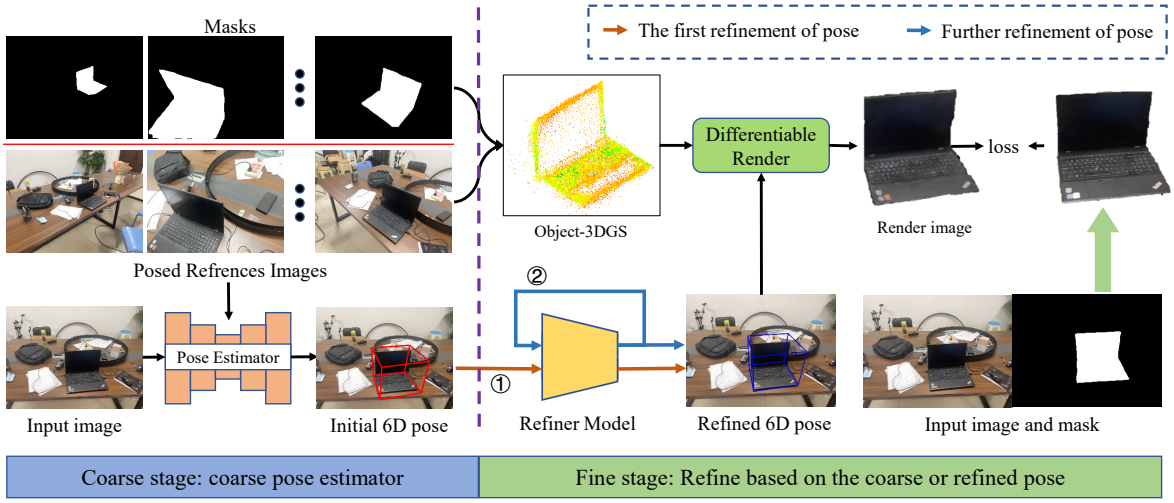

Figure 2. **Overview of iG-6DoF.** Our method employs a coarse-to-fine approach, where the pose estimator first estimates an initial pose from the input image, and then the pose refiner is employed to achieve a precise final pose.

dition to this, FoundationPose [69] has constructed a unified framework for handling both model-based and model-free scenarios simultaneously.

2.3. Pose Estimation with Neural Rendering

Recently, some methods based on Neural Rendering (NeRF[44] and 3DGS[33]) have made significant strides in representing three-dimensional scenes [4, 5, 7, 23, 28, 52, 65, 66]. These methods train a neural network to minimize the errors between rendered images and real images, thereby modeling the color and volumetric density of a scene as a function of spatial position, thereby enabling high expressiveness for complex three-dimensional environments. Several efforts have applied this framework to tasks such as pose estimation and Simultaneous Localization and Mapping (SLAM) [16, 31, 36, 57, 76]. For instance, the iNeRF method [72] assumes a given camera pose, creates an image via the rendering process and proceeds to contrast the pixel variances with the query image. The gradient data obtained in this way is then applied to iteratively adjust the camera pose in a step-by-step manner until the rendered image matches the query image precisely. Similarly, Nerf-pose[38] makes use of NeRF's implicit portrayal of 3D scenes and trains a pose regression network so as to set up associations between 2D and 3D data. iComMa [60] inverts 3DGS to achieve accurate pose estimation without training, using a gradient-based framework and an end-to-end matching module to improve robustness and precision under difficult conditions. Although these methods can achieve an accurate pose estimation using pixel-level comparison losses, they encounter difficulties in achieving effective convergence in complex situations. Specifically, when there is a substantial disparity between the rendered images and the query images, it becomes a bottleneck for precise pose estimation.

3. Method

Given a set of reference images of an object with known camera poses and intrinsics, our goal is to estimate the 6D pose (translation $\mathbf{T} = (t_x, t_y, t_z) \in \mathbb{R}^3$ and rotation $R \in \mathbb{SO}^3$) of the same object in a query image. The pose transformation maps points from the object coordinate system to the camera coordinate system using the equation $P_{cam} = \mathbf{R}P_{obj} + \mathbf{T}$.

As illustrated in Figure 2, iG-6DoF comprises three primary modules: an object detector, an initial pose estimator, and a pose refiner. The object detector segments the object region within the image (Section 3.2). Subsequently, the initial pose estimator determines an initial rotation and translation by identifying the most similar feature within a multi-scale SO(3) group feature space (Section 3.3). Upon the initial translation and rotation, the 3DGS pose refiner computes a precise pose estimate (Section 3.4).

3.1. Preliminaries

Data Acquisition. To implement our method, we require a set of reference images with parameters $\{I_i^{ref}, R_i^{ref}, T_i^{ref}\}_{i=1}^{N_r}$, where I, R, and T represent the image and its corresponding camera extrinsics. N_r is the number of reference images. Owing to off-the-shelf toolboxes provided by OnePose [59] and ARKit [3], we can easily manually annotate the 3D bounding box of an object in a video sequence and obtain camera parameters.

3D Gaussian Splatting. 3DGS is a recent and innovative technique for representing and rendering 3D scenes. They

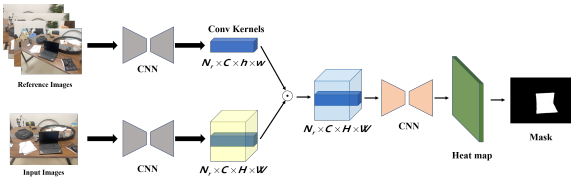

Figure 3. Detector architecture: We use the features from reference images as kernels to convolve with query image features, generating heap maps. This heap maps are then processed by a CNN to produce a object mask.

first recover camera poses and sparse 3D point clouds of the scene from a sequence of captured images using Structure from Motion (SfM), and then construct 3D Gaussian spheres based on these point clouds. Each 3D Gaussian is parameterized by a 3D coordinate $\mu \in \mathbb{R}^3$, a 3D rotation quaternion $r \in \mathbb{R}^4$, a scale vector $s \in \mathbb{R}^3$, an opacity factor $\alpha \in \mathbb{R}$, and spherical harmonic coefficients $h \in \mathbb{R}^k$, where k denotes the number of degrees of freedom. Finally, We can calculating the loss between the rendered image and the real image, and using the backpropagation algorithm to optimize the Gaussian parameters.

3.2. Object Detector

Our detector builds on the TGID [2] and Gen6D [42] frameworks, which apply a correlation-based object detector. Since we need to construct a 3DGS model of the object, a more precise object mask is required, so we replace the output bounding box with a segmentation mask. Specifically, we set a per-pixel confidence score, and pixels are considered part of the target object when their confidence exceeds a certain threshold. The core idea is to use TDID embeddings to convolve the feature map of the reference image over the query image features, calculating the correlation for each pixel. A threshold is then set to identify high-confidence pixels as belonging to the target object, resulting in the object's mask.

As shown in Figure 3, our detector architecture employs a shared feature extractor, like VGG-11 [56], to extract features from the target and scene images. These features are subsequently combined in a joint embedding layer. Finally, a set of convolutions predicts class scores and segmentation mask regression parameters for a set of default anchor boxes on the embedding feature map.

3.3. Initial Pose Estimator

The primary objective of the initial pose estimator is to select the most accurate pose hypothesis from a set of candidates. Previous methods often relied on template matching, where the closest match to the query image is selected from a reference image database. However, due to the sparsity of viewpoints in the reference image set, this approach can lead to significant errors, particularly when the query image's viewpoint differs substantially from those in the reference set.

As shown in Figure 4, we first apply multi-scale data augmentation to the reference images to enrich the candidate pose database. Specifically, each reference image is rotated $k\pi/2$ clockwise and scaled by factors of 2 and 0.5, respectively. Inspired by RoReg [64] and GIFT [43], we utilize rotation-equivariant feature to embed the reference images. Specifically, we treat the RGB color values as 3D coordinates in a three-dimensional space, establishing a mapping from the color space to the 3D space, so that we can apply point set feature extractor PointNet [51] as backbone to extract 3D feature from 2D image. To prevent the same color at different positions from being mapped to a single 3D point, we added positional encoding [44]. Subsequently, we define a neighborhood space on the 2D image and employ a icosahedral group feature encoder to encode the reference images, yielding a multi-scale group feature space $\{V_i^{ref}\}_{i=1}^{N_r} \in \mathbb{R}^{60 \times N_r}$. In a similar manner, a feature vector $V^{que} \in \mathbb{R}^{60}$ is extracted from the query image. To obtain the initial pose parameters, we compute the cosine similarity between V^{que} and each reference feature vector V_i^{ref}. The reference vector with the highest similarity score is selected, and its associated pose parameters are assigned as the initial estimate.

Group Feature Space. Given a target image that has been segmented using a mask, we employ our proposed method to project each pixel within the segmentation mask onto a corresponding 3D point in space, resulting in a set of 3D points denoted as $\{P_i \in \mathbb{R}^3\}$. To establish local neighborhoods for each pixel, we define $N_P = \{p_i | \|p_i - p\| < 5\}$, where N_P represents the neighborhood of pixel p, and p_i denotes the position of a neighboring pixel located within a 5-pixel radius of p.

Given an input neighborhood point set N_P, we apply an element g of the icosahedral group G to generate rotated point sets. Each rotated point set is processed by a shared point set feature extractor, denoted as ϕ, to produce an n-dimensional feature vector, expressed as:

$$f_0(g) = \phi(T_g \circ N_P), \tag{1}$$

where $f_0 : G \to \mathbb{R}^{n_0}$ represents the output group feature for point p, and $T_g \circ N_P$ denotes the application of rotation g to the point set N_P. Since the icosahedral group G comprises 60 rotations, the group feature f_0 can be efficiently stored as a $60 \times n_0$ matrix. We apply PointNet [51] as backbone ϕ. Then, we adopt a localized icosahedral group convolution for feature embedding:

$$[f_{l+1}(g)]_j = \sum_i^{13} w_{j,i}^T f_l(h_i g) + b_j, \tag{2}$$

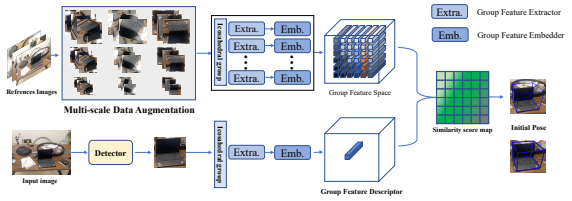

Figure 4. **Architecture of the pose estimator.** We first apply multi-scale image augmentations to the reference images, including rotations and scaling. Subsequently, we extract rotation-equivariant features using the icosahedral group. Finally, the optimal initial pose is determined by comparing the similarity of the feature vectors.

where l denote the layer index, $f_l(g) \in \mathbb{R}^{n_l}$ and $f_{l+1}(g) \in \mathbb{R}^{n_{l+1}}$ represent the input and output feature vectors, respectively. $[\cdot]_j$ extracts the j-th element from a vector. The neighborhood set $h_i \in H$ is denoted by where each is an element of the group G. The trainable weight associated with the i-th neighbor and j-th output feature is represented by $w_{j,i} \in \mathbb{R}^{n_k}$, with being the corresponding bias b_j. Note that j ranges from 1 to n, indexing the output feature dimensions. Given the group's closure property, the composition $h_i g$ is also an element of G.

3.4. Pose Refiner

The pose refiner aims to refine an initial pose $\mathcal{T}_{\text{init}}$ with an input image. To achieve this, we leverage the high rendering quality of 3DGS [33]. By iteratively rendering and comparing the rendered image with the input image, we progressively update the pose estimate until convergence. As shown in Figure 5, the refiner takes as input $\mathcal{T}_{\text{init}}^k$ and a 3DGS model and predicts an updated pose $\mathcal{T}_{\text{init}}^{k+1} = \mathcal{T}_{\Delta}^{k+1}\mathcal{T}_{\text{init}}^k$ and a rendered images I_{render}^{k+1}. We iteratively refine the pose parameters by minimizing the SSIM loss between the rendered and input images I_{que}. Similar to [35, 36, 39], we decompose $\mathcal{T}_{\Delta}^{k+1}$ into its rotational component R_{Δ}^{k+1} and translational component T_{Δ}^{k+1} (Note that $\mathcal{T} \in \mathbb{SE}(4)$ and $T \in \mathbb{R}^3$). To decouple the rotation and translation components, the rotation center is shifted from the camera origin to the object's center, as determined by the current pose estimate. This modification ensures that applying a rotation does not alter the object's position within the camera frame. The iterative optimization process of the refiner is as follows:

$$\mathcal{T}_{\Delta}^{k+1} = \arg\min_{T_{\Delta}^{k+1}} \mathcal{L}_T(\mathcal{R}_{gs}(T_{\Delta}^{k+1} + \mathcal{T}^k, GSM), I_{que}) + \arg\min_{R_{\Delta}^{k+1}} \mathcal{L}_R(\mathcal{R}_{gs}(R_{\Delta}^{k+1} \odot (T_{\Delta}^{k+1} + \mathcal{T}^k), GSM), I_{que}), \tag{3}$$

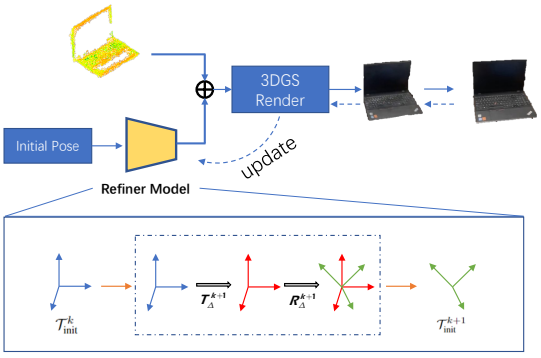

Figure 5. **Diagram of pose refiner.** Given the pose from the previous time step $\mathcal{T}_{\text{init}}^k$, we decouple $\mathcal{T}_{\Delta}^{k+1}$ into R_{Δ}^{k+1} and T_{Δ}^{k+1} for separate estimation. We first estimate the translation vector, followed by the rotation vector. This process is iterated until reaching the specified number of steps or convergence.

where $\mathcal{R}_{gs}$ denotes the 3D gaussian renderer, $\odot$ signifies the application of a rigid rotation and GSM is a 3DGS model.

3.5. Loss Functions

We use the widely adopted Binary Cross Entropy (BCE) loss to train our detector for pixel-wise segmentation, denoted as $\mathcal{L}_{det}$:

$$\mathcal{L}_{det} = \mathcal{L}_{BCE}(M, \bar{M}), \tag{4}$$

where M and $\bar{M}$ represent the predicted and ground truth segmentation masks, respectively.

We apply the descriptor construction loss from RoReg [64] to train pose estimator. Given a batch of ground-truth image pairs (I_q, I_r) and their corresponding ground-truth rotations R_{I_q}, we compute the outputs of the group feature embedder, which include the rotation-invariant descriptors $(d_{I_q}, d_{I_r}^+)$, the rotation-equivariant group features $(f_{I_q}, f_{I_r}^+)$, and the corresponding ground truth coarse rotations $g_{I_r}^+$. For every sample in the batch, we compute the loss:

$$\mathcal{L}_1(d, d^+, D^-) = \frac{e^{||d-d^+||_2} - \min\limits_{d^- \in D^-} e^{||d-d^-||_2}}{e^{||d-d^+||_2} + \sum\limits_{d^- \in D^-} e^{||d-d^-||_2}} \tag{5}$$

$$\mathcal{L}_2(f, f^+, g^+) = -log(\frac{e^{\langle f, P_{g^+} \circ f^+ \rangle}}{\sum\limits_{g \in G} e^{\langle f, P_g \circ f^+ \rangle}}) \tag{6}$$

$$\mathcal{L}_{group} = \lambda * \mathcal{L}_1(d, d^+, D^-) + \mathcal{L}_2(f, f^+, g^+), \tag{7}$$

where the subscript I_r is omitted for simplicity. Equation 5 supervises the rotation-invariant descriptor, where d is the descriptor, d^+ is the matched descriptor, D^- are the negative descriptors in the batch, and $|\cdot|_2$ is the L2 norm.

Finally, based on the aforementioned L_{pose} defined as

$$\mathcal{L}_{pose} = \mathcal{L}_R + \mathcal{L}_T \tag{8}$$

$$\mathcal{L}_T = \mathcal{L}_{SSIM}, \tag{9}$$

$$\mathcal{L}_R = \mathcal{L}_{SSIM} + \mathcal{L}_{MS-SSIM}, \tag{10}$$

where $\mathcal{L}_{SSIM}$ and $\mathcal{L}_{MS-SSIM}$ represent the SSIM-based [68] and multi-scale SSIM-based [67] loss functions, respectively. The overall loss function of our method is:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{det} + \lambda_2 \mathcal{L}_{group} + \lambda_3 \mathcal{L}_{pose}, \tag{11}$$

where $\lambda_{\{1,2,3\}}$ represent the hyperparameters, which we set to 0.3, 0.2, and 0.5, respectively.

4. Experiments

Training Data. We employ the synthetic MegaPose dataset [36] for training, which generated using BlenderProc [17] with 1,000 diverse objects from the Google Scanned Objects dataset [18], comprising one million synthetic RGB images.

Evaluation data. We evaluate our proposed model on three widely used benchmarks: LINEMOD, OnePose-LowTexture, and GenMOP, to demonstrate its generalization ability across diverse object categories and scenes. The LINEMOD dataset [29], comprising 13 objects, is a commonly employed benchmark for 6D object pose estimation. Adhering to the established protocol [25, 37, 42, 47, 59], the training partition of LINEMOD is designated as reference data, while the testing partition serves as the evaluation set. The OnePose-LowTexture dataset [59] presents a challenging scenario with objects exhibiting minimal or absent texture, containing eight scanned objects for evaluation. The GenMOP [42] dataset comprises ten distinct objects. For each object, two video sequences were captured under varying environmental conditions, including background and lighting variations. Each video sequence is segmented into approximately 200 individual images

Metrics. To evaluate our model, we employ the commonly used Average Distance (ADD) metric [29] and projection error. For ADD, we calculate both the recall rate at 10% of the object diameter (ADD-0.1d) and the Area Under the Curve (AUC) within a 10 cm radius (ADD-AUC). Regarding projection error, we compute the recall rate at a pixel threshold of 5 (Prj-5).

Setups. We primarily compare iG-6DoF against Gen6D [42], Cas6D [47], Onepose [59], GS-Pose [8] and MFOS [37]. To ensure a fair comparison and demonstrate the effectiveness of each module, we evaluated our initial pose estimator and pose refiner on the aforementioned three separate datasets.

4.1. Results on LINEMOD

We first evaluate iG-6DoF on a subset of LINEMOD objects against OSOP [55], Gen6D [42], Cas6D [47], GS-Pose [8] and LocPoseNet [74] and present quantitative results in Table 1. Without pose refinement, iG-6DoF achieves an ADD(S)-0.1d of 45.99%. After refinement, performance improves to 83.22%.

Then, we compare our method against state-of-the-art one-shot approaches, including Gen6D [42], OnePose [59], OnePose++ [25] and MFOS [37], using ADD(S)-0.1d and Proj2D metrics. As indicated in Table 2, our method consistently outperforms these baselines. Notably, unlike OnePose and OnePose++ which rely on pre-reconstructed 3D shape models, our approach operates without requiring prior 3D object knowledge. This leads to improvements of 8.2% and 2.3% on ADD-S and Proj2D, respectively, over the strongest baseline.

4.2. Results on OnePose-LowTexture

We then evaluate iG-6DoF on the challenging OnePose-LowTexture dataset [25], comparing it against state-of-the-art baselines including OnePose [59], OnePose++ [25], Gen6D [42], and the instance-specific PVNet [48]. Table 3 presents quantitative standard cm-degree accuracy for different thresholds, demonstrating the superior performance of iG-6DoF. Specially, our method outperforms all baseline methods at the threshold 1cm/1deg and 5cm/5deg. OnePose++ eliminates reliance on local feature matching by adopting the keypoint-free LoFTR [58], improving performance Onepose to 72.1%, yet still falls short of iG-6DoF despite requiring ground-truth bounding boxes.

4.3. Results on GenMOP

We finally compare iG-6DoF with generalizable image-matching based ObjDesc [1], two instance-specific estimators PVNet [48] and RLLG [9] and model-free method Gen6D [42] on GenMOP dataset. To ensure a fair comparison, we adopt the same experimental setup as Gen6D, using the original reference images without data augmentation. All testing data is unseen during the training of iG-6DoF, Gen6D, and ObjDesc. For PVNet and RLLG, we train a separate model for each object. Quantitative results are shown in Table 5, our method essentially achieves the current state-of-the-art performance.

4.4. Ablation Study

To verify the effectiveness of each module in our proposed method, we conducted ablation studies on the widely used LM [29] dataset. Performance is assessed using the BOP [30] metric.

Ablation study on the pose estimator. To demonstrate the designs in the initial pose estimator, we conduct ablation studies on the LM dataset and results are shown in Table 4

Method	Pose Refiner	cat	duck	bvise	cam	driller	Avg.
OSOP [55]	w/o	34.43	20.08	50.41	32.30	43.94	36.23
Gen6D [42]		15.97	7.89	25.48	22.06	17.24	17.73
LocPoseNet [74]		-	-	-	-	-	27.27
GS-Pose [8]		**47.80**	30.70	**63.47**	**44.61**	47.27	**46.77**
iG-6DoF (Ours)		46.53	**31.61**	61.97	41.55	**48.31**	45.99
OSOP [55]	w/	42.54	22.16	55.59	36.21	49.57	42.21
Gen6D [42]		60.68	40.47	77.03	66.67	67.39	62.45
Cas6D [47]		60.58	51.27	86.72	70.10	84.84	70.72
iG-6DoF (Ours)		**80.89**	**66.39**	**95.88**	**87.23**	**85.69**	**83.22**

Table 1. Quantitative results on a subset of objects from the LINEMOD dataset [29] in terms of ADD(S)-0.1d. The best performance is highlighted in bold.

Method	Object Name													Avg.
	ape	benchwise	cam	can	cat	driller	duck	eggbox*	glue*	holepuncher	iron	lamp	phone	
	ADD(S)-0.1d													
Gen6D	-	62.1	45.6	-	40.9	48.8	16.2	-	-	-	-	-	-	-
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++	31.2	**97.3**	88.0	89.8	70.4	**92.5**	42.3	**99.7**	48.0	69.7	**97.4**	**97.8**	**76.0**	76.9
MFOS	47.2	73.5	87.5	85.4	80.2	92.4	60.8	99.6	69.7	93.5	82.4	95.8	51.6	78.4
Ours	**64.3**	96.3	**88.6**	**92.1**	**83.2**	88.6	**73.3**	99.6	**81.3**	**94.3**	81.3	88.6	73.1	**85.1**
	Proj2D													
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++	97.3	**99.6**	**99.6**	**99.2**	98.7	93.1	**97.7**	98.7	51.8	98.6	**98.9**	**98.8**	**94.5**	94.3
MFOS	97.1	94.1	98.4	98.2	98.4	**95.7**	96.3	99.0	94.8	**99.3**	94.6	94.2	88.9	96.1
Ours	**97.8**	99.2	97.8	98.2	**99.1**	91.5	97.6	**99.3**	**95.1**	98.9	95.2	95.6	90.3	**96.6**

Table 2. Results on LINEMOD and comparison with other model-free baselines. Symmetric objects are indicated by *. The best performance is highlighted in bold, while the second best results are underlined.

	GT-Mask	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
PVNet	✓	15.1	33.2	48.6
Gen6D	✗	11.5	31.6	25.9
OnePose	✓	12.4	35.7	45.4
OnePose++	✓	16.8	**57.7**	72.1
MFOS	✓	14.1	54.3	74.2
Ours	✗	16.6	53.2	73.5
Ours	✓	**17.2**	55.6	**75.1**

Table 3. Comparison with Baselines on OnePose-LowTexture. We denote the methods relying on an GT object mask as 'GT-Mask'.

Row	Method	LM		
		AR_VSD	AR_MSSD	AR_MSPD
A0	iG-6DoF	0.549	0.689	0.853
B1	A0: GS refiner→ Gen6D refiner	0.538	0.672	0.812
B2	A0: GS refiner→ DeepIM refiner	0.512	0.638	0.779
C1	A0: Pose Estimator → Objdesc selector	0.424	0.503	0.637
C2	A0: Pose Estimator → Gen6D selector	0.432	0.511	0.669
D1	A0: w/o data augmentation	0.521	0.624	0.801
D2	B1: w/o data augmentation	0.501	0.613	0.786
D3	B2: w/o data augmentation	0.478	0.601	0.732
E0	A0: $N_r \rightarrow 16$	0.432	0.492	0.766
E1	A0: $N_r \rightarrow 32$	0.446	0.624	0.789
E2	A0: $N_r \rightarrow 64$	0.533	0.657	0.834
E3	A0: $N_r \rightarrow 128$	0.587	0.712	0.866

Table 4. Ablation study under BOP setup on LM dataset.

(C1 and C2). We select ObjDesc [70] and Gen6D [42] for the comparison baseline. The results show that our method is capable of achieving a more accurate initial pose because we search within a multi-scale pose hypothesis space, whereas the baseline method only selects the most similar candidate from the reference image as the initial pose.

Ablation study on the pose refiner. To highlight the advantages of our 3DGS-based refiner for unseen objects over other 6D pose estimation methods, such as those used in Gen6D and DeepIM [39, 42], we present results in Table 4 (B1 and B2). For the baseline refiner, DeepIM [39], we treat the reference image selected by our selector as the rendered image and use DeepIM to match it with the query image to update the pose. It is important to note that further refinement using additional iterations of DeepIM is not feasible, as there is no object model available to render a new image based on the updated pose. All refiners, including DeepIM, Gen6D, and our 3DGS-based refiner, are trained on the same dataset. The results indicate that our 3DGS-based refiner demonstrates superior generalization capabil-

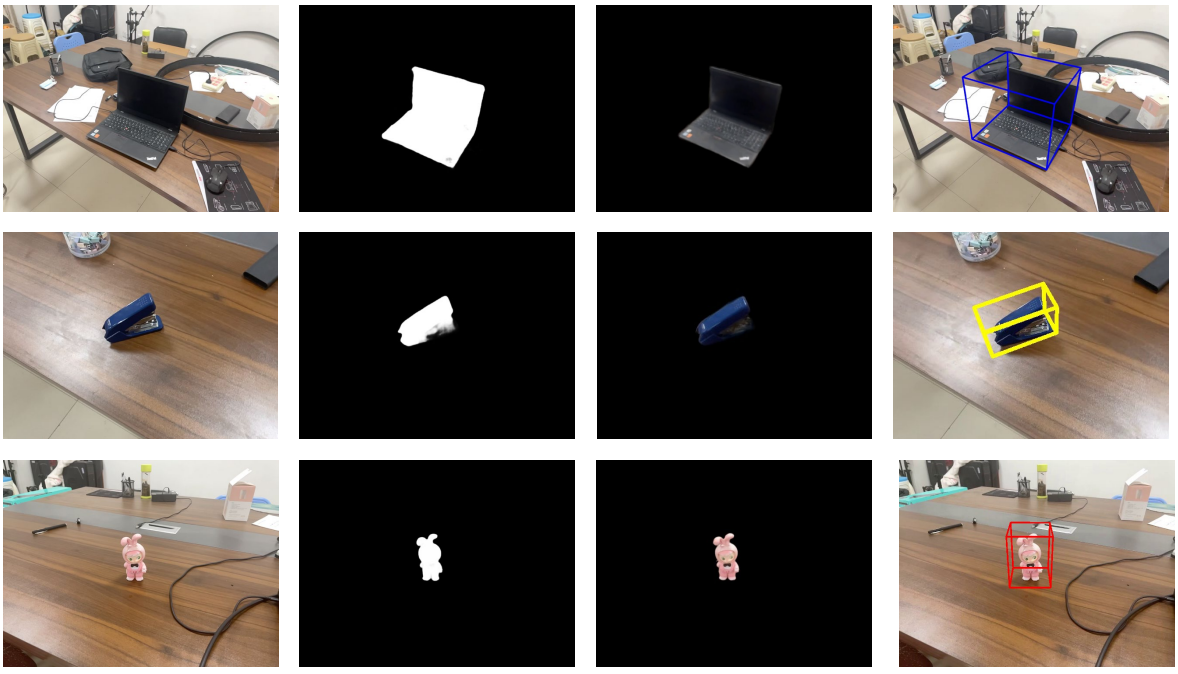

Figure 6. Qualitative results captured by us in real-world scenes. *More visual results, discussion and analysis are provided in the supplementary material.*

Metrics	Method	Object Name					avg.
		Chair	PlugEN	Piggy	Scissors	TFormer	
ADD-0.1d	ObjDesc [70]	3.50	5.14	14.07	1.25	7.54	8.55
	Gen6D w/o Ref.	14.00	7.48	39.70	16.81	11.51	17.90
	Gen6D w Ref.	61.50	19.63	**75.38**	32.76	**62.70**	50.39
	Ours w/o Ref.	46.32	17.93	71.84	29.57	55.92	44.32
	Ours w Ref.	**66.83**	**32.61**	**79.84**	**40.35**	60.81	**56.10**
Proj2D	ObjDesc [70]	4.00	10.75	4.52	18.53	8.33	9.23
	Gen6D w/o Ref.	11.50	40.65	33.17	34.05	64.29	36.73
	Gen6D w Ref.	55.00	72.90	92.96	**93.53**	**98.81**	82.64
	Ours w/o Ref.	48.91	65.93	84.6	81.34	81.61	72.49
	Ours w Ref.	**66.83**	**79.64**	**95.11**	92.18	97.92	**86.34**

Table 5. Performance on the GenMOP dataset. "Ours w/o Ref." means not using the pose refiner in the iG-6DoF estimator.

ities on unseen objects compared to DeepIM and Gen6D.

Ablation study on data augmentation. To demonstrate the impact of our data augmentation module, we selected B0, B1, and B2 as baselines and compared the quantitative results before and after removing the data augmentation module. As shown in Table 4 (D1, D2 and D3), the results indicate that our data augmentation module significantly improves overall performance.

Ablation study on number of reference images. Finally, we evaluated the impact of the number of reference images on our method's performance by setting the reference image count to 16, 32, 64, and 128 in Table 4(E0 to E3). As expected, the model's performance improves with an increasing number of reference images, aligning with our intuition. Thanks to the effectiveness of our data augmentation module, even with a smaller number of reference images, our method still achieves commendable results.

Runtime. iG-6DoF processes each image (resolution 480×640) in approximately 0.5 seconds on a desktop equipped with an Intel Xeon Silver 4310 CPU @ 2.10GHz and an Nvidia GeForce RTX 3090 GPU. This includes 0.12 seconds for object detection, 0.01 seconds for initial pose estimation, and 0.4 seconds for pose refinement.

5. Conclusion

In this paper, we introduced a novel end-to-end pose estimation method based on 3D Gaussian Splatting without the object's CAD model. Our method demonstrates strong generalization capabilities, effectively estimating the pose of unseen objects with only a set of reference images. Unlike previous work, which always relies on precise 3D models, additional supervisory data, and struggles with significant object translations or rotations, our method is robust and versatile. Our method consistently achieves state-of-the-art performance, as evidenced by results on the widely used benchmarks. Furthermore, we conducted experiments on our captured scenes, validating our method's generalization potential and efficacy in diverse scenarios.

6. Acknowledgments

This work is partially supported by National Nature Science Foundation of China (No.62372336 and No.62172309).

References

[1] Adel Ahmadyan and Liangkai Zhang. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. 6

[2] Phil Ammirato, Cheng-Yang Fu, Mykhailo Shvets, Jana Kosecka, and Alexander C Berg. Target driven instance detection. *arXiv preprint arXiv:1803.04610*, 2018. 4

[3] Apple. Arkit. https://developer.apple.com/augmentedreality/, 2017. 3

[4] Gil Avraham, Julian Straub, Tianwei Shen, Tsun-Yi Yang, Hugo Germain, Chris Sweeney, Vasileios Balntas, David Novotny, Daniel DeTone, and Richard Newcombe. Nerfels: renderable neural codes for improved camera pose estimation. In *CVPR*, pages 5061–5070, 2022. 3

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 3

[6] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE TPAMI*, 1992. 1

[7] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, pages 4160–4169, 2023. 3

[8] Dingding Cai and Janne Heikkilä. Gs-pose: Cascaded framework for generalizable segmentation-based 6d object pose estimation. *arXiv preprint arXiv:2403.10683*, 2024. 1, 6, 7

[9] Ming Cai and Ian Reid. Reconstruct locally, localize globally: A model free method for object pose estimation. In *CVPR*, 2020. 6

[10] Tuo Cao and Fei Luo. Dgecn: A depth-guided edge convolutional network for end-to-end 6d pose estimation. In *CVPR*, 2022. 1

[11] Tuo Cao, Wenxiao Zhang, Yanping Fu, Shengjie Zheng, Fei Luo, and Chunxia Xiao. Dgecn++: A depth-guided edge convolutional network for end-to-end 6d pose estimation via attention mechanism. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):4214–4228, 2023. 1

[12] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Freeze: Training-free zero-shot 6d pose estimation with geometric and vision foundation models. In *European Conference on Computer Vision*, pages 414–431. Springer, 2024. 2

[13] Pedro Castro and Tae-Kyun Kim. Posematcher: One-shot 6d object pose estimation by deep feature matching. In *ICCVW*, 2023. 1, 2

[14] Jianqiu Chen and Mingshan Sun. Zeropose: Cad-model-based zero-shot pose estimation. *arXiv preprint arXiv:2305.17934*, 2023. 1, 2

[15] Kai Chen and Qi Dou. Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In *ICCV*, 2021. 1

[16] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Garf: gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *arXiv e-prints*, pages arXiv–2204, 2022. 3

[17] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 6

[18] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 6

[19] Zhaoxin Fan and Zhenbo Song. Object level depth reconstruction for category level 6d object pose estimation from monocular rgb image. In *ECCV*, 2022. 1

[20] Martin A. Fischler and Robert C. Bolles. Random sample consensus. *COMMUN ACM*, 1981. 1

[21] Minghao Gou and Haolin Pan. Unseen object 6d pose estimation: A benchmark and baselines. *arXiv preprint arXiv:2206.11808*, 2022. 1

[22] Frederik Hagelskjær and Rasmus Laurvig Haugaard. Keymatchnet: Zero-shot pose estimation in 3d point clouds by generalized keypoint matching. *arXiv preprint arXiv:2303.16102*, 2023. 1

[23] Huasong Han, Kaixuan Zhou, Xiaoxiao Long, Yusen Wang, and Chunxia Xiao. Ggs: Generalizable gaussian splatting for lane switching in autonomous driving. *arXiv preprint arXiv:2409.02382*, 2024. 3

[24] Kaiming He and Georgia Gkioxari. Mask r-cnn. In *ICCV*, 2017. 2

[25] Xingyi He and Jiaming Sun. Onepose++: Keypoint-free one-shot object pose estimation without cad models. In *NeurIPS*, 2022. 1, 2, 6

[26] Yisheng He and Wei Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *CVPR*, 2020. 1

[27] Yisheng He and Yao Wang. Fs6d: Few-shot 6d pose estimation of novel objects. In *CVPR*, 2022. 1, 2

[28] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM TOG*, 37(6):1–15, 2018. 3

[29] Stefan Hinterstoisser and Vincent Lepetit. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, 2012. 6, 7

[30] Tomas Hodan and Martin Sundermeyer. Bop challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects. *arXiv preprint arXiv:2403.09799*, 2024. 6

[31] Lin Huang and Tomas Hodan. Neural correspondence field for object pose estimation. In *ECCV*, 2022. 3

[32] Corsetti Jaime and Boscaini Davide. Open-vocabulary object 6d pose estimation. In *CVPR*, 2024. 2

[33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time

radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 5

[34] Alexander Kirillov and Eric Mintun. Segment anything. In *ICCV*, 2023. 2

[35] Yann Labbé and Justin Carpentier. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *ECCV*, 2020. 5

[36] Yann Labbé and Lucas Manuelli. Megapose: 6d pose estimation of novel objects via render & compare. In *CoRL*, 2022. 1, 3, 5, 6

[37] JongMin Lee and Yohann Cabon. Mfos: Model-free & one-shot object pose estimation. In *AAAI*, 2024. 6

[38] Fu Li and Shishir Reddy Vutukur. Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation. In *ICCV*, 2023. 3

[39] Yi Li and Gu Wang. Deepim: Deep iterative matching for 6d pose estimation. In *ECCV*, 2018. 5, 7

[40] Jiehong Lin and Lihua Liu. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *CVPR*, 2024. 2

[41] Jian Liu, Wei Sun, Hui Yang, Zhiwen Zeng, Chongpei Liu, Jin Zheng, Xingyu Liu, Hossein Rahmani, Nicu Sebe, and Ajmal Mian. Deep learning-based object pose estimation: A comprehensive survey. *arXiv preprint arXiv:2405.07801*, 2024. 1

[42] Yuan Liu and Yilin Wen. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *ECCV*, 2022. 1, 2, 4, 6, 7

[43] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Gift: Learning transformation-invariant dense visual descriptors via group cnns. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[44] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3, 4

[45] Maxime Oquab and Timothée Darcet. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[46] Evin Pınar Örnek and Yann Labbé. Foundpose: Unseen object pose estimation with foundation features. *arXiv preprint arXiv:2311.18809*, 2023. 2

[47] Panwang Pan and Zhiwen Fan. Learning to estimate 6dof pose from limited data: A few-shot, generalizable approach using rgb images. In *3DV*, 2024. 6, 7

[48] Sida Peng and Yuan Liu. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, 2019. 1, 6

[49] Giorgia Pitteri and Aurélie Bugeau. 3d object detection and pose estimation of unseen objects in color images with local surface embeddings. In *ACCV*, 2020. 2

[50] Giorgia Pitteri and Slobodan Ilic. Cornet: Generic 3d corners for 6d pose estimation of new objects without retraining. In *ICCVW*, 2019. 1, 2

[51] Charles R Qi and Hao Su. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 4

[52] Jiongming Qin, Fei Luo, Tuo Cao, Wenju Xu, and Chunxia Xiao. Hs-surf: A novel high-frequency surface shell radiance field to improve large-scale scene rendering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6006–6014, 2024. 3

[53] Zheng Qin and Hao Yu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, 2022. 2

[54] Paul-Edouard Sarlin and Daniel DeTone. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[55] Ivan Shugurov and Fu Li. Osop: A multi-stage one shot object pose estimation framework. In *CVPR*, 2022. 6, 7

[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[57] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *ICCV*, pages 6229–6238, 2021. 3

[58] Jiaming Sun and Zehong Shen. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021. 2, 6

[59] Jiaming Sun and Zihao Wang. Onepose: One-shot object pose estimation without cad models. In *CVPR*, 2022. 1, 2, 3, 6

[60] Yuan Sun, Xuan Wang, Yunfan Zhang, Jie Zhang, Caigui Jiang, Yu Guo, and Fei Wang. icomma: Inverting 3d gaussians splatting for camera pose estimation via comparing and matching. *arXiv preprint arXiv:2312.09031*, 2023. 3

[61] Chen Wang and Danfei Xu. Densefusion: 6d object pose estimation by iterative dense fusion. In *CVPR*, 2019. 1

[62] Gu Wang and Fabian Manhardt. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, 2021. 1

[63] He Wang and Srinath Sridhar. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 1

[64] Haiping Wang, Yuan Liu, Qingyong Hu, Bing Wang, Jianguo Chen, Zhen Dong, Yulan Guo, Wenping Wang, and Bisheng Yang. Roreg: Pairwise point cloud registration with oriented descriptors and local rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10376–10393, 2023. 4, 5

[65] Yusen Wang, Zongcheng Li, Yu Jiang, Kaixuan Zhou, Tuo Cao, Yanping Fu, and Chunxia Xiao. Neuralroom: Geometry-constrained neural implicit surfaces for indoor scene reconstruction. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 3

[66] Yusen Wang, Kaixuan Zhou, Wenxiao Zhang, and Chunxia Xiao. Megasurf: Scalable large scene neural surface reconstruction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6414–6423, 2024. 3

[67] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402. Ieee, 2003. 6

[68] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[69] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024. 3

[70] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7, 8

[71] Yu Xiang and Tanner Schmidt. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1

[72] Lin Yen-Chen and Pete Florence. inerf: Inverting neural radiance fields for pose estimation. In *IROS*, 2021. 3

[73] Ruida Zhang and Yan Di. Ssp-pose: Symmetry-aware shape prior deformation for direct category-level object pose estimation. In *IROS*, 2022. 1

[74] Chen Zhao and Yinlin Hu. Locposenet: Robust location prior for unseen object pose estimation. In *3DV*, 2024. 6, 7

[75] Heng Zhao and Shenxing Wei. Learning symmetry-aware geometry correspondences for 6d object pose estimation. In *ICCV*, 2023. 2

[76] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *CVPR*, pages 12786–12796, 2022. 3

[77] Lu Zou and Zhangjin Huang. 6d-vit: Category-level 6d object pose estimation via transformer-based instance representation learning. *IEEE TIP*, 2022. 1