HiPose: Hierarchical Binary Surface Encoding and Correspondence Pruning for RGB-D 6DoF Object Pose Estimation

Yongliang Lin[1*] Yongzhi Su[2,3*] Praveen Nathan[2] Sandeep Inuganti[2,3] Yan Di[4,5]
Martin Sundermeyer[5] Fabian Manhardt[5] Didier Stricker[2,3] Jason Rambach[2,†] Yu Zhang[1,†]
[1] Zhejiang University [2] German Research Center for Artificial Intelligence (DFKI)
[3] RPTU Kaiserslautern-Landau [4] Technische Universität München [5] Google
{lyl1020; zhangyu80}@zju.edu.cn, jason.rambach@dfki.de

Abstract

In this work, we present a novel dense-correspondence method for 6DoF object pose estimation from a single RGB-D image. While many existing data-driven methods achieve impressive performance, they tend to be time-consuming due to their reliance on rendering-based refinement approaches. To circumvent this limitation, we present HiPose, which establishes 3D-3D correspondences in a coarse-to-fine manner with a hierarchical binary surface encoding. Unlike previous dense-correspondence methods, we estimate the correspondence surface by employing point-to-surface matching and iteratively constricting the surface until it becomes a correspondence point while gradually removing outliers. Extensive experiments on public benchmarks LM-O, YCB-V, and T-Less demonstrate that our method surpasses all refinement-free methods and is even on par with expensive refinement-based approaches. Crucially, our approach is computationally efficient and enables real-time critical applications with high accuracy requirements.

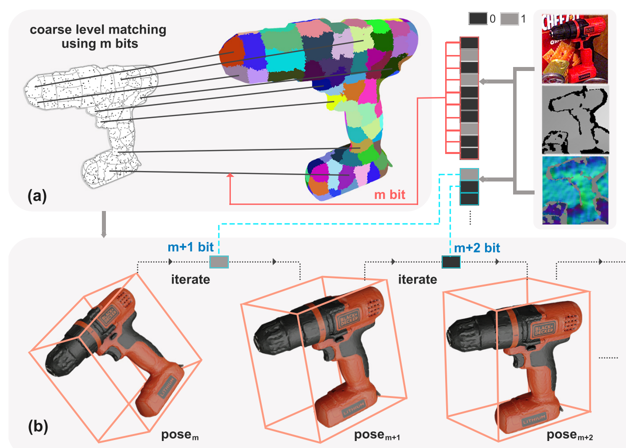

Figure 1. **Illustration of HiPose :** (a) For every point cloud with color and normals as inputs, our network outputs a binary code to establish a correspondence to a sub-surface on the object. (b) With the coarse level matching, we estimate an initial pose $pose_m$. The additional n bits are used for iterative fine-grained matching and pose estimation and gradual outlier rejection. Note that this process is render-free and RANSAC-free, ensuring fast performance of our algorithm.

1. Introduction

Estimating the 6 Degrees of Freedom (6DoF) object pose stands as a fundamental challenge in 3D computer vision. This task plays a pivotal role in numerous real-world applications, including augmented reality [45, 48, 70], robotic grasping [68, 69], and autonomous driving [39, 51]. Despite its significance, achieving accurate 6DoF pose estimation remains challenging, particularly in scenarios characterized by homogeneous object textures and heavy occlusion.

The advent of deep learning has helped to overcome those challenges. A recent branch of RGB-only works [41, 50, 67] shows promising results for handling occlusion. Despite these advancements, estimating object pose from RGB images alone remains challenging due to the inherent depth ambiguity in monocular images.

Analogous to the prediction of 2D-3D correspondence in RGB-only approaches, [5, 62] predict sparse 3D-3D correspondences. However, most methods with depth input either do not utilize RGB information [10, 11, 31] or rely on RGB images only to segment the object from the background [4, 11, 25, 34], thus discarding valuable RGB features. To preserve rich RGB information, [16, 17, 58, 74]

*Contributed equally, †Corresponding authors
Code: https://github.com/lyltc1/HiPose

proposed novel feature fusion networks to better leverage RGB and depth information but lag behind in public benchmarks such as BOP [54].

In contrast, most current state-of-the-art approaches typically obtain an initial pose using RGB-only methods and then apply a computationally expensive, often iterative, pose refinement step using depth information [35, 47]. Directly utilizing RGB-D images to estimate the initial pose promises to yield more precise and reliable object pose estimates.

In this paper, we aim to fully exploit the detailed information in RGB-D images to estimate accurate object poses without any time-consuming refinement step. Using RGB-D input, we benefit from additional information such as point-to-surface distances. Taking inspiration from the recent work ZebraPose [50], a dense 2D-3D correspondence prediction method, we introduce HiPose, a network that efficiently predicts dense 3D-3D correspondences between the input depth map and the object model. Unlike ZebraPose [50], we process the encoding in a manner that takes better advantage of its coarse-to-fine properties by iteratively removing outliers.

Instead of solving the pose using the predicted correspondences within the RANSAC framework, as commonly done in conjunction with the Kabsch algorithm [56], we propose a novel and more stable hierarchical correspondence pruning approach, eliminating the need for RANSAC. Specifically, the coarse-level prediction in the hierarchical binary code output is less error-prone, providing a robust initial pose. This coarse pose helps identifying and removing outlier matches based on point-to-surface distance. Subsequently, we apply a finer-level prediction with each iteration, refining our pose prediction and eliminating outliers at finer levels to enhance accuracy as shown in Figure 1.

Overall, our contributions can be summarized as follows:

- We present an approach for estimating object pose that fully exploits RGB-D data, focusing on 3D-3D correspondence matching through hierarchical binary surface encoding.
- We introduce a RANSAC-free hierarchical correspondence pruning approach for pose estimation through coarse-to-fine sub-surfaces based outlier filtering.
- Extensive experiments on LM-O, YCB-V and T-LESS datasets demonstrate our method's effectiveness. We achieve state-of-the-art results without any additional refinement, making our approach notably faster than alternative methods and suitable for real-time applications.

2. Related Work

We limit our in-depth discussion of related work to the instance-level pose estimation methods based on deep learning where the 3D CAD model of the target object is available during training.

2.1. RGB-only Pose Estimation

Most top-performing RGB object pose estimation methods [7, 20, 41, 50, 59, 67] attempt to establish dense 2D-3D correspondences between 2D coordinates in the RGB image and 3D coordinates on the object surface. The 6D pose is then computed by solving the Perspective-n-Point(PnP) problem [28]. Dense correspondence-based methods have been shown to outperform the keypoint-based methods [36, 40, 42, 46, 71] and holistic approaches [8, 23, 49, 63] nowadays, as also demonstrated in the BOP challenge results [54]. We draw inspiration from ZebraPose [50], a dense correspondence-based method that employs a coarse-to-fine surface encoding to represent correspondences. This approach has demonstrated significant improvements in accuracy, motivating our own idea. Overall, RGB-only methods are still limited in performance due to the absence of geometric information.

2.2. Depth-only and RGB-D Pose Estimation

The development of point cloud processing networks [2, 21, 30, 43] boosted pose estimation methods that exclusively used 3D measurements [1, 6, 60, 61, 65, 66]. These methods have demonstrated excellent generalization. However, discarding RGB appearance severely limits the performance of these approaches, due to pose ambiguities and exclusion of color features.

RGB-D methods attempt to fuse the information of the RGB and Depth modalities. [26, 29, 33] treat the depth information as an extra channel of RGB images, which are then fed to a CNN-based network. A more effective utilization of RGB and depth images is to extract features from these two modalities individually, then fuse them for pose estimation [16, 17, 44, 57, 58, 64, 72, 74]. Such approaches benefit from visual information and geometric information and show higher accuracy [54]. FFB6D [17] designed bidirectional fusion modules to enhance representation of appearance and geometry features. Recently, [74] proposed a transformer-based fusion network based on FFB6D.

2.3. Pose Refinement with Depth Information

An additional pose refinement stage, often in an iterative manner, improves the result significantly. The Iterative Closest Point algorithm (ICP) is typically used as a refinement strategy, making use of depth information to align the estimated object point cloud to the image [52, 53, 63]. PFA [22] proposes a non-iterative pose refinement strategy by predicting a dense correspondence field between the rendered and real images. CIR [35] iteratively refines both pose and dense correspondence together using a novel differentiable solver layer under a rendering and comparison strategy. However, the rendering is time-consuming.

3. Method: HiPose

This section provides a detailed description of the proposed model-based method for 6D object pose estimation.

Inspired by the successful application of binary codes in the RGB-only setting, we extend the method to encode object surfaces in a coarse-to-fine way in the RGB-D setting. Our approach consists of a hierarchical binary surface encoding which is fed into a coarse-to-fine pose solver. The solver achieves rapid pose estimation through several iterations of surface partitioning and outlier rejection without RANSAC or the need for rendering.

3.1. Problem Definition and Notation

For each target object in an RGB-D image (RGB image + 2D depth map), our goal is to estimate the transformation between the predefined object coordinate system and the camera coordinate system. This transformation consists of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$.

We are given an object $\mathbf{O}$ with a 3D Scan or CAD model mesh, denoted $\mathcal{M}$, consisting of N 3D vertices $\mathbf{v}_i \in \mathbb{R}^3$, with i being the vertex index. A binary encoding of the N vertices, is a binary code $\mathbf{c}_i$ of d bits that uniquely corresponds to a vertex $\mathbf{v}_i$. We preprocess the object mesh by upsampling so that $N = 2^d$.

ZebraPose [50] constructs this binary encoding iteratively, by splitting the mesh into parts of equal amount of vertices at each step, and assigning a bit to each group. In the iteration $it, it \in \{0, 1, \ldots, d-1\}$ of the surface partition, we have 2^{it} separate sub-surfaces. Assuming a surface contains L vertices, balanced k-means is used for the partitioning, resulting in two sub-surfaces containing $\lfloor L/2 \rfloor$ and $L - \lfloor L/2 \rfloor$ vertices respectively.

This procedure creates a hierarchical encoding, meaning that all vertices whose first x bits are equal, belong to the same sub-surfaces of the object mesh surface until the x-th partition. Expressed differently, a binary code $\mathbf{c}$ describes a manifold of coarse-to-fine object surfaces $\mathcal{S}_k, k = 0, \cdots, d$ where $\mathcal{S}_0$ is the full object mesh and $\mathcal{S}_d$ is a single vertex of the mesh.

3.2. Point-to-Surface Correspondences

Prior work [50] used a binary encoding of surfaces as described in Section 3.1 for pose estimation from RGB images. This was done by training a neural network to estimate a binary code of d bits for every pixel $\mathbf{p}_{u,v}$ within a detected object bounding box. This can then be used to establish 2D-3D correspondences between pixels and encoded vertices $\mathbf{v}_i$ of the object model. These correspondences are presented to a Perspective-n-Point (PnP) solver (e.g. RANSAC+EPnP [28]) to estimate the object pose. Results show that this encoded surface representation is well suited for neural network training [50] since it allows to progressively learn finer details.

However, such a method: (1) Is not designed to make use of depth map information, apart from the final pose refinement stage, (2) does not take explicit advantage of the hierarchical nature of the encoded surface prediction, and (3) does not exploit the inherent confidence in predicted surface codes - instead, the continuous binary code estimates in the range $[0, 1]$ are quantized to discrete bit values which discards all confidence information.

In contrast, our method-HiPose is designed to take a single RGB-D image as input and extract features from both modalities to predict 3D-3D correspondences. Our network receives as input a cropped Region of Interest (RoI) from a detected object, both, from the RGB image and the depth map as seen in Figure 2. The inputs are processed by two branches of a bidirectional fusion network as in FFB6D [17]. Details on the network architecture and implementation are provided in Section 4.1. The pixels of the depth image are converted to a point cloud. For each 3D point $\mathbf{P}$, our network is trained to predict a binary code $\hat{\mathbf{c}}$. This code represents a 3D-3D correspondence between the point cloud and the object model. A solution for the pose can be obtained by passing these correspondences to the Kabsch pose estimation algorithm [56].

This approach is in line with similar correspondence methods. However, the hierarchical nature of the encoding presents an opportunity for coarse-to-fine processing that has not been exploited yet. This is where our hierarchical point-to-surface approach is applied. As discussed in Section 3.1, a binary encoding represents a manifold of object surfaces depending on how many bits of the encoding are considered. Instead of directly processing the full encoding of d bits, which includes high uncertainty for the last bits, we propose to split the binary encoding into two groups: The first m and the last n bits ($d = m + n$). Unlike in [50], we utilize the $(m+1)_{th}$ until the d_{th} bits in an iterative manner, as explained in the following.

The first m bits of the encoding yield a 3D point-to-surface correspondence from a point in the point cloud to a surface segment $\mathcal{S}_m$ on the object model. From the surface $\mathcal{S}_m$, a centroid point can be computed, and used as a 3D point on the model to create a 3D-3D correspondence. A coarse pose estimate $\hat{\mathbf{R}}_0, \hat{\mathbf{t}}_0$ can then be obtained from these correspondences. Crucially, the coarse pose estimate is used for outlier pruning as described in Section 3.4. We iteratively repeat this process for bits $m+1$ until $m+n$, in a coarse-to-fine manner, with the surfaces of interest $\mathcal{S}_{m+1}$ until $\mathcal{S}_{m+n}$ decreasing in size at each iteration. Simultaneously, the pose estimates at each iteration have gradually higher accuracy and can be used for finer outlier removal.

3.3. Hierarchical Binary Code Decoding

Existing methods such as ZebraPose [50], use estimated binary codes in a straightforward manner. The continuous es-

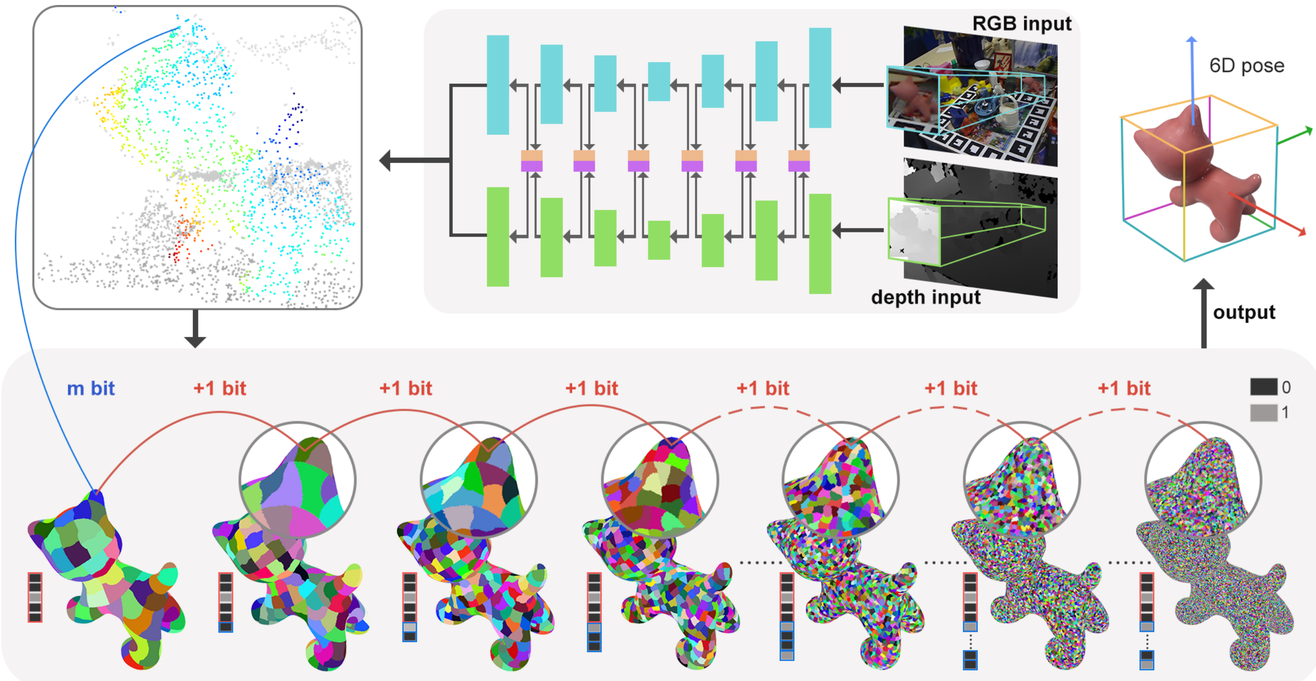

Figure 2. **Overview :** Our framework uses an RGB-D image crop as input and predicts an $m + n$ bits binary code using a full-flow bidirectional fusion network for every point cloud patch on the target object. The first m bit codes point to a relatively coarse surface (blue line), while the final n bit codes are used n times as indicators to perform hierarchical surface partitioning (red lines). Through the iterative process of identifying fine-grained point-to-surface correspondences, the algorithm finally yields an accurately estimated pose. The colored patches on the model represent different surface partitions.

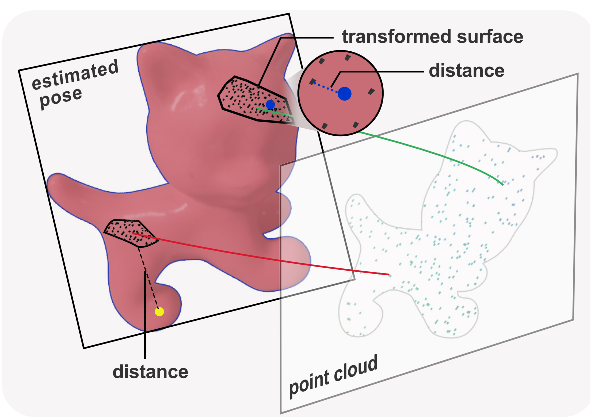

Figure 3. **Correspondence Pruning.** The green line demonstrates an example where the point cloud lies in the transformed surface under the estimated pose. The distance between the point in the point cloud and the transformed surface is represented by the blue dashed line. The red line demonstrates another case where the transformed surface is far away from the point in the point cloud (yellow point). Consequently, the correspondence depicted by the red line will be removed in the next iteration.

timated code from the neural network is transformed into a binary code by quantizing the values from the range $[0, 1]$ to bit values. Then, the estimated binarized code $\hat{\mathbf{c}}$ has direct correspondence to a vertex code $\mathbf{c}_i$. However, this method discards valuable confidence information that is inherent in the predicted code and makes the process highly reliant on the performance of the RANSAC-PnP solver.

Denoting the direct (non-binarized) prediction output vector as $\bar{\hat{\mathbf{c}}}$ and its quantized version as $\hat{\mathbf{c}}$, we propose to compute a bit correctness probability/confidence vector $\mathbf{p}_c \in \mathbb{R}^{d\times 1}_{[0,1]}$ as

$$\mathbf{p}_c = \mathbf{1}_{d\times 1} - |\bar{\hat{\mathbf{c}}} - \hat{\mathbf{c}}|. \tag{1}$$

HiPose introduces a method to leverage this probability information, enabling superior results to be achieved with several rendering-free iterations, thereby improving algorithmic efficiency. Our binary code decoding consists of initial surface selection and sub-surface partitioning.

Initial surface selection. The blue line in Figure 2 indicates the process of initial surface selection. As the number of iterations increases, each surface is further divided and the corresponding bits become more difficult to learn for the network. Therefore, it is essential to select an appropriate starting point $\mathcal{S}_m$ from the surface manifold where the pose estimation iterations should begin, or, equivalently select the value m, where the binary encoding will be split.

The initial surface selection for every correspondence is based on the bit correctness probability vector $\mathbf{p}_c$. The last bit j, for which $\mathbf{p}_c^j$ is higher than the probability threshold, is called the trust bit. We set $m_{default}$ as the minimum value of m which limits the maximum initial size of the sub-surfaces and begin our pose estimation iterations at $m = max(j, m_{default})$. There are two advantages of setting $m_{default}$: Ensuring a certain degree of accuracy in the initial pose estimation and reducing computational complexity.

Sub-surface Partitioning. As shown in Figure 2, the red lines indicate the process of sub-surface partitioning. As the surfaces reduce in size between iterations, correspondences become more challenging to learn and therefore less reliable. Consequently, during each iteration, prior to performing pose estimation on a subdivided surface, we employ our hierarchical correspondence pruning process, which efficiently eliminates outliers within a few iterations, as described in detail in the next Section 3.4. We perform $d-m$ iterations of hierarchical correspondence pruning and a sub-surface $\mathcal{S}_m$ encoded by m bit can only be partitioned $d - m$ times. If the trust bit j of a prediction is larger than $m_{default}$, this sub-surface $\mathcal{S}_j$ will retain its size in the former $j - m_{default}$ iterations of correspondence pruning. In this way, a prediction with a higher trust bit prioritizes matching a finer correspondence, leading to a more reliable and precise pose in each iteration. For simplicity, we assume that the trust bit j equals $m_{default}$ in Section 3.4, thereby not skipping the sub-surface partitioning.

3.4. Hierarchical Correspondence Pruning

Through the point-to-surface matching process, starting from bit m of the estimated code, we can compute the centroid $\mathbf{g}_m$ of the corresponding surface $\mathcal{S}_m$ and all corresponding vertices $\mathbf{v}_i$. The 3D coordinate of $\mathbf{g}_m$ is the average of the M vertex coordinates, $\mathbf{g}_m = \frac{1}{M}\sum_{i=1}^{M}\mathbf{v}_i$.

The sub-surface partitioning and pose estimation is repeated n times (from bit $m+1$ to bit d). In the it_{th} iteration, $it \in [0, 1, ..., n-1]$, we compute the centroid $\mathbf{g}_{it}$ of the corresponding sub-surface for every masked point $\mathbf{P}$ in our point cloud and estimate a pose $[\hat{\mathbf{R}}_{it}|\hat{\mathbf{t}}_{it}]$ in this step through a Kabsch solver. With this estimated pose, we select inliers using the distance calculated below.

In the $(it{+}1)_{\text{th}}$ iteration, we calculate the distance for every correspondence between point $\mathbf{P}$ and transformed sub-surface $\mathcal{S}'_{m+it}$ under pose $[\hat{\mathbf{R}}_{it}|\hat{\mathbf{t}}_{it}]$, which will be used as the threshold to distinguish inliers and outliers. Figure 3 is a visual representation of this process. The minimum distance between a point $\mathbf{P}$ in the point cloud and transformed surface $\mathcal{S}'_{m+it}$ is defined as:

$$l = \min_{\mathbf{v}_i} ||\hat{\mathbf{R}}_{it}\mathbf{v}_i + \hat{\mathbf{t}}_{it} - \mathbf{P}|| \tag{2}$$

The correspondences are distinguished as inliers and outliers based on the median of the distance l. The different options for distinguishing inliers and outliers are compared in an ablation study, Section 4. Formally, in the general case the pose at iteration it is solved by Kabsch $\mathcal{K}$ with an input set of point-to-surface centroid 3D-3D correspondences as:

$$[\hat{\mathbf{R}}_{it}|\hat{\mathbf{t}}_{it}] = \mathcal{K}(\{\mathbf{P}_k, \mathbf{g}^k_{m+it}\}, \mathbf{P}_k \in \text{inliers}_{it} \tag{3}$$

After completing the n iterations of surface partitioning, all point-to-surface correspondences have converged to point-to-point correspondences. Finally, we perform one round of the Kabsch algorithm with all the point-to-point correspondences that were never recognized as outliers in the hierarchical correspondence pruning process to generate the final estimated pose, $[\hat{\mathbf{R}}_n|\hat{\mathbf{t}}_n]$.

4. Experiments

In this section, we start with the implementation, datasets and evaluation metrics. Next, we show ablation studies of our method with different 3D-3D correspondence solvers. Finally, we present the experimental results of our method and compare to recent pose estimation methods from the literature and from the BOP Challenge.

4.1. Implementation Details.

Our approach can be easily integrated into a variety of existing RGB-D networks. This paper utilizes the full flow bidirectional fuse network from FFB6D [17] as a baseline and applies some modifications to create the HiPose network.

Similarly to FFB6D, our HiPose network has two branches to deal with images and point clouds, respectively. We feed the HiPose network with 1) a zoomed-in Region of Interest (RoI) image, and 2) a point cloud generated by uniform sampling of a fixed number of pixels from the RoI depth map.

Modifications are also made to the output layers. We replace three output heads with a single head containing the visibility mask and a binary encoding of length d for every randomly selected point. We use L1 loss for both the visible mask and binary encoding. Our training loss is defined as

$$Loss = L_{mask} + \alpha L_{code} \tag{4}$$

where α is a weight factor between the mask and the binary encoding losses, set to $\alpha = 3$ throughout all experiments. Note that for the binary encoding prediction, we only calculate a loss for the points within the predicted visible mask.

For the network backbone, we use the recent ConvNext [38] architecture which builds up on ResNet [15]. ConvNext shows on-par performance to Vision Transformer [9] while retaining the efficiency and simplicity of ResNet.

We train our network for $380K$ iterations with a batch size of 32. We use the Adam [24] optimizer with a fixed

A0	Kabsch	86.65
A1	RANSAC+ Kabsch	89.12
A2	Our Hierarchical Pruning	**89.62**
B0	Trust Bit Threshold 0.08	89.55
B1	Trust Bit Threshold 0.06	89.61
B2	Trust Bit Threshold 0.04	89.59
C0	Inlier Threshold: median $\rightarrow$ mean	89.47
D0	ConvNext-B [38] $\rightarrow$ ResNet34 [15]	88.64
E0	ZebraPose(pbr)+RANSAC Kabsch	87.0
E1	ConcatRGBD+RANSAC Kabsch	84.5

Table 1. **Ablation study on LM-O [3].** We conduct several ablation studies, comparing our proposed method to a RANSAC-based approach and exploring the impact of hyperparameters on the results. The results are presented in terms of average recall of ADD(-S) in %.

learning rate of $1e-4$. During training, we employ RGB image augmentations used in ZebraPose [50] and depth augmentations from MegaPose [27] for the depth maps.

4.2. Datasets

We conduct our experiments on the LM-O [3], YCB-V [63] and T-LESS [18] datasets. These datasets collectively encompass a wide range of scenarios, including instances of heavy occlusion, texture deficiency and symmetrical objects. Since annotating real data for object pose can be very time-consuming, we utilize publicly available synthetic physically-based rendered (PBR) images provided by the BOP challenge [54] to demonstrate that our network can be effectively trained using only synthetic data.

4.3. Metrics

For the LM-O dataset, we report the ADD(-S) metric which is the most common metric for 6DoF pose among contemporary works. ADD calculates the percentage of object vertices that fall under a distance threshold (object size dependent) when projected to the camera coordinates using the estimated pose vs. using the ground truth pose. In the case of a symmetric object, ADD(-S) differs in that it matches the closest model points (taking symmetry into account) instead of the exact same model points. For the YCB-V dataset, we report the area under curve (AUC) of the ADD(-S) with a maximum threshold of 10cm as described in [63]. Additionally, for both datasets, we report the BOP score metric defined by the BOP Challenge [54].

4.4. Ablation Studies

In the following, we perform several ablation studies with the LM-O dataset. We summarize the results in Table 1, Table 2 and Figure 4.

Iteration Step	1	2	3	4	5	6	7	final
precision(%)	66.1	67.4	70.3	73.3	73.8	75.9	75.9	76.1

Table 2. **Outlier removal metrics.** We evaluate precision at each iteration to validate the outlier removal process.

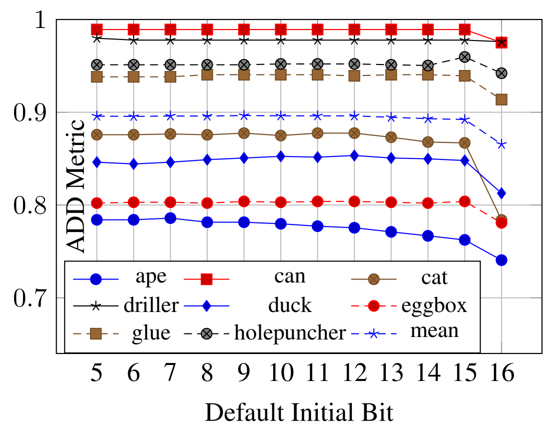

Figure 4. We conduct an ablation study on selecting the default initial bit $m_{default}$ using 8 objects from the LM-O dataset. The flat curve illustrates that our proposed design is robust and has a clear advantage to the non-hierarchical variant (16 as initial bit).

Effectiveness of Correspondence Pruning. We first focus on the effectiveness of our proposed hierarchical correspondence pruning. In the experiment (A0) in Table 1, we directly solve the object pose with the Kabsch Algorithm. The promising results highlight the effectiveness of the binary encoding. However, compared to our other experiments reveals that the predicted correspondences from the network are still noisy and contain outliers.

Comparing A1 with A0, the most common method for identifying outliers using RANSAC framework, we observed a 2.47% recall improvement. The results of A1 heavily depend on the choice of hyper-parameters, including the number of correspondences used in each iteration, the number of RANSAC iterations, and the inlier threshold in each iteration. Additionally, random seed variations can also impact the results. In this experiment, we utilized the public RANSAC and Kabsch algorithm from Open3D [75]. Note that we explored multiple parameter combinations and reported the best results among them.

In contrast to the RANSAC scheme, our hierarchical correspondence pruning provides stable results analytically, irrespective of the random seed. In experiment A2, we chose the 10_{th} bit as our initial bit and defined the confidence bit based on predicted logits higher than 0.52 or lower than 0.48. As shown in Table 1, compared to not using any outlier strategy (A0), our approach (A2) improves recall by about 3% while outperforming the best results achievable by RANSAC.

Method	RGB Input		RGB-D Input				
	GDR-Net [59]	ZebraPose [50]	PR-GCN [73]	FFB6D [17]	RCVPose [62]	DFTr [74]	**Ours**
ape	46.8	57.9	40.2	47.2	60.3	64.1	**78.0**
can	90.8	95.0	76.2	85.2	92.5	96.1	**98.9**
cat	40.5	60.6	57.0	45.7	50.2	52.2	**87.5**
driller	82.6	94.8	83.2	81.4	78.2	95.8	**97.8**
duck	46.9	64.5	30.0	53.9	52.1	72.3	**85.3**
eggbox*	54.2	70.9	68.2	70.2	**81.2**	75.3	80.3
glue*	75.8	88.7	67.0	60.1	72.1	79.3	**94.1**
holepuncher	60.1	83.0	**97.2**	85.9	75.2	86.8	95.2
mean	62.2	76.9	65.0	66.2	70.2	77.7	**89.6**

Table 3. **Comparison with State of the Art on LM-O [3].** We report the Recall of ADD(-S) in % and compare with state of the art. (*) denotes symmetric objects.

	Method	AUC of ADD-S	AUC of ADD(-S)
RGB	SO-Pose [7]	90.9	83.9
	GDR-Net [59]	91.6	84.4
	ZebraPose [50]	90.1	85.3
RGB-D	PVN3D [16]	95.5	91.8
	RCVPose [62]	96.6	95.2
	FFB6D [17]	96.6	92.7
	DFTr [74]	96.7	94.4
	Ours	**97.6**	**95.5**

Table 4. **Comparison with State of the Art on YCB-V[63].** We compare our HiPose with state of the art w.r.t AUC of ADD(-S) and AUC of ADD-S in %.

We also estimate the precision metrics of outlier removal at each iteration in Table 2. A true sample is defined when the distance between estimated coordinates and ground truth fall under a 10mm threshold. The increase in precision confirms the gradual removal of low-quality correspondences. In the following ablation studies, we demonstrate that our approach is also robust to the choice of hyper-parameters.

Influence of Default Initial Bit. Using smaller initial bits implies matching the point to a relatively coarse surface correspondence. However, our trust-bit-based initial surface selection ensures that each vertex is considered separately and corresponds to its most reliable initial bit.

As demonstrated in Figure 4, our proposed design is robust to the choice of initial bit from 5_{th} bit to 11_{th} bit. We observe some performance drops when we start from the 12_{th} bit, and a significant drop when directly using 16_{th} bit, underscoring the importance of our hierarchical correspondence pruning. By calculating the mean recall across all 8 objects, we noticed that using the 9_{th} to 11_{th} bits as the initial bit provides slightly higher results. Considering both accuracy and computational efficiency, we consistently use the 10_{th} bit as our default initial bit.

Threshold for the Trust Bit. The initial bit for each point and our inlier identification strategy is closely tied to the choice of the trust bit. By varying the threshold 0.02 used in experiment A2 (predicted logits greater than 0.52 or smaller than 0.48), we alter the trust bit threshold in experiments B0, B1, and B2. As indicated by the experimental results, when the threshold is greater than 0.08, we start to observe a small performance drop. Overall, the results remain quite stable within the threshold range of 0.02 to 0.08. This demonstrates the relative robustness of our approach to the choice of the trust bit.

Criteria used in Correspondence Pruning. The default criterion for distinguishing inliers and outliers in correspondence pruning is based on the median of distance l defined in Equation 2. We replaced the median criterion for the inlier threshold used in A2 with a mean criterion in C0, resulting in a decrease in average recall. It is comprehensible that utilizing indicators associated with the median produces superior outcomes, given the median's ability to disregard the impact of extreme values.

Effectiveness of CNN Backbone. To ensure comparability with recent research employing the transformer architecture and ConvNext [38] feature backbone, we default to using ConvNext as our image feature extraction network. Additionally, we provide results of experiment (D0) in Table 1 using ResNet as the feature backbone to offer further insights for comparison with earlier approaches. Results show that the choice of feature backbone only has a marginal effect.

Naive RGB-D baselines. Using the networks provided by ZebraPose, we back-project 2D pixels into 3D points using the depth map, followed by pose estimation using RANSAC + Kabsch for 3D-3D correspondence(E0). We also implement "Concat"(E1), a naive baseline model that

Method	Refinement	LM-O	YCB-V	T-LESS	mean	time(sec)
FFB6D-CVPR21-PBR-NoRefinement [17]	-	68.7	75.8	-	72.3*	0.19*
RCVPose 3D_SingleModel_VIVO_PBR [62]	ICP [47]	72.9	84.3	70.8	76.0	1.33
SurfEmb-PBR-RGBD [14]	custom [14]	76.0	79.9	82.8	79.6	9.04
RADet+PFA-PBR-RGBD [13]	PFA [22]	79.7	82.6	80.2	80.8	2.63
GDRNPP-PBR-RGBD-MModel [37]	∼CIR [35]	77.5	90.6	**85.2**	84.4	6.37
HiPose (ours)	-	**79.9**	**90.7**	83.3	**84.6**	**0.16**

Table 5. Comparison to leading methods of the BOP Challenge [54] that trained on synthetic PBR data only w.r.t. BOP score. (∼) denotes similar to CIR[35]. * averaged over LM-O and YCB-V only as T-LESS results are not provided for this method.

learns 2D-3D correspondence, yet solves the pose with 3D-3D correspondences. In the "Concat" baseline, we concatenate the RGB and depth channels as input for the CNN. However, the absence of a pretrained CNN model appears to make the results worse. Nonetheless, none of the baselines surpass HiPose, suggesting that direct learning of 3D-3D correspondences is more effective.

4.5. Comparison to State of the Art

In the following, we compare HiPose to the state of the art using various metrics on multiple datasets.

Results on LM-O. In Table 3, we compare our HiPose with the state-of-the-art methods on the LM-O dataset w.r.t. the ADD(-S) score. We used the 2D detection provided by CDPN [32], which is based on the FCOS [55] detector and trained only with PBR images provided by the BOP challenge. According to the results, HiPose outperforms all other methods by a large margin of 11.9% compared to the best RGB-D method DFTr and 12.7% compared to the best RGB-only method ZebraPose.

Results on YCB-V. In Table 4, we compare HiPose with the state-of-the-art methods on the YCB-V dataset w.r.t. the AUC of ADD-S and ADD(-S) score. All other methods used real and synthetic images in the training. To ensure a fair comparison, the 2D FCOS detections employed here are trained with both real and synthetic images. According to the results, HiPose again excels beyond all other methods with a significant margin (around 1%) when taking into account that scores on YCB-V are already close to saturation.

Results on the BOP Benchmark. The BOP benchmark provides a fairer ground for comparisons, offering uniform training datasets and 2D detections for all participating methods and more informative evaluation metrics [19]. We used the default detections provided for the BOP Challenge 2023.

Most methods in Table 5 rely on a time-consuming pose refinement step, while HiPose estimates accurate object pose directly without any pose refinement. HiPose surpasses the state-of-the-art on LM-O, YCB-V datasets and achieves a very comparable result on the T-LESS dataset. When considering the average recall across the three datasets, HiPose exhibits higher recall compared to all other methods.

GDRNPP [37] has the most closely aligned results with HiPose, yet HiPose is approximately **40** times faster than GDRNPP with refinement. This demonstrates that HiPose is both accurate and computationally efficient.

4.6. Runtime Analysis

The inference time mainly comprises two components: 1) object detection and 2) object pose estimation. For a fair comparison, we use the same method to calculate inference speed as GDRNPP on an RTX3090 GPU. Our average object pose estimation time across the LM-O, YCB-V, and T-LESS datasets is 0.075 seconds per image. The average 2D detection time with YOLOX [12] on those three datasets is 0.081 seconds. The fast object pose estimation time ensures the real-time applicability of our approach, especially since the costly refinement methods are not necessary.

5. Conclusion

We introduced HiPose, an RGB-D based object pose estimation method designed to fully exploit hierarchical surface encoding representations and iteratively establish robust correspondences. In contrast to existing methods, we consider the confidence of every bit prediction and gradually remove outliers. Our method is trained exclusively on synthetic images and outperforms the state-of-the-art in object pose estimation accuracy across several datasets. At the same time, HiPose is considerably faster as time-consuming pose refinements become redundant.

Acknowledgements

This work was partially supported by STI 2030-Major Projects 2021ZD0201403, in part by NSFC 62088101 Autonomous Intelligent Unmanned Systems and was partially funded by the EU Horizon Europe Framework Program under grant agreement 101058236 (HumanTech) and the German Ministry of Education and Research (BMBF) under Grant Agreements 01IW20002 (SocialWear) and 01IW20009(RACKET).

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7163–7172, 2019. 2

[2] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 2

[3] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016. 6, 7

[4] Dingding Cai, Janne Heikkilä, and Esa Rahtu. Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6803–6813, 2022. 1

[5] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, and Ales Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4233–4242, 2020. 1

[6] Zheng Dang, Lizhou Wang, Yu Guo, and Mathieu Salzmann. Learning-based point cloud registration for 6d object pose estimation in the real world. In *European Conference on Computer Vision*, pages 19–37. Springer, 2022. 2

[7] Yan Di, Fabian Manhardt, Gu Wang, Xiangyang Ji, Nassir Navab, and Federico Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021. 2, 7

[8] Thanh-Toan Do, Ming Cai, Trung Pham, and Ian Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 5

[10] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010. 1

[11] Ge Gao, Mikko Lauri, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. Cloudaae: Learning 6d object pose regression with on-line data synthesis on point clouds. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11081–11087. IEEE, 2021. 1

[12] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *ArXiv*, abs/2107.08430, 2021. 8

[13] Yang Hai, Rui Song, Jiaojiao Li, Mathieu Salzmann, and Yinlin Hu. Rigidity-aware detection for 6d object pose estimation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8927–8936, 2023. 8

[14] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6739–6748, 2021. 8

[15] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 5, 6

[16] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020. 1, 2, 7

[17] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3003–3013, 2021. 1, 2, 3, 5, 7, 8

[18] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888. IEEE, 2017. 6

[19] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018. 8

[20] Tomas Hodan, Daniel Barath, and Jiri Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11703–11712, 2020. 2

[21] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020. 2

[22] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6d object pose estimation. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 2, 8

[23] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. 2

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[25] Rebecca König and Bertram Drost. A hybrid approach for 6dof pose estimation. In *European Conference on Computer Vision*, pages 700–706. Springer, 2020. 1

[26] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[27] Yann Labb'e, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *Conference on Robot Learning*, 2022. 6

[28] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81:155–166, 2009. 2, 3

[29] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the european conference on computer vision (eccv)*, pages 254–269, 2018. 2

[30] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 2

[31] Zhujun Li and Ioannis Stamos. Depth-based 6dof object pose estimation using swin transformer. *arXiv preprint arXiv:2303.02133*, 2023. 1

[32] Zhigang Li, Gu Wang, and Xiangyang 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*, pages 7678–7687, 2019. 8

[33] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656, 2018. 2

[34] Xiao Lin, Deming Wang, Guangliang Zhou, Chengju Liu, and Qijun Chen. Transpose: 6d object pose estimation with geometry-aware transformer. *arXiv preprint arXiv:2310.16279*, 2023. 1

[35] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6728–6737, 2022. 2, 8

[36] Xingyu Liu, Rico Jonschkowski, Anelia Angelova, and Kurt Konolige. Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11602–11610, 2020. 2

[37] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. Gdrnpp. https://github.com/shanice-l/gdrnpp_bop2022, 2022. 8

[38] Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022. 5, 6, 7

[39] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019. 1

[40] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 2

[41] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019. 1, 2

[42] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 2

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[44] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[45] Jason Rambach, Alain Pagani, and Didier Stricker. [poster] augmented things: Enhancing ar applications leveraging the internet of things and universal 3d object tracking. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 103–108. IEEE, 2017. 1

[46] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International journal of computer vision*, 66: 231–259, 2006. 2

[47] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 2, 8

[48] Yongzhi Su, Jason Rambach, Nareg Minaskan, Paul Lesur, Alain Pagani, and Didier Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 222–227. IEEE, 2019. 1

[49] Yongzhi Su, Jason Rambach, Alain Pagani, and Didier Stricker. Synpo-net—accurate and fast cnn-based 6dof object pose estimation using synthetic training. *Sensors*, 21(1): 300, 2021. 2

[50] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022. 1, 2, 3, 6, 7

[51] Yongzhi Su, Yan Di, Guangyao Zhai, Fabian Manhardt, Jason Rambach, Benjamin Busam, Didier Stricker, and Fed-

erico Tombari. Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection. *IEEE Robotics and Automation Letters*, 8(3):1327–1334, 2023. 1

[52] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the european conference on computer vision (ECCV)*, pages 699–715, 2018. 2

[53] Martin Sundermeyer, Maximilian Durner, En Yen Puang, Zoltan-Csaba Marton, Narunas Vaskevicius, Kai O Arras, and Rudolph Triebel. Multi-path learning for object pose estimation across domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13916–13925, 2020. 2

[54] Martin Sundermeyer, Tomás Hodan, Yann Labbé, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Juan E. Sala Matas. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2785–2794, 2023. 2, 6, 8

[55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635, 2019. 8

[56] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04): 376–380, 1991. 2, 3

[57] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14540–14549, 2020. 2

[58] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019. 1, 2

[59] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021. 2, 7

[60] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019. 2

[61] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *Advances in neural information processing systems*, 32, 2019. 2

[62] Yangzheng Wu, Mohsen Zand, Ali Etemad, and Michael Greenspan. Vote from the center: 6 dof pose estimation in rgb-d images by radial keypoint voting. In *European Conference on Computer Vision*, pages 335–352. Springer, 2022. 1, 7, 8

[63] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 2, 6, 7

[64] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018. 2

[65] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020. 2

[66] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 733–750. Springer, 2020. 2

[67] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019. 1, 2

[68] Guangyao Zhai, Xiaoni Cai, Dianye Huang, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Sg-bot: Object rearrangement via coarse-to-fine robotic imagination on scene graphs. *arXiv preprint arXiv:2309.12188*, 2023. 1

[69] Guangyao Zhai, Dianye Huang, Shun-Cheng Wu, HyunJun Jung, Yan Di, Fabian Manhardt, Federico Tombari, Nassir Navab, and Benjamin Busam. Monograspnet: 6-dof grasping with a single rgb image. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1708–1714. IEEE, 2023. 1

[70] Chenyangguang Zhang, Yan Di, Ruida Zhang, Guangyao Zhai, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Ddf-ho: Hand-held object reconstruction via conditional directed distance field. *arXiv preprint arXiv:2308.08231*, 2023. 1

[71] Wanqing Zhao, Shaobo Zhang, Ziyu Guan, Wei Zhao, Jinye Peng, and Jianping Fan. Learning deep network for detecting 3d object keypoints and 6d poses. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 14134–14142, 2020. 2

[72] Guangliang Zhou, Yi Yan, Deming Wang, and Qijun Chen. A novel depth and color feature fusion framework for 6d object pose estimation. *IEEE Transactions on Multimedia*, 23: 1630–1639, 2020. 2

[73] Guangyuan Zhou, Huiqun Wang, Jiaxin Chen, and Di Huang. Pr-gcn: A deep graph convolutional network with point refinement for 6d pose estimation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2773–2782, 2021. 7

[74] Jun Zhou, Kai Chen, Linlin Xu, Qi Dou, and Jing Qin. Deep fusion transformer network with weighted vector-wise keypoints voting for robust 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13967–13977, 2023. 1, 2, 7

[75] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 6