This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

PCKRF: Point Cloud Completion and Keypoint Refinement with Fusion Data for 6D Pose Estimation

Yiheng Han[1*], Irvin Haozhe Zhan[2*], Long Zeng[3], Yu-Ping Wang[4], Ran Yi[5], Minjing Yu[6], Matthieu Gaetan Lin[2], Jenny Sheng[2] and Yong-Jin Liu[†]

***Abstract*—Some robust point cloud registration approaches with controllable pose refinement magnitude, such as ICP and its variants, are commonly used to improve 6D pose estimation accuracy. However, the effectiveness of these methods gradually diminishes with the advancement of deep learning techniques and the enhancement of initial pose accuracy, primarily due to their lack of specific design for pose refinement. In this paper, we propose Point Cloud Completion and Keypoint Refinement with Fusion Data (PCKRF), a new pose refinement pipeline for 6D pose estimation. The pipeline consists of two steps. First, it completes the input point clouds via a novel pose-sensitive point completion network. The network uses both local and global features with pose information during point completion. Then, it registers the completed object point cloud with the corresponding target point cloud by our proposed Color supported Iterative KeyPoint (CIKP) method. The CIKP method introduces color information into registration and registers a point cloud around each keypoint to increase stability. The PCKRF pipeline can be integrated with existing popular 6D pose estimation methods, such as the full flow bidirectional fusion network, to further improve their pose estimation accuracy. Experiments demonstrate that our method exhibits superior stability compared to existing approaches when optimizing initial poses with relatively high precision. Notably, the results indicate that our method effectively complements most existing pose estimation techniques, leading to improved performance in most cases. Furthermore, our method achieves promising results even in challenging scenarios involving textureless and symmetrical objects. Our source code is available at https://github.com/zhanhz/KRF.**

***Index Terms*—Pose estimation, Pose refinement, Point cloud completion, Data Fusion.**

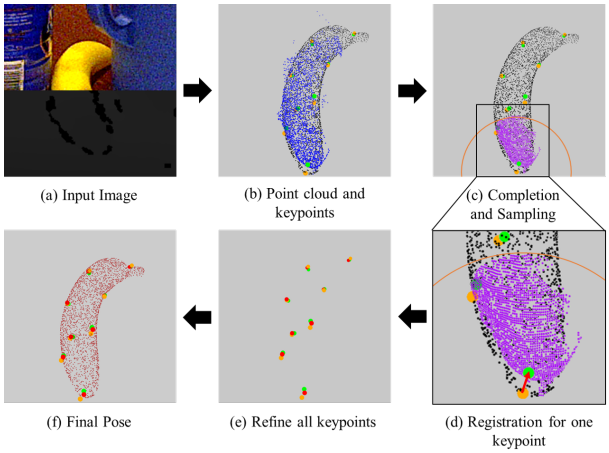


Fig. 1. **Steps of our method:** With input RGBD image (a) (the bottom half shows the depth map) and initial pose, we transform the visible point cloud (shown in blue, known object point cloud shown in black) and keypoints (shown in orange, groundtruth keypoints shown in green) to the object coordinate system (b). After completing the visible point cloud and sampling (purple points) around each keypoint within the sphere of radius r (c), we iteratively register purple and black point cloud (d) and get all refined keypoints (shown in red) (e). Then, we use the least squares fitting method to get the final pose. The model transformed by the final pose is shown in (f). It is evident that the refined keypoints are closer to groundtruth than the original keypoints.

I. Introduction

6D object pose estimation is an essential component in various applications, including robotic manipulation [1], [2], augmented reality [3], and autonomous driving [4], [5]. It has received extensive attention and has led to many research works over the past decade. Nonetheless, the task presents considerable challenges due to sensor noise, occlusion between objects, varying lighting conditions, and symmetries of objects.

Traditional methods [6], [7] attempted to extract hand-crafted features from the correspondences between known RGB images and object mesh models. However, these methods are less effective in heavy occlusion scenes or on low-texture objects. With the rapid development of deep learning, Deep Neural Networks (DNN) are now applied to the 6D object pose estimation task and demonstrate significant performance improvements. Specifically, some methods [8]–[10] use DNNs to directly regress the translation and rotation of each object. However, the non-linearity of the rotation results in poor generalization of these methods. More recently, works like

This work was partially supported by Beijing Natural Science Foundation (L222008) and Natural Science Foundation of Guangdong Province (2022A1515011234).

[1] Yiheng Han is with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. hanyiheng@bjut.edu.cn

[2]I.H. Zhan, M.G. Lin, J. Sheng and Y-J Liu are with BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing, China. {zhanhz20@mails. yh-lin21@mails. cqq22@mails. and liuyongjin@}tsinghua.edu.cn

[3]Long Zeng is with the Department of Advanced Manufacturing, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China. zenglong@sz.tsinghua.edu.cn

[4]Y-P Wang is with the Beijing Institute of Technology, Beijing, China. wyp_cs@bit.edu.cn

[5]Ran Yi is with the Shanghai Jiao Tong University, Shanghai, China. ranyi@sjtu.edu.cn

[6]Minjing Yu is with the College of Intelligence and Computing, Tianjin University, Tianjin, China. minjingyu@tju.edu.cn

*Joint first authors †Corresponding author

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

[11]–[13] utilize DNNs to detect the keypoints of each object and subsequently compute the 6D pose parameters using Perspective-n-Point (PnP) for 2D keypoints or Least Squares methods for 3D keypoints.

While DNN methods can solve the problem more rapidly, they are still unable to achieve high accuracy due to errors in segmentation or regression. To achieve higher accuracy and stability, many works have adopted pose refinement methods, of which the most common is the Iterative Closest Point (ICP) [14] algorithm. Given an estimated pose, the method tries to find the nearest neighbor of each point of the source point cloud in the target point cloud, considers it as the corresponding point, and solves for the optimal transformation iteratively. Moreover, works like [8], [15] use DNNs to extract more features for better performance. However, with the development of pose estimation networks, performance improvement of these pose refinement methods becomes less and less. The limited accuracy of existing registration methods can be attributed to their reliance on incomplete point clouds to register entire object mesh point clouds, resulting in numerous erroneous correspondences. Besides, despite the widespread use of color information in 6D estimation, its potential to enhance registration accuracy remains largely unexplored. Conventional methods have not effectively exploited the benefits of color information and are primarily designed to solve the large-scale optimization problem of point cloud registration, rather than to deal with the small-scale problem of pose refinement, resulting in an untapped area of research.

Our refinement method mainly contains two modules. Firstly, we propose a point cloud completion network to fully utilize the point cloud and RGB data. Our composite encoder of the network has two branches: the local branch fuses the RGB and point cloud information at each corresponding pixel, and the global branch extracts the feature of the whole point cloud. The decoder of the network follows [16] and employs a multistage point generation structure. Additionally, we add a keypoint detection module to the point cloud completion network during the training process to improve the sensitivity of the completed point cloud to pose accuracy, leading to better pose optimization. Secondly, to use color and point cloud data in registration and to enhance method stability, we propose a novel method named Color supported Iterative KeyPoint (CIKP), which samples the point cloud surrounding each key point and leverages both RGB and point cloud information to refine object keypoints iteratively. However, the CIKP method will make it hard to refine all key points when the point cloud is incomplete, which limits its performance. To address this issue, we introduce a combination of our completion network and the CIKP method, referred to as *Point Cloud Completion and Keypoint Refinement with Fusion* (PCKRF). This integrated approach enables the refinement of the initial pose prediction from the pose estimation network. We further conduct extensive experiments on YCB-Video [10] and Occlusion LineMOD [6] datasets to evaluate our method. The results demonstrate that our method can be effectively integrated with most existing pose estimation techniques, leading to improved performance in most cases.

Our main contribution is threefold:

- PCKRF: A pipeline that combines our completion network and CIKP method, utilizing RGBD information and keypoints throughout the refinement.
- A novel point completion network that includes a composite encoder and adds a keypoint detection module.
- A novel iterative pose refinement method CIKP that uses both RGB and point cloud information based on keypoints refinement.

Experiments demonstrate that our PCKRF exhibits superior stability compared to existing approaches when optimizing initial poses with relatively high precision. Notably, the results indicate that our method can be effectively integrated with most existing pose estimation techniques, leading to improved performance in most cases. Furthermore, our method achieves promising results even in challenging scenarios involving textureless and symmetrical objects.

II. Related Works

A. Pose Estimation

Pose estimation methods can be categorized into two types based on their optimization goal: holistic and keypoint-based methods. Holistic methods predict the 3D position and orientation of objects directly from the provided RGB and/or depth images. Traditional template-based methods construct a rigid template for an object from different viewpoints and compute the best-matched pose for the given image [17], [18]. Recently, some works utilized DNNs to directly regress or classify the 6D pose of objects. PoseCNN [10] used a multi-stage network to predict pose. It first utilized Hough Voting to determine the center location of objects and then directly regressed 3D rotation parameters. SSD-6D [19] first detected objects in the images and then classified their poses. DenseFusion [8] fused RGB and depth values at the per-pixel level, which significantly impacted 6D pose estimation methods based on RGBD images. However, the non-linearity of the rotation makes it challenging for the loss function to converge. Recently, Neural Radiance Fields have also been employed for 6D pose estimation, showcasing significant inspiration and research potential [20].

Pose estimation using only point cloud information is also called point cloud registration. Recently, the advancements in deep neural networks, particularly in three-dimensional geometry with methods like PointNet [21] and DGCNN [22], have significantly propelled the progress of deep point cloud registration. These methods are centered around the idea of utilizing deep neural networks to extract features from cross-source point clouds. These extracted features then serve as the basis for registrations or are directly used to regress transformation matrices. Techniques like SpinNet [23] aim to extract robust point descriptors through specialized neural network designs, focusing on feature learning. However, its reliance on a voxelization preprocessing step poses a challenge when dealing with cross-modality point clouds. Another approach, D3Feat [24], constructs features based on k-nearest neighbors. Nonetheless, this descriptor tends to struggle when confronted with significant density disparities. Beyond these

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

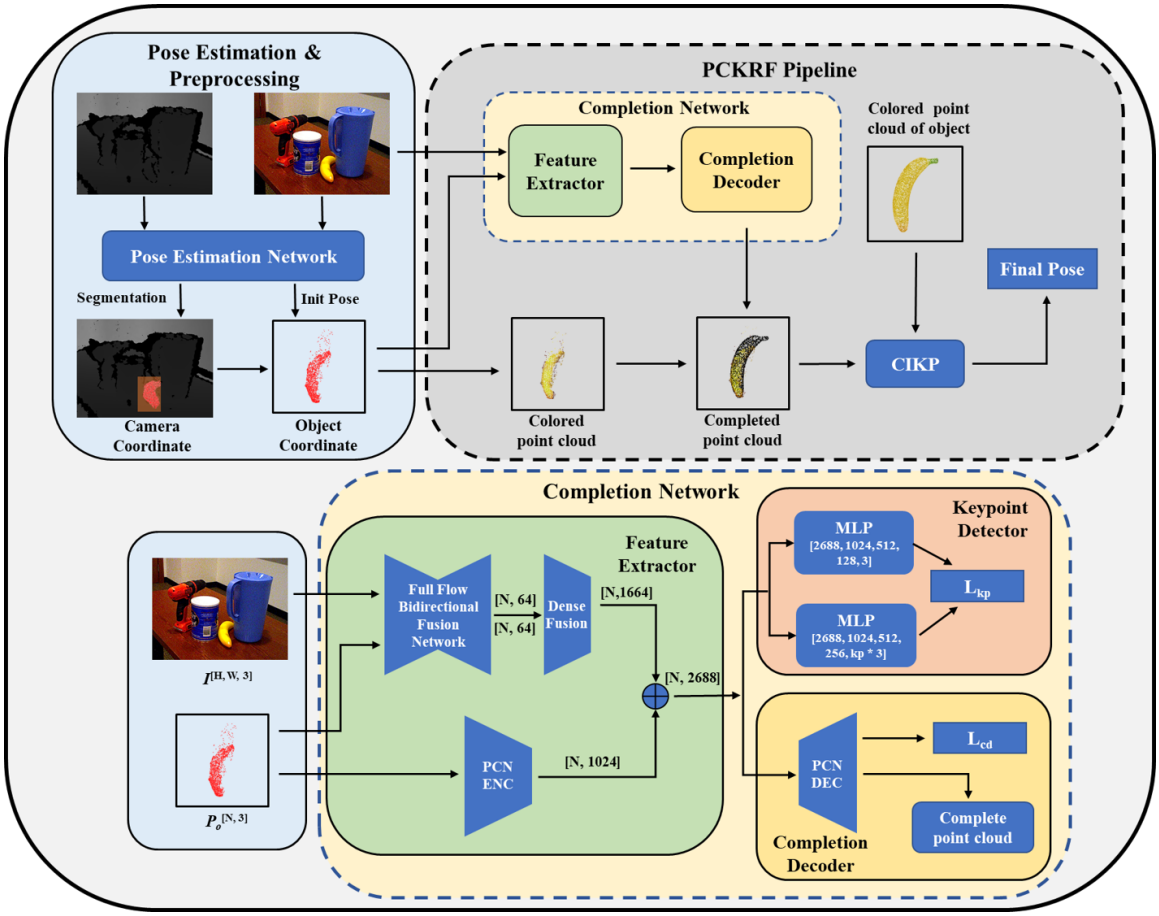


Fig. 2. The upper diagram features the PCKRF pipeline and the lower diagram is the architecture of our point cloud completion network. In the preprocessing step, we utilize the segmentation result and pose of the target object given by the pose estimation network to obtain the partial point cloud in the object coordinate system. The PCKRF pipeline first completes the partial point cloud by the point completion network and then refines the initial pose by our CIKP method. In the point cloud completion network, the Feature Extractor fuses the point cloud and RGB color at each corresponding pixel, and the Keypoint Detector predicts the offset from each point to each keypoint to improve the sensitivity of the completed point cloud to pose accuracy. The loss function of the completion network is a joint optimization of the keypoint detector Loss L_{kp} and the completion decoder Loss L_{cd}.

point descriptor-centered methodologies, several strategies emphasize feature matching. For instance, Deep Global Registration (DGR) [25] employs a UNet architecture to discern whether a point pair corresponds, reinterpreting the feature-matching challenge as a binary classification task. Alternatively, transformation learning approaches directly estimate transformations through neural networks. Feature-metric registration (FMR) [26] introduces a technique that aligns two point clouds by minimizing their feature metric projection error, offering a unique approach to point cloud registration. More recently, attempts have been made to leverage the Transformer for aggregating context between two point clouds, followed by estimating correspondences through the utilization of dual normalization [27] or some end-to-end pipelines without key points [28]. Another incremental method [29] that combined with deep-learned methods has also achieved excellent results. Moreover, the diffusion model is also applied to point cloud registration [30]. To further verify the effectiveness of our pipeline and its performance on texture-free objects, we selected a representative work [27], modified our framework, and conducted testing experiments using only the point cloud information.

Keypoint-based methods provide one way to address the above problems. Keypoint differs from superpoint [27], [31], [32], which relies on clustering or patching for point cloud registration without a prior model. Each keypoint is calculated based on the specific geometric features of the given model. YOLO-6D [33] employed the popular object detection model YOLO to predict 8 points and a center point for each bounding box of the object projected onto the 2D image. The method then computed the 6D pose using the PnP algorithm. PVNet [11] predicted a unit vector to each keypoint for each pixel, then voted the 2D location for each keypoint and calculated the final pose using the PnP algorithm. PVN3D [12] used additional depth information to detect 3D keypoints via Hough Voting and calculated the 6D pose parameters with the Least Squares method. In order to fully exploit the RGB and depth data, FFB6D [34] proposed a novel feature extraction network that applies fusion at both the encoding and decoding layers.

B. Pose Refinement

Most of the methods mentioned above apply pose refinement techniques to further improve the accuracy of their results. The most commonly used method is ICP [14], but

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

it only leverages the Euclidean distance between points. Some methods [35], [36] tried to change the optimization goal to accelerate the iterative process or improve the result. Others [37]–[39] introduced a color space into ICP, which results in faster convergence and better performance than 3D ICP methods. There are also some works [40] that pay more attention to convergence speed and propose an Anderson acceleration approach [41].

The main difference between the registration and refinement methods is whether there is an initial registration pose. Some probabilistic point cloud registration methodologies can also be used for refinement. These approaches often leverage Gaussian Mixture Models (GMMs) to represent the distribution of point clouds, framing point cloud refinement as an optimization task involving probability density functions. Notably, GMM [42] emerges as a widely adopted method due to its robustness against considerable noise and outliers, despite its relatively higher computational demands. Alternatively, FilterReg [43] innovatively reformulates the correspondence challenge in point set registration as a filtering problem through the application of Gaussian filtering. However, these methods have higher adaptability to low-precision initial poses, but often perform poorly in further optimizing the accuracy, making them less suitable for high-precision 6D pose estimation problems.

Recently, DNN-based approaches were also used to address the refinement issue. For instance, given the initial pose and 3D object model, DeepIM [15] refines the pose iteratively by matching the rendered image with the observed image. Manhardt et al. [44] proposed a novel visual loss that refines the pose by aligning object contours with the initial pose. Densefusion [8] also proposed their refinement network, which follows their main network with the original RGB features and the corresponding features of the transformed point cloud as input. Considering 6D pose estimation methods and their impressive performance improvements, existing refinement methods may not always be able to maintain stability in the entire system and achieve the highest accuracy for existing pose estimation methods. To effectively optimize high-precision pose estimation, we propose a new pose estimation framework called PCKRF (Point Cloud Completion and Keypoint Refinement with Fusion Network), which relies on keypoints and point cloud completion.

C. Point Cloud Completion

VoxelNet [45] endeavors to segment the point cloud into voxel grids and utilizes convolutional neural networks, yet this voxelization process inevitably sacrifices intricate point cloud details. Additionally, enhancing the voxel grid's resolution leads to a substantial surge in memory usage. Yuan et al. [16] introduced PCN, a sophisticated approach rooted in PointNet [21] and FoldingNet [46], which employs a coarse-to-fine methodology. However, its decoder struggles to reconstitute uncommon object geometries, such as seat backs with gaps. Consequently, subsequent methods [47], [48] have shifted their emphasis towards multi-step point cloud generation, facilitating the reconstruction of intricate details in the final point cloud. Furthermore, inspired by DGCNN [22], several researchers have ventured into graph-based techniques [49], [50], emphasizing regional geometric nuances. Recently, the transformer structure originated in the field of natural language processing and has also been employed in addressing the issue of point cloud completion, yielding notably effective results [51].

III. Our PCKRF Method

Given an RGB image and/or a depth map, the task of 6D pose estimation is to predict the rigid transformation matrix $P \in SE(3)$ for each object in the image, which comprises a rotation matrix $R \in SO(3)$ and a translation matrix $T \in \mathbb{R}^3$. It transforms the object from its own coordinate system to the camera coordinate system. Given an observed RGB image and a depth map, we first obtain the predicted pose and segmentation result of an object from the pose estimation network. Then, we utilize our PCKRF pipeline to compute a relative transformation $\Delta P \in SE(3)$ for correcting the result. Specifically, we first complete the visible point cloud by our completion network, and then use the CIKP method to compute the refined pose. The summary of the inference process is shown in Fig. 2.

A. Preprocessing

This part aims to obtain the keypoints, initial pose, and visible point cloud of each object. Firstly, we follow keypoint-based approaches [12], [34] to select K keypoints on each object surface. Subsequently, the pose estimation network takes in an RGBD image as input and outputs the segmentation results and predicts the pose of each object in the image. Additionally, we convert the given segmented depth map $D^{H_s \times W_s}$ to the point cloud of the target object $P_c^{3\times N}$, where H_s, W_s are the height and weight of the segmented depth map, respectively, and N is the number of points in the point cloud. Furthermore, we transform the point cloud from the camera coordinate system P_c to the object coordinate system P_o using the provided initial pose $R_{init} \in SO(3)$ and $T_{init} \in \mathbb{R}^3$ as follows:

$$P_o = R_{init}^{-1}(P_c - T_{init}), \tag{1}$$

Finally, we apply the MeanShift clustering algorithm [52] to eliminate outliers, which enhances the subsequent performance of the point cloud registration.

B. Completion Network for Pose Refinement

We improved the limited accuracy of existing refinement methods by incorporating a completion network as our pipeline's first module. This compensates for incomplete point clouds used to register entire object point clouds, which previously caused numerous erroneous correspondences. The architecture of our point cloud completion network is illustrated in Fig. 2, which is composed of a feature extractor, a point cloud completion decoder, and a training-only keypoint detector.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

Feature Extractor: To extract effective features from the partial point cloud and leverage the RGB and depth fusion data, we combine the Full Flow Bidirectional Fusion Network (FFB) [34], Dense Fusion module (DF) [8] and PCN encoder [16] as Feature Extractor. The Feature Extractor includes two branches: the local feature extraction branch and the global feature extraction branch. After preprocessing, we get the object point cloud with the predicted pose in its coordinate system P_o. Then, P_o and the observed RGB image $I^{H\times W\times 3}$ are fed into the FFB Fusion Network and DF module to get the local features of each point. Concretely, the FFB modules are applied to each representation learning layer as bridges for information communication between the RGB and point cloud feature extraction networks. Using the correspondence between individual points in the point cloud and pixels in the RGB image, the DF module fuses the two data sources and extracts pixel-wise dense features. Simultaneously, the PCN encoder takes the partial point cloud as input and extracts the global features by stacking two PointNet [53] feature extraction modules. Finally, local features and global features are concatenated as F.

Decoder and Keypoint Detector: In this section, we aim to generate a dense point cloud for the CIKP by implementing a keypoint detector with the completion network to enhance the sensitivity of the completed point cloud to pose accuracy. This is essential because CIKP requires high pose accuracy for the partial point clouds around each keypoint. Therefore, we incorporate a keypoint decoder and a PCN Decoder [16] to process the learned features F. The keypoint decoder predicts the keypoints and center offset, while the PCN decoder completes the visible point cloud. Given a set of keypoints $P_k^{3\times K}$ and a visible point cloud P_o, we define the translation offset from the i_{th} point $p_i \in P_o$ to the j_{th} keypoint $kp_j \in P_k$ as of_i^j. We supervise the keypoint detector module using L_{kp} loss:

$$L_{kp} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}||of_i^j - of_i^{j*}||, \tag{2}$$

where K is the number of keypoints, of_i^{j*} is the groundtruth of translation offset. Note that $K = 1$ if L_{kp} is used for predicting center point offset. We denote the loss of keypoints offset and center offset as L_{kp} and L_c, respectively.

$$L_c = \frac{1}{N}\sum_{i=1}^{N}||of_i^j - of_i^{j*}|| \tag{3}$$

We supervise the completion decoder using L_{cd} loss:

$$L_{cd} = \frac{1}{|P_o|}\sum_{x\in P_o}\min_{y\in P_d}||x-y||_2 + \frac{1}{|P_d|}\sum_{y\in P_d}\min_{x\in P_o}||x-y||_2, \tag{4}$$

where P_d is the output dense point cloud. The first term of the formula represents the sum of the minimum distance from any point x in P_o to P_d, while the second term represents the minimum distance from any point y in P_d to P_o. The overall loss L is calculated as below, where α, β, γ represent the weights of different losses:

$$L = \alpha L_{kp} + \beta L_c + \gamma L_{cd}, \tag{5}$$

By treating the positions of keypoints as an optimization objective, we aim to encourage the neural network to extract more features in their vicinity and to focus on the information surrounding the keypoints. As keypoints represent critical locations of an object, optimizing their positions can be viewed as a way to enhance their completion weights.

C. Color Supported Iterative KeyPoint

Traditional registration algorithms for pose estimation only utilize the Euclidean distance between the source point cloud and the target point cloud. These methods are less stable since they only consider the registration of the entire point cloud without color information. In contrast, pose estimation networks that fuse color and point cloud information demonstrate notable improvement in estimation accuracy. Motivated by this, we propose CIKP that iteratively refines each keypoint using the position and color information of each point.

We first define the distance between two colored points as in Eq. (6). Given two colored points p_1 and p_2, we divide them into a position component $x_1, x_2 \in \mathbb{R}^3$ and a color component $c_1, c_2 \in \mathbb{R}^3$ respectively. Note that $c_1, c_2 \in [0,1]^3$. The final distance between p_1, p_2 is their distance in the Euclidean space plus the distance in weighted color space. We stipulate that the closer the spatial distance, the higher the weight of the color distance. To mitigate the issue of excessively large weights resulting from small spatial distances, we implement a minimum weight value of 1. In addition, we introduce a threshold parameter, denoted as ϵ, which determines the threshold distance between two points. If the distance between two points is below ϵ, the weight assigned to them will remain at 1. By adjusting the value of ϵ, we can effectively control the impact of the color information on the overall computation process.

$$D = D_1 + wD_2, \text{ where } \begin{cases} D_1 = ||x_1 - x_2||_2, \\ D_2 = ||c_1 - c_2||_2, \\ w = \min(\frac{\epsilon}{D_1}, 1) \end{cases} \tag{6}$$

We summarize the process of CIKP in Alg. 1. The source point cloud P_s is a colored point cloud of objects, and the target point cloud P_t consists of visible colored point clouds and uncolored point clouds completed by our completion network, where both P_s and P_t are in the object coordinate system. We first select a keypoint, take it as the center, and collect all points in the P_t within the sphere of radius r, denoted by S_p. If $|S_p|$ is less than a threshold m_1, we keep its original state and select the next keypoint. If $|S_p|$ is more than m_2, we randomly select m_2 points in S_p. Then, for each $p_t \in S_p$, we find its closest point in P_s. If p_t is colored, we use Eq. (6) to calculate distance, otherwise we calculate their Euclidean distance directly. After that, we calculate the optimal translation transformation T_k between the two point clouds and transform the selected keypoint with T_k. After all keypoints have been transformed, the refined pose

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

is calculated using the Least Squares method. We repeat the above steps until the mean distance between corresponding points is less than a threshold τ or reaches the maximum number of iterations. Note that we decouple translation and rotation here. Only translation is taken into account when calculating the transformation of each keypoint. Rotation is only considered after all the keypoints are optimized. This approach helps to mitigate the risk of local overfitting, which will be demonstrated in subsequent experiments.

Algorithm 1: Color Supported Iterative KeyPoint

Input: source point cloud P_s, target point cloud P_t, keypoint set S_k, search radius r, threshold m_1, m_2

Output: refined rotation R and translation T

Initialize: sampled point cloud set S_p, closest point set M, refined keypoint set S_{kpr}, source point cloud in camera coordinate system P_a

$T \leftarrow [0\ 0\ 0]^T$
$R \leftarrow I^{3\times3}$
while *not converged* **do**
  $P_a \leftarrow R \times P_s + T$
  $S_{kpr} \leftarrow R \times S_k + T$
  foreach $i \leftarrow 1 : n_{kp}$ **do**
    $S_p \leftarrow \varnothing$
    $M \leftarrow \varnothing$
    foreach $p_t \in P_t$ **do**
      if $distXYZ(S_{kpr}[i], p_t) < r$ **then**
        push p_t into S_p
      end
    end
    if $|S_p| < m_1$ **then**
      continue
    end
    if $|S_p| > m_2$ **then**
      Random select m_2 points in S_p
    end
    foreach $p_t \in S_p$ **do**
      $p \leftarrow FindClosestPointInP_a(p_t)$
      push p into M
    end
    $T_k \leftarrow \arg\min_{T_k} \sum_j distXYZ(S_p[j], M[j] + T_k)$
    $S_{kpr}[i] \leftarrow S_{kpr}[i] + T_k$
  end
  $[R, T] = \arg\min_{R,T} \sum_i distXYZ(R \times S_k[i] + T, S_{kpr}[i])$
end

The poses R and T calculated in this part represent the relative transformations from the source point cloud to the target point cloud in the object coordinate system. However, we need to perform a final conversion step to obtain the final pose estimation result, which is transforming the object from the object coordinate system to the camera coordinate system. The initial rotation and translation are denoted as R_{init} and T_{init} respectively. Then, the initial pose θ_{init} and the calculated relative pose $\Delta\theta$ can be represented as follows:

$$\theta_{init} = \begin{bmatrix} R_{init}^{3\times3} & T_{init}^{3\times1} \\ \mathbf{0}^{1\times3} & 1 \end{bmatrix} \tag{7}$$

$$\Delta\theta = \begin{bmatrix} R^{3\times3} & T^{3\times1} \\ \mathbf{0}^{1\times3} & 1 \end{bmatrix} \tag{8}$$

the final refined pose can be calculated as follows:

$$\begin{aligned} \theta &= \theta_{init} \times \Delta\theta \\ &= \begin{bmatrix} R_{init} \times R & R_{init} \times T + T_{init} \\ \mathbf{0}^{1\times3} & 1 \end{bmatrix} \end{aligned} \tag{9}$$

IV. Experiment

A. Datasets

We evaluate our proposed method on two benchmark datasets.

a) YCB-Video: The YCB-Video dataset consists of 21 objects selected from the YCB object set. It contains 92 videos captured by RGBD cameras, and each video consists of 3-9 objects, leading to a total of over 130K frames. We followed previous works [8], [10], [34] to split them into the training and testing sets. The training set also includes 80,000 synthetic images released by [10].

b) Occlusion LineMOD: The Occlusion LineMOD dataset [6] is re-annotated from the Linemod [7] dataset to compensate for its lack of occlusion. Unlike the LineMOD dataset, each frame in the Occlusion LineMOD dataset contains multiple heavily occluded objects, making it more challenging. We follow the previous work [10] to split the training and testing sets and generate synthetic images for training.

B. Evaluation Metrics

We use the average distance (ADD) and average distance for symmetric objects (ADD-S) as metrics. Given the set of object point cloud $\mathcal{O}$ with m points, the ground truth pose $[R^*, T^*]$ and predicted pose $[R, T]$, ADD, and ADD-S are defined as follows:

$$ADD = \frac{1}{m} \sum_{x \in \mathcal{O}} ||(Rx + T) - (R^*x + T^*)||, \tag{10}$$

$$ADD\text{-}S = \frac{1}{m} \sum_{x_1 \in \mathcal{O}} \min_{x_2 \in \mathcal{O}} ||(Rx_1 + T) - (R^*x_2 + T^*)||, \tag{11}$$

In the YCB-Video dataset, we report the area under ADD-S and ADD(S) curve (AUC) following [10] and set the maximum threshold of success cases to be 0.1m. The ADD(S) computes ADD-S for symmetric objects and ADD for others. In the Occlusion LineMOD dataset, we report the accuracy of ADD(S) with distance less than 10% (ADD(S)-0.1) and 5% (ADD(S)-0.05) of the diameter of the objects.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

TABLE I
RESULT OF 6D POSE ESTIMATION ON THE YCB-VIDEO DATASET USING FFB6D OUTPUT AS INITIAL POSE. THE ADD-S AND ADD(S) AUC ARE REPORTED. OBJECTS WITH BOLD NAMES ARE SYMMETRIC. BOLD VALUES ARE THE HIGHEST SCORE. UNDERLINE VALUES INDICATE RESULTS LOWER THAN THE INITIAL VALUE.

	FFB6D [34]		+GeoTransformer [27]		+ICP-plane [35]		+GICP [36]		+Colored 6D [37]		+Ours	
	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)
master_chef_can	96.4	80.7	95.4	80.1	95.1	76.9	95.3	79.1	95.5	80.1	**96.5**	**81.7**
cracker_box	96.4	95.0	94.8	93.2	96.4	95.4	96.3	94.9	94.9	94.1	**96.7**	**95.6**
sugar_box	97.7	96.8	95.7	94.9	97.6	96.7	97.8	97.1	97.6	97.0	**98.0**	**97.3**
tomato_soup_can	95.8	88.1	95.2	88.0	**95.9**	82.4	95.6	84.3	95.5	87.4	**95.9**	**88.3**
mustard_bottle	98.1	97.6	98.3	98.0	98.1	97.6	98.1	97.8	98.2	**98.0**	**98.4**	**98.0**
tuna_fish_can	97.2	91.3	96.5	91.1	96.7	88.7	96.3	86.6	97.2	91.6	**97.3**	**92.0**
pudding_box	96.3	93.1	94.9	91.3	95.0	90.4	96.1	92.6	95.1	90.6	**96.7**	**94.2**
gelatin_box	97.8	95.8	97.8	95.8	97.8	95.0	**98.3**	**97.0**	97.8	95.0	98.1	96.2
potted_meat_can	92.6	89.8	91.2	88.9	92.6	88.9	92.4	88.2	92.3	87.9	**92.9**	**90.0**
banana	97.4	94.9	96.8	94.1	97.9	96.0	**98.2**	**97.1**	97.8	96.0	97.8	95.8
pitcher_base	97.7	97.0	95.5	93.6	97.9	97.4	**98.0**	**97.6**	97.9	97.4	97.8	97.2
bleach_cleanser	96.5	93.7	96.9	94.5	**97.3**	**96.2**	**97.3**	96.0	96.9	95.2	96.9	94.5
bowl	95.8	95.8	97.1	97.1	96.6	96.6	**97.3**	**97.3**	96.5	96.5	96.6	96.6
mug	97.5	95.3	96.8	95.7	97.7	95.1	**97.8**	95.7	97.6	95.4	**97.8**	**96.2**
power_drill	97.3	96.2	97.9	96.7	97.8	97.2	**98.0**	**97.4**	97.6	96.8	97.8	97.2
wood_block	93.1	93.1	92.7	92.7	94.4	94.4	**95.2**	**95.2**	94.5	94.5	94.3	94.3
scissors	**98.1**	**97.1**	96.1	93.6	97.2	95.2	96.3	93.5	95.9	95.6	**98.1**	97.0
large_marker	96.9	90.0	96.6	88.5	98.0	89.9	**98.1**	88.4	97.8	89.8	97.8	**90.5**
large_clamp	**96.8**	**96.8**	96.7	96.7	95.8	95.8	95.4	95.4	95.8	95.8	96.7	96.7
extra_large_clamp	**96.1**	**96.1**	96.0	96.0	94.1	94.1	93.3	93.3	95.4	95.4	95.6	95.6
foam_brick	97.6	97.6	97.4	97.4	**97.8**	**97.8**	97.6	97.6	97.3	97.3	97.7	97.7
ALL	96.6	92.9	96.0	93.2	96.5	91.9	96.4	92.1	96.3	92.8	**96.8**	**93.4**

C. Implementation Details

All experiments were conducted on a PC with an Intel E5-2640-v4 CPU and NVIDIA RTX2080Ti GPU. For the segmentation results and initial poses required as inputs, we utilize pre-trained FFB6D [34] and PVN3D [12] results. In Full Flow Bidirectional Fusion Network, a PSPNet [54] with a ResNet34 [55] pre-trained on ImageNet [56] is applied to extract RGB image features. For each object, We randomly sample 2,048 points and apply RandLA-Net [57] to extract the point cloud geometry features. These features are then fused by DenseFusion [8]. The PCN-ENC block consists of two stacks of PointNet to extract the point cloud geometry features. The keypoint detector block consists of two MLPs whose details are shown in Fig. 2. We follow [16] to employ a multi-stage structure in PCN-DEC to output a coarse point cloud (2,048 points) and a detailed point cloud (8,192 points). We set $\alpha = \beta = 1, \gamma = 10$ in Eq. (5) and search radius r to be 0.8 times the radius of the selected object. For the keypoints, we apply the SIFT-FPS [34] algorithm to select $K = 8$ keypoints for each target object. We set threshold $m_1 = 100$ and $m_2 = 3000$ in Alg. 1, we also set the tolerance to 0.00001 and max iteration number to 50. For the threshold ϵ in Eq. (6), we set $\epsilon = 5e^{-3}$, This means that if the Euclidean distance between two points is less than or equal to 5mm, the color distance weight will always be equal to the space distance weight.

D. Evaluation on the YCB-Video Dataset

Table I shows the results for all the 21 objects on the YCB-Video dataset using the pre-trained FFB6D [34] model output as the initial pose. Additionally, Table II shows the results when the pretrained PVN3D [12] model output is the initial pose. We report the ADD-S AUC and ADD(S) AUC metrics for four ICP variants and our proposed method.

The results in Table I show that when FFB6D output is the initial pose, the point-to-plane ICP method [35] and the GICP method [36] yield inferior results compared to the initial ones. Additionally, all ICP variants are inferior to the initial results on almost half of the objects, which confirms that the classic ICP methods make it difficult to optimize the existing high-precision pose estimation method. Besides, the learning-based approach GeoTransformer [27] is also inferior to the initial pose in most cases. In contrast, our method only shows slightly inferior results in scissors, large clamp, and extra-large clamp, where two of the objects are only 0.1% lower than the initial results. It is worth noting that the point-to-plane ICP and GICP methods show better results in ADD-S than in ADD(S), which demonstrates that these two methods encounter a significant amount of mismatch in establishing correspondence between the two point clouds. The reason behind this is the use of point-to-plane or plane-to-plane registration methods, which converge quickly in large-scale point cloud registration. However, these methods are more prone to mismatches when solving high-precision registration of small objects.

The results in Table II show that when PVN3D output is used as the initial pose, our method also outperforms the initial results in ADD-S by 0.3% and is one of the highest results among all methods. Compared with FFB6D, all ICP methods improve the original results when PVN3D output is used as the initial pose. However, point-to-plane ICP and GICP are still inferior to the initial results on 6 objects in ADD(S),

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

TABLE II
RESULT OF 6D POSE ESTIMATION ON THE YCB-VIDEO DATASET USING PVN3D OUTPUT AS INITIAL POSE.

	PVN3D [12]		+GeoTransformer [27]		+ICP-plane [35]		+GICP [36]		+Colored 6D [37]		+Ours	
	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)
master_chef_can	96.0	80.8	95.1	80.1	95.0	77.6	95.1	79.4	**96.1**	**81.1**	95.7	80.4
cracker_box	96.0	94.5	93.8	93.4	**96.4**	**95.3**	96.3	94.9	96.2	94.9	96.3	95.2
sugar_box	97.1	95.5	97.1	95.8	97.6	96.7	**97.6**	**96.9**	97.4	96.1	**97.6**	**96.9**
tomato_soup_can	95.5	88.4	95.1	88.1	**95.8**	82.8	95.5	83.8	95.5	88.5	**95.8**	**88.9**
mustard_bottle	97.8	97.0	96.9	96.7	**98.1**	97.6	**98.1**	**97.8**	97.9	97.2	97.9	97.2
tuna_fish_can	96.3	90.1	94.9	90.0	96.8	88.5	96.4	87.2	96.3	90.2	**97.0**	**90.4**
pudding_box	96.9	95.3	94.9	91.3	94.9	90.4	95.9	92.1	**97.2**	**95.8**	**97.2**	95.4
gelatin_box	97.8	96.2	97.7	95.8	97.7	94.9	**98.2**	**97.0**	97.8	96.3	**98.2**	96.3
potted_meat_can	93.0	88.6	91.2	88.9	92.8	88.3	92.6	87.7	92.9	88.4	**93.1**	**88.9**
banana	96.5	93.5	96.8	94.5	97.9	96.0	**98.1**	**96.9**	96.9	94.3	**98.1**	96.5
pitcher_base	96.9	95.6	95.8	94.8	97.9	97.4	**98.0**	**97.5**	97.1	96.1	97.5	96.6
bleach_cleanser	96.3	93.6	96.4	94.3	**97.3**	**96.2**	97.2	95.9	96.6	94.4	96.8	94.7
bowl	89.4	89.4	**94.1**	**94.1**	92.4	92.4	90.9	90.9	91.5	91.5	92.8	92.8
mug	97.4	95.0	96.7	95.5	97.8	95.5	97.8	95.7	97.5	95.4	**97.9**	**96.0**
power_drill	96.6	95.1	96.8	95.7	97.8	97.2	**98.0**	**97.4**	97.0	95.8	97.5	96.6
wood_block	90.4	90.4	91.8	91.8	**93.9**	**93.9**	93.4	93.4	90.5	90.5	91.5	91.5
scissors	96.5	92.7	93.1	90.3	**97.2**	**95.1**	95.4	91.4	96.8	93.2	96.0	92.7
large_marker	96.8	91.7	96.5	88.4	98.2	91.4	98.3	91.3	97.2	92.1	**98.4**	**92.7**
large_clamp	90.5	90.5	91.7	91.7	**93.0**	**93.0**	92.6	92.6	91.7	91.7	92.3	92.3
extra_large_clamp	87.1	87.1	90.0	90.0	**90.9**	**90.9**	89.6	89.6	88.8	88.8	89.6	89.6
foam_brick	96.8	96.8	96.4	96.4	**97.6**	**97.6**	97.1	97.1	96.7	96.7	97.3	97.3
ALL	95.2	91.5	94.9	92.3	**95.9**	91.6	**95.9**	91.6	95.6	92.0	**95.9**	**92.5**

whereas Colored 6D ICP and ours are only inferior to the initial results on 2 objects each. With the ADD(S) evaluation metric, GeoTransformer exhibits inferior performance compared to the initial results on 13 objects. The learning-based method GeoTransformer demonstrates poor performance when utilizing the initial poses from PVN3D and FFB6D. We believe this is primarily due to the fact that learning-based methods operate as black boxes, which are primarily designed for registration rather than refinement. Additionally, the lack of adjustable parameters specific to the task contributes to their suboptimal performance. These results reinforce the stability of our proposed method.

Figure 3 shows some of the qualitative test results using the PVN3D output as the initial pose. The figure reveals that our proposed method produces a more stable result, with a smaller change in the initial pose compared to the ICP method. Furthermore, the proposed method's performance improves with the assistance of the completed point cloud in scenes with severe occlusion, leading to excellent results.

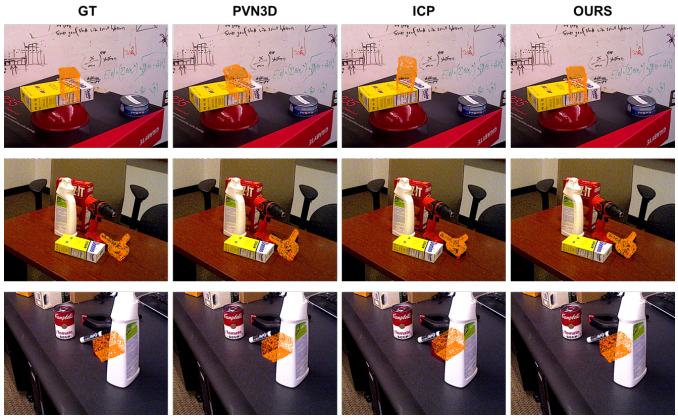


Fig. 3. Qualitative results on the YCB-Video Dataset. Our method can outperform other methods with both low-precision initial pose (row 1) and high-precision initial pose (rows 2 and 3) with higher accuracy.

E. Evaluation on the Occlusion LineMOD Dataset

Table III and Table IV show the experimental results on the Occlusion LINEMOD dataset using FFB6D and PVN3D outputs as initial poses, respectively. The results demonstrate that when using FFB6D as the initial pose, our method outperforms GICP and point-to-plane ICP by 0.9% in ADD(S)-0.1 and improves the initial results by 2.0%. Moreover, our method also outperforms GICP by 0.7% in ADD(S)-0.05 and improves the initial results by 6.9%. In contrast, all other ICP variants perform worse than the initial results for almost half of the objects in both ADD(S)-0.1 and ADD(S)-0.05, although their overall mean is higher than the initial results. These results indicate that our proposed method can maintain stability under severe occlusion and significantly improve the overall performance compared to other ICP methods. GeoTransformer performs better on ADD(S)-0.05 than on ADD(S)-0.1, achieving optimal results on two object categories. In summary, its performance has improved compared to the YCB-Video dataset. However, there is still a gap between its performance and our method, and the improvement is not consistent enough.

Figure 4 shows the qualitative results on the Occlusion LINEMOD dataset, with FFB6D output as the initial pose. Our proposed method significantly outperforms the ICP method in severely occluded scenes, which can be attributed to the completed point cloud added by our method and the step-by-step keypoint optimization strategy. In addition, it is clear that the third row of the ICP method not only failed to

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

TABLE III
RESULT OF 6D POSE ESTIMATION ON OCCLUSION LINEMOD DATASET WITH FFB6D OUTPUT AS INITIAL POSE. THE ADD(S)-0.1 AND ADD(S)-0.05 METRICS ARE REPORTED. OBJECTS WITH BOLD NAMES ARE SYMMETRIC. BOLD VALUES ARE THE HIGHEST SCORE. UNDERLINE VALUES INDICATE RESULTS LOWER THAN THE INITIAL VALUE.

	FFB6D [34]		+GeoTransformer [27]		+ICP-plane [35]		+GICP [36]		+Colored 6D [37]		+Ours	
ADD(S)-N	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05
ape	55.0	20.0	55.3	21.2	57.2	18.4	57.2	19.7	57.3	19.7	**57.6**	**21.9**
can	82.5	56.3	82.1	**65.7**	81.6	60.4	81.6	60.6	81.7	63.1	**83.4**	64.5
cat	37.1	14.1	37.9	18.3	37.0	16.5	36.9	16.1	36.9	16.2	**38.3**	**22.3**
driller	76.5	50.7	80.3	67.3	83.4	71.6	**84.3**	**73.6**	81.8	68.3	79.8	61.4
duck	56.6	13.6	56.9	17.3	60.7	14.9	60.3	16.6	60.4	17.0	**61.4**	**17.9**
eggbox	59.1	19.7	60.6	24.8	58.2	24.6	57.4	**26.1**	57.6	25.1	**59.5**	24.1
glue	66.7	53.8	65.8	52.1	64.1	53.3	64.0	52.9	63.9	50.7	**66.9**	**57.7**
holepuncher	84.6	41.4	83.5	**57.6**	85.3	49.6	85.6	53.4	84.6	57.2	**87.4**	54.6
Average	64.8	33.7	65.3	39.7	65.9	38.7	65.9	39.9	65.5	39.7	**66.8**	**40.6**

TABLE IV
RESULT OF 6D POSE ESTIMATION ON OCCLUSION LINEMOD DATASET USING PVN3D OUTPUT AS INITIAL POSE.

	PVN3D [12]		+GeoTransformer [27]		+ICP-plane [35]		+GICP [36]		+Colored 6D [37]		+Ours	
ADD(S)-N	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05	0.1	0.05
ape	57.5	15.7	57.2	15.8	62.1	18.1	62.3	19.6	62.5	19.1	**63.8**	**22.1**
can	91.8	40.9	92.3	64.5	92.5	60.6	92.4	60.5	92.8	61.7	**94.8**	**65.7**
cat	32.4	5.2	37.2	17.4	37.6	14.2	37.2	14.2	37.4	14.2	**39.8**	**14.7**
driller	76.6	19.0	80.1	**66.4**	77.5	63.6	75.0	64.2	81.5	65.3	**82.3**	57.8
duck	33.9	2.3	53.3	**16.9**	50.9	12.8	54.9	**15.2**	50.9	13.1	50.5	13.3
eggbox	**64.6**	15.0	58.6	19.9	60.9	19.3	62.0	20.8	61.2	19.2	63.7	**21.2**
glue	69.8	43.3	67.8	46.4	69.4	50.1	67.7	51.2	67.9	46.1	**70.8**	**53.9**
holepuncher	70.1	18.8	79.5	48.6	79.8	38.7	81.9	48.4	80.5	44.2	**83.7**	**49.1**
Average	62.1	20.0	65.8	37.0	66.3	34.7	66.7	36.8	66.8	35.4	**68.7**	**37.2**

effectively optimize the initial pose but produced even worse results. This can be attributed to the severe occlusion and cluttered scenes, which cause a significant deviation between the given segmentation result and the ground truth label, thereby disrupting the post-processing optimization. Figure 5 shows the given segmentation result in this scene, where valid masks are marked by a red box, and incorrect masks by a white box. It is evident that the inaccurate labeling in the white box has a significant impact on the performance of the ICP method, resulting in erroneous optimization outcomes. Conversely, our method can effectively mitigate the impact of inaccurate segmentation by leveraging completed point clouds and local registration strategies.

In the experiment using PVN3D output as the initial pose, our method and the ICP variants showed a greater improvement in the initial results compared to the previous set of experiments. In ADD(S)-0.1, our method outperformed the Colored 6D ICP by 1.9%, achieving an improvement of 6.6% over the initial result. In ADD(S)-0.05, our method outperformed GICP by 0.4%, achieving an improvement of 17.2% over the initial result. Notably, in the stricter ADD(S)-0.05, the effect of post-processing optimization is greatly improved, which highlights the importance of further optimizing the results at the end. In this experiment, our method achieved the best results for most objects, with the eggbox being the only exception where it performed lower than the initial result on ADD(S)-0.1. Although the point-to-point ICP method outperformed our method by 7.6% on the ADD(S)-0.05 of the driller, our method showed superior performance and stability over the ICP variants in most of the other results.

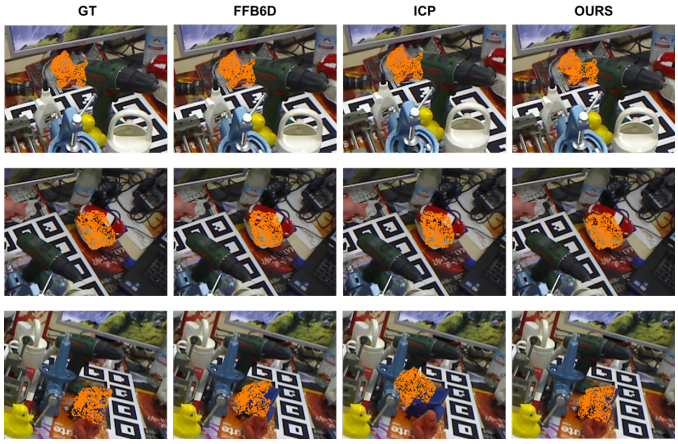


Fig. 4. Qualitative results on the Occlusion LineMOD Dataset. Ours can outperform other methods with both high-precision initial pose (row 1) and low-precision initial pose (rows 2 and 3) with higher accuracy.

It is worth noting that while the PVN3D's pose estimation results may not be as accurate as those of FFB6D, the post-processing optimization still produced lower results for ADD(S)-0.1 compared to FFB6D, whether using ICP variants or our method. However, the optimization results for FFB6D are still higher than those of PVN3D when evaluating using a stricter metric like ADD(S)-0.05. This may be due to

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

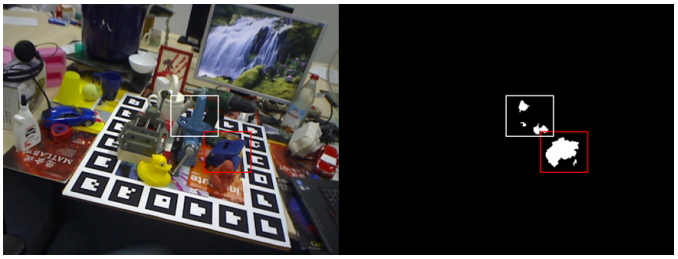


Fig. 5. Incorrect segmentation results lead to poor performance of the ICP method. Valid masks are marked by a red box, and incorrect masks by a white box.

TABLE V
RGB MASK EXPERIMENT

Method \ Mask ratio	25%	50%	75%	100%
PVN3D [12]+Ours+YCB	92.5	92.4	92.4	92.1
FFB6D [34]+Ours+YCB	93.4	93.4	93.3	93.1
PVN3D [12]+Ours+LO	68.5	68.2	68.0	67.1
FFB6D [34]+Ours+LO	66.7	66.7	66.3	65.9

the differences in segmentation results obtained by different methods, which can result in variations in the quality of the point cloud used in registration, ultimately affecting the overall process. Nevertheless, the estimated results for FFB6D are still significantly higher than those of PVN3D after optimization when evaluating using a stricter metric, which is consistent with expectations.

F. RGB Mask Experiment

To investigate the impact of texture information on our proposed method, we randomly masked a proportion of RGB information from points in the synthetic data from the YCB-Video and Linemod Occlusion datasets, gradually increasing the masking ratio until all RGB information was obscured.

In Table V, LO denotes the Linemod Occlusion dataset, while YCB represents the YCB-Video dataset. The experimental results demonstrate several points. Firstly, our approach can maintain a stable improvement even under conditions of low texture and no texture. Secondly, the effectiveness of our method experiences the largest decrease when transitioning from low texture (75%) to no texture (100%). Finally, in the absence of texture, only the keypoint module in our approach remains functional, and experimental results show that this module still provides a stable improvement.

G. Experiments on symmetric objects

We conducted a comparative experiment by selecting seven symmetrical objects from the YCB-Video dataset and LineMOD datasets to compose a symmetrical object dataset. This dataset comprises the bowl, wood block, large clamp, extra large clamp, and foam brick from the YCB dataset, as well as the egg box and glue from the LineMOD dataset. The experimental results indicate that, in the case of symmetrical objects, the performance of refinement methods is generally unsatisfactory. Most refinement approaches even deteriorate the accuracy of FFB6D. Only our proposed method consistently maintains a significant and stable improvement, further corroborating the robustness of our approach.

TABLE VI
COMPARISON OF SYMMETRIC OBJECTS

	symmetric objects dataset
PVN3D [12]	84.1
PVN3D [12]+ICP-point [14]	83.8
PVN3D [12]+ICP-plane [35]	85.4
PVN3D [12]+GICP [36]	84.8
PVN3D [12]+Colored 6D [37]	84.0
PVN3D [12]+Ours	85.4
FFB6D [34]	86.5
FFB6D [34]+ICP-point [14]	85.9
FFB6D [34]+ICP-plane [35]	85.9
FFB6D [34]+GICP [36]	85.7
FFB6D [34]+Colored 6D [37]	85.9
FFB6D [34]+Ours	**86.8**

H. Experiments about point cloud completion

In addition to comparing the entire PCKRF framework, we also conduct experiments on the point cloud completion method. For our pipeline, we are particularly concerned with the details around keypoints, so we designed a keypoint detector as a decoder to enhance the details around keypoints. However, we also found that our detector did not perform well for overly complex completion frameworks, and even had a counterproductive effect. Based on this, we chose PCN as our backbone.

Table VIII is performed on the YCB dataset. We employed SeedFormer [51], a Transformer-based point cloud completion method, along with the traditional PCN [16] as the backbone while keeping other modules unchanged apart from the completion network. Experimental results demonstrated that the inclusion of SeedFormer for point cloud completion had minimal impact on the final outcome. Surprisingly, the direct application of PCN even showed adverse effects on the results. However, when combined with our custom-designed completion network, it positively influenced the accuracy. We postulate that this is primarily due to PCN's superior adaptability compared to Transformer in handling bidirectional fusion of RGB and point cloud information for the task of point cloud completion.

Table VII shows the experimental results of applying point cloud completion methods to point cloud registration on the YCB-Video dataset, with ADD(S) AUC serving as the evaluation metric. We evaluate the impact of three methods: not using any point cloud completion (None), using the PCN method, and using our proposed method (Ours) with the point-to-point ICP, Colored 6D ICP, and CIKP methods, respectively. The results show that regardless of whether the initial pose estimation results of FFB6D or PVN3D are used, using the PCN method as the point cloud completion method is similar to the optimization results without using point cloud completion for point-to-point ICP and Colored 6D ICP methods. However, for the CIKP method, the result is the opposite and

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

TABLE VII
EXPERIMENTAL RESULTS USING DIFFERENT POINT CLOUD COMPLETION METHODS ON THE YCB-VIDEO DATASET

	Init	ICP-point [14]			Colored 6D ICP [37]			CIKP		
Method / Init Pose	/	None	PCN	Ours	None	PCN	Ours	None	PCN	Ours
FFB6D	92.9	92.8	92.7	**92.9**	92.8	92.9	**93.1**	93.2	92.9	**93.4**
PVN3D	91.5	91.9	92.0	**92.2**	92.0	92.0	**92.2**	92.2	91.8	**92.5**

TABLE VIII
COMPARISON OF DIFFERENT POINT CLOUD COMPLETION METHODS

	YCB-Video Dataset	
	FFB6D	PVN3D
None	93.2	92.2
PCN [16]	92.9	91.8
SeedFormer [51]	93.2	92.0
SeedFormer [51]+Ours	93.2	92.0
PCN+Ours	**93.4**	**92.5**

PCN gets lower performance than without completion. On the contrary, after applying our proposed point cloud completion method, the results of the three methods are improved to varying degrees compared to those without a point cloud. In particular, the CIKP method registers the point cloud near the keypoint. Without completion, only visible points near the keypoints participate in optimization. With completion, all keypoints (visible or not), participate in optimization. As a result, the impact of completed point clouds on the CIKP method is more significant than on the ICP and Colored 6D ICP methods. Therefore, the effect of the CIKP method drops significantly when the PCN method is adopted, while the other methods are not affected as much.

The scale of the sampled point cloud has a significant impact on point cloud registration. To determine the appropriate scale for participating in registration, we utilized the PVN3D output as the initial pose on the YCB-Video dataset. The influence of different point cloud scales on the point-to-point ICP method and CIKP method is shown in Table IX. It can be observed that both the point-to-point ICP method and the CIKP method achieved the best results when the point cloud size was set to 3000. As a result, we selected a downsampling threshold $m_2 = 3000$ in Section IV-C to ensure consistency with the best-performing approaches.

I. Ablation Study

In this part, we conduct ablation experiments on the point cloud completion network and the CIKP method respectively.

TABLE IX
INFLUENCE OF DIFFERENT POINT CLOUD SCALES ON ICP-POINT AND CIKP METHOD

Size of Point Cloud	ICP-point	CIKP
2000	91.94	91.92
3000	**92.19**	**92.47**
5000	92.16	92.22

TABLE X
ABLATION STUDY FOR COMPLETION NETWORK ON THE YCB-VIDEO DATASET. KPDEC MEANS KEYPOINT DETECTOR BLOCK.

FFB	DF	KPDEC	ADD(S)
			92.7
✓			93.1
✓		✓	93.2
✓	✓	✓	**93.4**

TABLE XI
ABLATION STUDY FOR CIKP ON THE YCB-VIDEO DATASET. KP:REFINEMENT BY KEYPOINTS. PCLD:USING COMPLETION NETWORK.

	Init Pose	KP	Color	Pcld	Rotation	ADD(S)
FFB6D	92.9		✓			93.0
		✓				93.1
		✓	✓			93.2
		✓		✓		93.2
		✓	✓	✓		**93.4**
		✓	✓	✓	✓	91.2
PVN3D	91.5	✓				92.1
		✓	✓			92.2
		✓		✓		92.4
		✓	✓	✓		**92.5**

a) Completion Network: Table X shows the ablation study of the point cloud completion network on the YCB-Video dataset. The table includes three modules: FFB, DF, and KPDEC, which respectively stand for Full Flow Bidirectional fusion module, DenseFusion feature fusion module, and the keypoint detection module used only during the training process. The first row represents the original PCN network without any modification. The results show that all three modules effectively improve the performance of the network. The FFB module allows for the full fusion of texture and point cloud features of the object in each pixel, while the feature is extended by stacking DF modules to prevent the loss of critical information when concatenating with global features of the point cloud. The KPDEC module, as previously shown, leads to a more accurate pose estimation when the overall quality of the completed point cloud is not much different.

b) CIKP: Table XI shows the ablation study of the CIKP approach with various initial poses on the YCB-Video dataset. The abbreviation KP stands for the iterative keypoint optimization method for pose estimation; while in its absence, the entire point cloud is optimized iteratively, similar to ICP. Color stands for the use of color information, while Pcld stands for the utilization of the completed point cloud generated by our completion network. Rotation refers to the consideration of both translation and rotation

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

TABLE XII
COMPARISON OF POSE REFINEMENT METHODS WITH DIFFERENT INITIAL POSE ACCURACY ON YCB-VIDEO DATASET

Method	ICP-point	ICP-plane	GICP	Ours
Level1	92.2	91.5	91.9	**92.3**
Level2	90.3	89.7	88.9	88.6
Level3	87.4	85.6	84.3	72.1

TABLE XIII
TIME COMPARISON OF POSE REFINEMENT METHODS

Method	ICP-point	Ours	GICP	ICP-plane
Time(s/object)	0.77	0.66	0.53	**0.48**

terms during keypoint optimization. The results indicate that when considering the rotation item, the ADD(S) of the CIKP method decreases significantly, suggesting that overfitting of the optimal transformation of the local point cloud around the keypoint is a significant issue. This issue is resolved by solely considering the optimal translation transformation during each keypoint's pose optimization. Regarding other ablation items, it is evident that when using the predicted pose of FFB6D as the initial pose, all modules contribute similarly to the overall result, and the iterative optimization of each keypoint ensures optimization process stability. The use of color information can introduce texture information to solve the optimization problem of regular-shaped but rich-textured objects, whereas the completed point cloud provides occluded point cloud information that is absent in the input information. In addition, the results obtained by using the predicted pose of PVN3D as the initial pose show that the performance of adding completed point clouds is superior to that of adding color information. This is due to unsatisfactory cloud segmentation results, which affect the performance of all methods, including the post-processing optimization. However, this issue is alleviated after incorporating the completed point cloud.

J. Limitations

We conduct a comparison of pose refinement methods with different initial pose accuracy on the ycb-video dataset as shown in Table XII. Level1,2,3 respectively represent a fixed initial pose difference of 5 degrees and 10 centimeters, 10 degrees and 20 centimeters, and 15 degrees and 30 centimeters. Experimental results in Table XII reveal that as the initial pose accuracy diminishes, the effectiveness of our method rapidly declines. Conversely, methods like ICP exhibit better performance than our approach under these conditions. The reason is that when our method selects the point cloud around the keypoint, it chooses from the vicinity of the keypoint previously selected in the object coordinate. If the initial pose is too far from the ground truth, our method will not be able to select enough points for refinement and effective optimization. Based on this limitation, we also speculated whether point cloud noise would have a more significant impact on our method. However, during testing, it was found that as noise increased, the effectiveness of both our method and the registration method declined at a similar rate, showing no significant difference. In future work, we may introduce the superpoint method [27], [32] to dynamically select the point cloud for the refinement process from object models.

Table XIII shows the time comparison of ICP variants and our method. It can be observed that our method performs significantly more work than the ICP method in a single iteration, but the overall time is slightly faster than the ICP-point method, indicating that our method can converge relatively quickly. Hence, our method can substitute the original ICP method when it is necessary to optimize high-precision pose estimation. However, our method is still slower than the GICP and point-to-plane method and is not suitable enough for real-time environments, which is another direction that can be improved in the future.

V. CONCLUSIONS

In this paper, we proposed a pose refinement pipeline PCKRF that integrates the point cloud completion network and CIKP method. The point cloud completion network incorporates a keypoint detection module during the training process to enhance the sensitivity of the completed point cloud, thereby improving the performance of pose refinement. The CIKP method employs a keypoint refinement strategy and incorporates color information to enhance the accuracy and stability of the refinement results. Experiments show that all novel components are effective, and our method outperforms existing refinement methods in optimizing high-precision pose estimation methods on both the YCB-Video dataset and the Occlusion LineMOD dataset. Notably, the results unequivocally demonstrate that our method can seamlessly integrate with the majority of existing pose estimation techniques, resulting in significantly enhanced performance across most cases. Additionally, our approach consistently yields promising outcomes, even in challenging situations characterized by textureless and symmetrical objects. Our experiments also demonstrate that current learning-based point cloud registration methods are not suitable enough for pose refinement. In future work, we will explore the possibility of applying learning-based registration approaches to pose refinement.

REFERENCES

[1] M. Zhu, K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3936–3943.

[2] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[3] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[5] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

[6] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European Conference on Computer Vision*. Springer, 2014, pp. 536–551.

[7] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 548–562.

[8] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352.

[9] Z. Dong, S. Liu, T. Zhou, H. Cheng, L. Zeng, X. Yu, and H. Liu, "PPR-Net:Point-wise Pose Regression Network for Instance Segmentation and 6D Pose Estimation in Bin-picking Scenarios," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1773–1780.

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

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

[12] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 632–11 641.

[13] L. Zeng, W. J. Lv, X. Y. Zhang, and Y. J. Liu, "ParametricNet: 6DoF Pose Estimation Network for Parametric Shapes in Stacked Scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 772–778.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

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

[16] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 728–737.

[17] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6dof pose estimation for textureless objects," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2441–2448.

[18] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2011.

[19] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1521–1529.

[20] Y. Lin, T. Müller, J. Tremblay, B. Wen, S. Tyree, A. Evans, P. A. Vela, and S. Birchfield, "Parallel inversion of neural radiance fields for robust pose estimation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9377–9384.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[22] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[23] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 753–11 762.

[24] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6359–6367.

[25] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2514–2523.

[26] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 366–11 374.

[27] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 143–11 152.

[28] D. Lee, O. C. Hamsici, S. Feng, P. Sharma, and T. Gernoth, "Deeppro: Deep partial point cloud registration of objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5683–5692.

[29] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3d registration with maximal cliques," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 745–17 754.

[30] H. Jiang, M. Salzmann, Z. Dang, J. Xie, and J. Yang, "Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[31] H. Yu, F. Li, M. Saleh, B. Busam, and S. Ilic, "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 872–23 884, 2021.

[32] Q. Zhou, T. Sattler, and L. Leal-Taixe, "Patch2pix: Epipolar-guided pixel-level correspondences," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4669–4678.

[33] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 292–301.

[34] Y. He, H. Huang, H. Fan, Q. Chen, and J. 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*, 2021, pp. 3003–3013.

[35] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[36] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP." in *Robotics: Science and Systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[37] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3d data," *Image and vision computing*, vol. 17, no. 2, pp. 135–147, 1999.

[38] H. Men, B. Gebre, and K. Pochiraju, "Color point cloud registration with 4D ICP algorithm," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1511–1516.

[39] M. Korn, M. Holzkothen, and J. Pauli, "Color supported generalized-ICP," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3. IEEE, 2014, pp. 592–599.

[40] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[41] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3450–3466, 2021.

[42] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2010.

[43] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 095–11 104.

[44] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, "Deep model-based 6d pose refinement in rgb," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[45] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[46] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 206–215.

[47] X. Wang, M. H. Ang Jr, and G. H. Lee, "Cascaded refinement network for point cloud completion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 790–799.

[48] Y. Xia, Y. Xia, W. Li, R. Song, K. Cao, and U. Stilla, "Asfm-net: Asymmetrical siamese feature matching network for point completion," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 1938–1947.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

[49] H. Wu, Y. Miao, and R. Fu, "Point cloud completion using multiscale feature fusion and cross-regional attention," *Image and Vision Computing*, vol. 111, p. 104193, 2021.
[50] L. Zhu, B. Wang, G. Tian, W. Wang, and C. Li, "Towards point cloud completion: Point rank sampling and cross-cascade graph cnn," *Neurocomputing*, vol. 461, pp. 1–16, 2021.
[51] H. Zhou, Y. Cao, W. Chu, J. Zhu, T. Lu, Y. Tai, and C. Wang, "Seedformer: Patch seeds based point cloud completion with upsample transformer," in *European conference on computer vision*. Springer, 2022, pp. 416–432.
[52] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
[53] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.
[54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[56] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.
[57] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.

Yiheng Han is currently an assistant professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. He received his B.Eng. degree from Jilin University, China, in 2018, and his Ph.D. degrees from Tsinghua University, China, in 2023. His research interests include robot active vision, motion planning and computer vision.

Irvin Haozhe Zhan is a master degree candidate with the Department of Computer Science and Technology, Tsinghua University, China. He received his B. Eng. degree from Tsinghua University in 2020. His research interests include computer vision, robotics and deep learning.

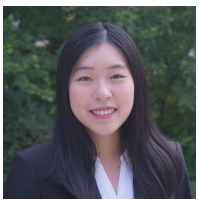
Long Zeng is an associate professor with the Shenzhen International Graduate School, Tsinghua University, China. He received his M.Phil degree from Zhejiang Unversity, China, in 2007, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2012. His research interests include intelligent manufacturing, computer-aided design, and robots. For more information, visit https://www.sigs.tsinghua.edu.cn/cl/main.htm

Yu-Ping Wang is an associate professor at the School of Computer Science and Technology, Beijing Institute of Technology. He received his B.S. degree from Northeastern University, China, in 2002, and his M.E. and Ph.D. degrees from Tsinghua University, China, in 2005 and 2009, respectively. His research interests include robot operating system, and distributed robotic system.

Ran Yi is an assistant professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. She received the BEng degree and the PhD degree from Tsinghua University, China, in 2016 and 2021. Her research interests include computer vision, computer graphics and computational geometry.

Minjing Yu received the BE degree from Wuhan University, Wuhan, China, in 2014, and the PhD degree from Tsinghua University, Beijing, China, in 2019. She is currently an associate professor with the College of Intelligence and Computing, Tianjin University, China. Her research interests include computer graphics, artificial intelligence, and cognitive computation.

Matthieu Gaetan Lin is PhD candidate with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computer vision, intelligent media processing, and human-computer interaction.



Jenny Sheng is a master degree candidate with the Department of Computer Science and Technology, Tsinghua University, China. She received her B.S.E. degree in Computer Science from Princeton University in 2022. Her research interests include computer vision, intelligent media processing, and human-computer interaction.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3390122

Yong-Jin Liu is a professor with the Department of Computer Science and Technology, Tsinghua University, China. He received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. His research interests include computer vision, computer graphics and computer-aided design. For more information, visit http://cg.cs.tsinghua.edu.cn/people/~Yongjin/Yongjin.htm.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on April 29,2024 at 03:56:43 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.