Depth-based 6DoF Object Pose Estimation using Swin Transformer

Zhujun Li[1] and Ioannis Stamos[1,2]

Abstract— **Accurately estimating the 6D pose of objects is crucial for many applications, such as robotic grasping, autonomous driving, and augmented reality. However, this task becomes more challenging in poor lighting conditions or when dealing with textureless objects. To address this issue, depth images are becoming an increasingly popular choice due to their invariance to a scene's appearance and the implicit incorporation of essential geometric characteristics. However, fully leveraging depth information to improve the performance of pose estimation remains a difficult and under-investigated problem. To tackle this challenge, we propose a novel framework called SwinDePose, that uses only geometric information from depth images to achieve accurate 6D pose estimation. SwinDePose first calculates the angles between each normal vector defined in a depth image and the three coordinate axes in the camera coordinate system. The resulting angles are then formed into an image, which is encoded using Swin Transformer. Additionally, we apply RandLA-Net to learn the representations from point clouds. The resulting image and point clouds embeddings are concatenated and fed into a semantic segmentation module and a 3D keypoints localization module. Finally, we estimate 6D poses using a least-square fitting approach based on the target object's predicted semantic mask and 3D keypoints. In experiments on the LineMod and Occlusion LineMod, SwinDePose outperforms existing state-of-the-art methods for 6D object pose estimation using depth images. We also provide competitive results on the YCB-Video dataset even without post-processing. This demonstrates the effectiveness of our approach and highlights its potential for improving performance in real-world scenarios. Our code is at `https://github.com/zhujunli1993/SwinDePose`.**

I. INTRODUCTION

6D pose estimation involves determining the rigid transformation between camera coordinate and object coordinate systems, including the 3D rotation matrix and the 3D translation vector. This is a critical step in many applications, such as robotic manipulation [1], autonomous driving [2], and augmented reality [3]. For example, in robotic manipulation, robots need the 6D poses of target objects for recognition and grasping [1]. In autonomous driving, vehicles must estimate the 6D poses of roads and obstacles for navigation [2]. In augmented reality, accurately estimating the 6D poses of real-world objects is essential for correctly placing virtual objects [3].

6D pose estimation techniques are generally classified into three groups based on the type of input data: RGB input [4], [5], RGB-D input [6], [7], and depth image input [8], [9], [10]. RGB-based and RGB-D-based methods rely on the appearance information provided by RGB images, limiting their pose estimation performance in challenging scenarios, such as poor lighting or textureless objects. In contrast, depth images provide 3D geometric information that is less sensitive to lighting conditions or texture. Additionally, depth sensors have become more affordable, leading to a growing number of works that focus solely on using depth images for 6D pose estimation.

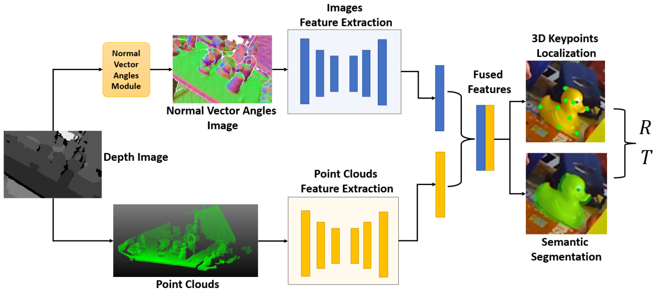


Fig. 1: We propose a novel framework called SwinDePose for 6D object pose estimation from depth images, which are lifted into point clouds and normal vector angle images. Their embeddings are learned for 3D keypoints localization and semantic segmentation, then 6D poses (R and T) of the target object are recovered. In this figure, we project the 3D keypoints onto 2D pixels in the RGB image for visualization.

Traditional depth-based 6D pose estimation methods, like Point Pair Features (PPF) [11], rely on hand-crafted features based on objects' geometry information. However, these features are sensitive to changes in viewpoints, objects' shapes and appearances, and require significant human efforts for feature extraction and model fitting. Recent advancements in deep learning have paved the way for the development of deep neural networks that can learn geometric representations for accurate estimation of 6D poses from depth images [8], [9], [10]. These methods lift depth images to point clouds and design neural networks for 3D geometry representation learning to estimate 6D poses. While these methods have shown promising results, they rely solely on point cloud embeddings and do not fully leverage the features from depth images. Therefore, incorporating 2D representations of depth images could provide additional information to improve pose estimation performance.

To introduce depth image embeddings into 6D pose estimation, treating a depth image as a gray-level intensity image and then applying existing vision algorithms are natural choices. However, traditional vision algorithms may not be well-suited for processing depth images due to noise, missing data, and differences in representation compared to RGB images. [12] opened up new opportunities to overcome these challenges by using normal vectors at each surface

[1] The Graduate Center, City University of New York `zli3@gradcenter.cuny.edu`
[2] Hunter College, City University of New York `istamos@hunter.cuny.edu`

point. In our work, we utilize normal vector angles to represent depth information. Specifically, we calculate the angles between each surface normal vector and the three coordinate axes. Then we normalize them to the RGB color range to form an RGB-like image, where each pixel has three normalized angle values. Such images are fed into our image representation learning network. In addition to the normal vector angles images, we lift depth images to point clouds using the given camera parameters and extract point clouds features via the point clouds representation learning network. Combining these two embeddings, our SwinDePose architecture can leverage depth images and point clouds information for more accurate 6D pose estimation.

Our proposed framework is shown in Fig. 1. We use Swin Transformer [13] for image feature extraction, while using RandLA-Net [14] for point cloud features extraction. The learned image and point cloud embeddings are then fused for 3D keypoints localization and semantic segmentation. Especially for image feature extraction, [6], [7], [15] have used the convolutional neural network (CNN). However, these methods have limitations such as small local receptive fields, sensitivity to object occlusions, and lack of global context. To overcome these limitations, we employ Swin Transformer, which leverages a self-attention mechanism to capture global context by attending to all positions in the input. Additionally, Swin Transformer exhibits robustness to deformations and occlusions, making it suitable for processing complex scenes. Works such as [16], [17] have applied Swin Transformer for computer vision tasks, such as object detection, instance semantic segmentation, and image classification. To the best of our knowledge, this is the first work using Swin Transformer for 6D object pose estimation based on depth information.

To evaluate our method, we conduct experiments on three popular datasets, the LineMod (LM), the Occlusion LineMod (O-LM), and the YCB-Video (YCBV) datasets. Experimental results show that the proposed approach outperforms the state-of-the-art depth-based methods on LM and O-LM.

To summarize, the main contributions of our work are:

- Generating normal vector angle images to fully leverage the geometry information from depth images, that can be combined with point clouds, for feature extraction.
- Introducing a novel framework including Swin Transformer and point cloud networks for 6D pose estimation.
- Achieving SOTA performance based on depth information on LM, and O-LM datasets and competitive results on the YCBV dataset even without post-processing.

II. RELATED WORKS

6D Pose Estimation from Depth Images. Most existing research on 6D pose estimation from depth images using deep learning has primarily focused on converting depth images into point clouds and utilizing existing semantic segmentation models to extract object masks from depth images. These masks are then used to crop objects from point clouds and feed them into their proposed 6D pose estimation framework. For instance, [8] proposed a framework based on the augmented autoencoder and trained it on a large synthetic point cloud dataset. [18] proposed the OVE6D method that was trained on a large synthetic image dataset containing ShapeNet objects. Other systems, such as those introduced by [9] and [10], utilized instance semantic segmentation masks from depth images and point clouds to regress 6D poses. In contrast, our proposed method includes a semantic segmentation module and integrates point clouds embeddings with normal vector angles images embeddings obtained from depth images to estimate 6D poses.

Vision Transformer. The Transformer, originally developed for natural language processing tasks, has been adapted for computer vision tasks and has demonstrated significant improvements in performance. Researchers have designed various transformer networks for object detection, segmentation, and pose estimation tasks. [13] proposed Swin Transformer, which constructs hierarchical feature maps from input images using shifted windows and has been applied as a backbone network for various tasks. For example, [16] designed SimCrossTrans using Swin Transformer for 2D object detection task. [17] introduced a novel image fusion network for multi-modal image fusion and digital photography image fusion. Moreover, [19] utilized Swin Transformer to learn image representations for human pose estimation. Our framework uses Swin Transformer as an encoder to extract image features for 6D object pose estimation.

Depth Images Feature. Exploring depth information from depth images has been a well-studied area in computer vision. For example, surface curvature [20], and kernel features [21] have been used. Another [22] has computed normal vectors from depth images and used spherical angles to represent the normal vectors for object recognition. Inspired by these studies, we extract normal vectors from depth images and compute the angles between each normal vector and the XYZ coordinate axes in the camera coordinate system. These angles are then normalized and grouped as an image, which is fed into the image features extraction module.

III. METHOD

Given the input images, 6D pose estimation predicts the rigid transformation of objects from the object coordinate system to the camera coordinate system. The 6D includes rotation matrix $R \in SO(3)$ and translation vector $T \in \mathbb{R}^3$.

A. Overview.

Our proposed framework follows a pipeline that consists of several steps, as shown in Fig. 2. First, we feed the depth image D into the normal vector angles generation module, which produces the normal vector angles image I_{nrm}. At the same time, we lift D to point clouds P using the given camera parameters K. We then utilize two encoder-decoder networks to learn the representations of I_{nrm} and P, respectively. Specifically, the image embeddings $\mathrm{F}'_{I_{nrm}}$ and point clouds embeddings F_p are extracted and concatenated to form the fused features F_{fuse}. We then feed F_{fuse} into the semantic segmentation and 3D keypoints localization modules to predict the mask and 3D keypoints of the target

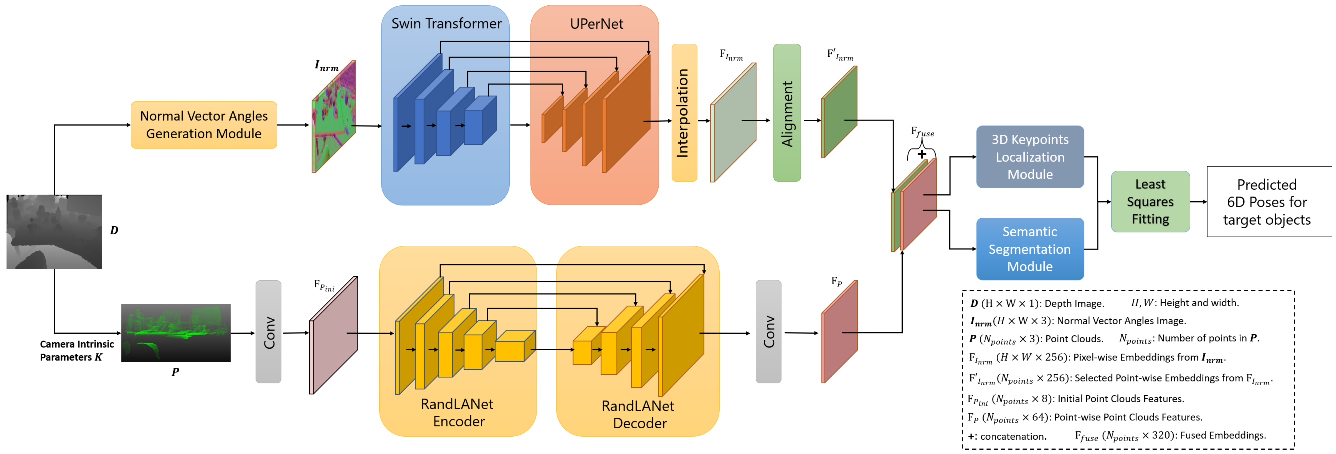

Swin Transformer
UPerNet
I_{nrm}
Normal Vector Angles Generation Module
Interpolation
$F_{I_{nrm}}$
Alignment
$F'_{I_{nrm}}$
F_{fuse}
+
3D Keypoints Localization Module
Semantic Segmentation Module
Least Squares Fitting
Predicted 6D Poses for target objects
D
Camera Intrinsic Parameters K
P
Conv
$F_{P_{ini}}$
RandLANet Encoder
RandLANet Decoder
Conv
F_P
D (H × W × 1): Depth Image.
H, W: Height and width.
I_{nrm}(H × W × 3): Normal Vector Angles Image.
P (N_{points} × 3): Point Clouds.
N_{points}: Number of points in P.
$F_{I_{nrm}}$ ($H \times W \times 256$): Pixel-wise Embeddings from I_{nrm}.
$F'_{I_{nrm}}$(N_{points} × 256): Selected Point-wise Embeddings from $F_{I_{nrm}}$.
$F_{P_{ini}}$ (N_{points} × 8): Initial Point Clouds Features.
F_P (N_{points} × 64): Point-wise Point Clouds Features.
+: concatenation.
F_{fuse} (N_{points} × 320): Fused Embeddings.

Fig. 2: The pipeline of our proposed framework. The normal vector angles generation module outputs the normal vector angles image. Two encoder-decoder networks extract features of normal vector angles images and point clouds, respectively. The extracted image and point clouds embeddings are concatenated and then fed into the semantic segmentation and 3D keypoints localization modules to predict the mask and 3D keypoints for the target object. Finally, a least-squares fitting manner is adopted to estimate 6D poses from the predicted 3D keypoints.

object. Finally, we adopt a least-squares fitting method to estimate the 6D poses based on the predicted 3D keypoints.

B. Normal Vector Angles Image Generation.

Previous depth-based 6D pose estimation studies [18], [10] directly fed depth information into image encoders, but these approaches may not fully utilize the geometry information in depth images. Depth images capture depth information and 2D grid structures and contain implicit local geometric features and directional information.

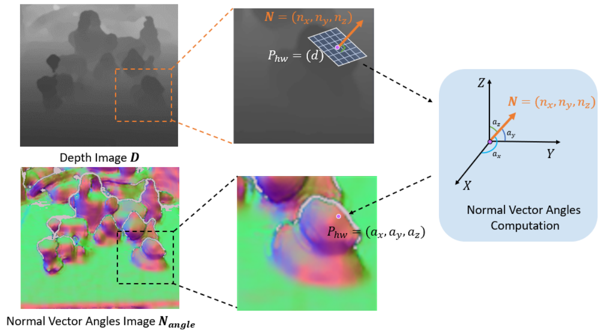

$N = (n_x, n_y, n_z)$
$P_{hw} = (d)$
Depth Image D
Z
$N = (n_x, n_y, n_z)$
a_z
a_y
a_x
Y
X
Normal Vector Angles Computation
$P_{hw} = (a_x, a_y, a_z)$
Normal Vector Angles Image N_{angle}

Fig. 3: Examples of the normal vector angles image generation module. A pixel value of P_{hw} in D is the distance d between an object's surface point and the camera. We compute the normal vector $\mathbf{N}(n_x, n_y, n_z)$ for each surface point based on its depth information, and then obtain the angles (a_x, a_y, a_z) between $\mathbf{N}$ and XYZ axes.

To capture both the local geometric features and directional information from depth images, we generate a *normal vector angles image* for each scene, as shown in Fig. 3. To achieve this, we calculate the surface normal vector $\mathbf{N}(n_x, n_y, n_z)$ for each pixel (u, v) in the depth image D, and then determine the angles between $\mathbf{N}(n_x, n_y, n_z)$ and the XYZ axes in the camera coordinate system. The X-axis, Y-axis, and Z-axis are represented as vectors $\mathbf{x}(1,0,0)$, $\mathbf{y}(0,1,0)$, and $\mathbf{z}(0,0,1)$, respectively. The angles between $\mathbf{N}(n_x, n_y, n_z)$ and these three axes vectors can be expressed as:

$$a_x = \arccos(\mathbf{N} \cdot \mathbf{x}), a_y = \arccos(\mathbf{N} \cdot \mathbf{y}), a_z = \arccos(\mathbf{N} \cdot \mathbf{z}). \quad (1)$$

Finally, the angles (a_x, a_y, a_z) are normalized into the range $0 \sim 255$ and used to create the normal vector angles image I_{nrm}, where each pixel has the normalized (a_x, a_y, a_z) as its value. I_{nrm} consists of three channels, each representing one of the normalized angles.

C. Image Feature Extraction.

We propose an encoder-decoder network to extract pixel-wise embeddings from normal vector angles image I_{nrm}, as shown in the top panel of Fig. 2. The network employs a Tiny Swin Transformer (Swin-T) [13] as the encoder to learn multi-scale representations from I_{nrm}. The Swin-T includes 4 stages of Swin Transformer blocks with modified self-attention layers, linear embedding layers, and patch merging layers. I_{nrm} is fed into the encoder as tokens. In stage 1, the linear embedding layer and a Swin Transformer block with modified self-attention layers extract the features. Within stages 2, 3, and 4, patch merging layers reduce feature map resolutions and Swin Transformer blocks are applied for feature transformation to enlarge feature dimensions. The embeddings from all stages are combined as hierarchical representations and then fed into the decoder network. Compared to the other Swin Transformer models with larger sizes, such as Small, Based, and Large Swin Transformer models, Swin-T has fewer parameters, making it more efficient.

After encoding the normal vector angles image with Swin-T, the network employs UPerNet [23] and bilinear interpolation to generate dense pixel-wise image features. The multi-scale feature maps from Swin-T's are passed through a Pyramid Pooling Module in UPerNet to generate feature maps with the same dimension. These feature maps are resized by a bilinear interpolation process and then concatenated. The dimension of the concatenated features is reduced through a series of CNNs and an additional interpolation stage, resulting in the pixel-wise image embeddings $\mathrm{F}_{I_{nrm}}$, whose size matches the input image size.

D. Point Clouds Feature Extraction.

We begin by converting depth images to point clouds P by the given camera parameters. Next, we adopt RandLA-Net [14] to extract representations from P, as shown in the bottom panel of Fig. 2. Initially, P goes through a series of CNNs to generate initial features $\mathrm{F}_{P_{ini}}$. These initial features are then input to RandLA-Net encoder and decoder. The encoder consists of five stages that extract multi-scale embeddings, while the decoder recovers the resolution of the features in each stage. Finally, we apply additional CNNs to produce the final point-wise point clouds embeddings F_P.

E. 3D Keypoint-based Pose Estimation.

Recently, several works on 6D pose estimation [6], [7] estimate 6D poses by establishing keypoint correspondences between 3D models and input RGB-D images, and then applying a least-squares fitting algorithm to recover 6D poses based on these correspondences. Similar to [6], [7], we follow this way to estimate 6D poses. As shown in Fig. 2, after obtaining pixel-wise image embeddings $\mathrm{F}_{I_{nrm}}$ and point-wise point clouds embeddings F_P, we take advantage of the alignment between the image and the point clouds, and then select the point-wise image features $\mathrm{F}'_{I_{nrm}}$ from $\mathrm{F}_{I_{nrm}}$. Afterward, we concatenate F_P and $\mathrm{F}'_{I_{nrm}}$ to obtain the fused features F_{fuse}, which is fed into the semantic segmentation module and 3D keypoints localization module to predict the mask and detect 3D keypoints of the target object. Finally, we adopt a least-squares fitting manner to estimate the object's pose. The approaches are described in detail as follows:

1) 3D Keypoints Localization: We utilize 3D keypoints of the target object obtained through a keypoint voting module, as in [7]. Specifically, the module takes the concatenated features F_{fuse} as input and predicts offsets of_i^j from each 3D point p_i to the selected 3D keypoint kp_j of the target object. The module is supervised using L1 loss function [7]:

$$L_{\text{keypoints}} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{M}\left\| of_i^j - of_i^{j*}\right\| \mathbb{I}\left(p_i \in I\right), \tag{2}$$

where N is the total number of points on the object's surface, M is the total number of selected target keypoints, of_i^{j*} is the ground truth offset, and $\mathbb{I}$ is an indication function indicating whether point p_i belongs the object. Moreover, to supervise the offset prediction between point p_i and the object's 3D centroid, we use another L1 loss function:

$$L_{\text{centroid}} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\left\|\Delta x_i - \Delta x_i^*\right\| \mathbb{I}\left(p_i \in I\right), \tag{3}$$

where Δx_i is the predicted offset and Δx_i^* is the ground truth offset. Once the offsets are predicted, can be further refined using the MeanShift clustering method [24].

2) Semantic Segmentation: We integrate an instance segmentation module into our pipeline to predict pixel labels and segment the target object from the scene. Unlike previous works [18], [10] that rely on an external segmentation network, such as Mask R-CNN [25], to preprocess the input image, our approach has several benefits. Firstly, it makes the framework more comprehensive and streamlined. Secondly, by forcing the segmentation module to distinguish objects, it facilitates the extraction of both global and local features, benefiting the 3D keypoints localization module. As in [7], we employ the Focal Loss [26] to supervise this module:

$$L_{\text{segment}} = -\alpha\left(1-q_i\right)^{\gamma}\log\left(q_i\right), \tag{4}$$

where $q_i = c_i \cdot l_i$, α represents the α-balance parameter, γ is the focusing parameter, c_i denotes the predicted confidence for the point p_i belonging to each class, and l_i is the one-hot representation of the ground truth class label.

3) Loss Function: Similar to the method of [7], we trained the network in a supervised manner using the following composite loss function:

$$L_{\text{loss}} = \lambda_1 L_{\text{keypoints}} + \lambda_2 L_{\text{segment}} + \lambda_3 L_{\text{centroid}}\ , \tag{5}$$

where λ_1, λ_2, and λ_3 represent the respective weights assigned to each module.

4) Least Squares Fitting: To compute the rotation matrix R and the translation vector T, given the 3D keypoints in the object coordinates system $\{p_i\}_{i=1}^{N}$ and the corresponding 3D keypoints in camera coordinates system $\{c_i\}_{i=1}^{N}$, the lease-squares fitting algorithm [27] minimizes the squared loss:

$$L = \sum\nolimits_{i=1}^{N}\left\|c_i - \left(Rp_i + T\right)\right\|^2. \tag{6}$$

IV. EXPERIMENTS

We conduct experiments on three public benchmark datasets including the LM [28], the O-LM [29], and the YCBV [30]. Compared with SOTA baselines, our proposed SwinDePose outperforms them on LM and O-LM datasets. We are also presenting ablation studies to demonstrate the effectiveness of the components in SwinDePose.

A. Experimental Setup.

Datasets. The LM is wildly used in 6D pose estimation. It contains RGB-D images and 13 indoor texture-less objects in cluttered scenes. Following [6], we split the training and testing sets and generated 20K synthetic depth images for each category in the LM.

The O-LM is a subset of the LM and contains 8 objects in the LM. Each scene in the O-LM has heavy occlusions, making the task more challenging. We follow [6] to split the training and testing sets in the O-LM.

The YCBV contains 21 YCB objects and 92 RGB-D videos. We followed [30] to split the training and testing set. The training set contains real and synthetic images.

Evaluation Metrics. We use ADD and ADDS metrics to evaluate the models' performance for asymmetric and symmetric objects, following [30]. For asymmetric objects, ADD computes the mean distance between two transformed

3D CAD model points using the estimated pose and the ground truth pose, defined as follows:

$$\mathrm{ADD} = \frac{1}{m}\sum\nolimits_{x\in M}\|(Rx+T)-(\tilde{R}x+\tilde{T})\|, \tag{3}$$

where M denotes the set of 3D CAD model points and m is the number of points. R and T are the predicted rotation and translation, respectively. $\tilde{R}$ and $\tilde{T}$ are ground truth rotation and translation. $||.||$ denotes the Euclidean norm.

For rotational symmetric objects, we compute the ADDS, which is the mean distance based on the closest point distance between two transformed 3D CAD model points:

$$\mathrm{ADDS} = \frac{1}{m}\sum\nolimits_{x_1\in M}\min\nolimits_{x_2\in M}\left\|(Rx_1+T)-\left(\tilde{R}x_2+\tilde{T}\right)\right\|. \tag{4}$$

In experiments, we report the accuracy in terms of ADD or ADDS less than 10% of object diameter as in [6], [28].

B. Implementation Details.

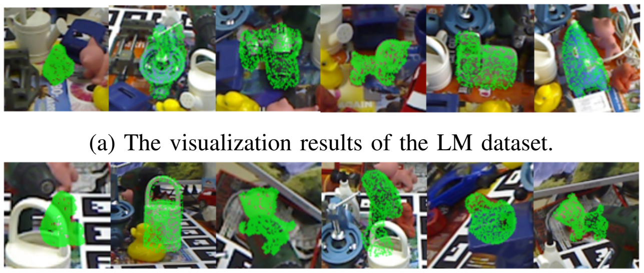
Network Architecture. SwinDePose utilizes encoder-decoder modules to extract features from normal vector angle images and point clouds. To learn image embeddings, Swin-T [13] serves as an encoder, generating feature maps F_{swin} at 4 various dimensions and resolutions. UPerNet then decodes and interpolates the multi-scale feature map F_{swin} to obtain the feature map F_{intep}. After concatenating all features from F_{intep}, CNNs are used to reduce its dimensionality. An additional interpolation process expands F_{concat}, resulting in the dense pixel-wise image embeddings F_{Inrm}. For point cloud feature extraction, depth images are converted to point clouds, and 12288 points are sampled following [6]. RandLANet is employed to extract point-wise features with a dimension of 64. With these settings, the model consists of approximately 37M parameters. The average inference time per frame is approximately 2.099 s, and the average time for generating a normal angle image is 0.0027 s when using a single NVIDIA GPU Tesla V100.

Keypoint Detection. We conduct the SIFT-FPS keypoints selection algorithm from [6] by detecting 2D keypoints in images using SIFT, then projecting 2D keypoints to 3D in the object coordinates system, and finally applying the FPS algorithm to choose N points from them.

C. Comparison with State-of-the-Art Methods.

Evaluation on the LM Dataset. Our proposed SwinDePose has been evaluated against state-of-the-art 6D pose estimation methods on the LM dataset. We categorize the selected methods into three groups based on their input types: RGB, RGB-D, and Depth-Only. Table I presents the accuracy comparison between SwinDePose and others without any post-processing refinement. The results demonstrate that without the ground-truth (GT) mask, our SwinDePose outperforms the previous best-performed depth-based baseline, CATRE [10] by 5.94% on the accuracy of ADD(S) metric. With the GT masks, our SwinDePose still outperforms the previous best baseline, OVE6D (with GT masks) by 1.14%.

Moreover, our proposed method outperforms some RGB-based or RGB-D based methods on certain categories, even without GT masks and only with geometry information as input. This highlights the effectiveness of our method in leveraging the geometry information from depth images for 6D object pose estimation. Qualitative results showing the performance of SwinDePose are displayed in Fig. 4a, which demonstrate that our proposed model can accurately predict 6D object poses.



(a) The visualization results of the LM dataset.

(b) The visualization results of the O-LM dataset.

Fig. 4: Qualitative evaluation of SwinDePose on the LM and the O-LM datasets. 3D surface points of the object meshes are transformed by the predicted poses and projected onto 2D images by intrinsic parameters. For better visualizations, we display projected points in green.

Evaluation on the O-LM Dataset. For the O-LM dataset, the results in Table II indicate that SwinDePose outperforms the best-performing depth-based baseline, which is CloudAAE, by 2.38% without GT masks. With GT masks, SwinDePose achieves an 8.74% higher accuracy on ADD(S) compared to OVE6D. This improvement is observed without any post-processing refinement, highlighting the robustness of our model. Furthermore, SwinDePose exceeds some RGB-based or RGB-D based methods on certain categories even without GT masks. Due to heavy occlusions, GT masks significantly improve our performance since our semantic segmentation module faces difficulty in segmenting objects. Qualitative results are displayed in Fig. 4b, showing accurate 6D pose estimation results even in scenes with heavy occlusion, as demonstrated in the images.

Evaluation on the YCBV Dataset. In the YCBV dataset, the findings in Table III demonstrate that our method performs slightly worse than CloudAAE, which utilizes ICP as post-processing. This observation highlights the robustness of our model, as it achieves competitive results without any post-processing techniques. Moreover, our method outperforms certain RGB-based approaches with GT masks.

D. Ablation Study.

We performed extensive ablation studies on our model design using the LM dataset. The results shown in Table IV are the average ADD(S) on the LM dataset.

Effect of the Images Feature Extraction. To validate the effectiveness of using Swin-T for learning image embeddings, we conducted an ablation study in top of Table IV. Initially, the scenario was tested without employing any module for image feature extraction (w/o Any Encoder). Then, 4-layer CNNs were added to extract image features (w Conv Encoder). Finally, the CNNs were replaced with

TABLE I: The accuracy in terms of ADD(S) results for the LM dataset. Symmetric objects are noted with *. We highlight the best performance in bold for each group.

INPUTS	RGB			RGB-D			Depth-Only					
METHODS	PVNet [4]	Pix2Pose [5]	RNNPose [31]	PVN3D [7]	DenseFusion [15]	KPD [32]	CloudAAE [8]	CATRE [10]	OVE6D [18] (w Mask R-CNN)	OVE6D [18] (w GT Masks)	OURS (w/o GT Mask)	OURS (w GT Masks)
ape	43.6	58.1	**88.2**	**97.3**	92	94.2	74.5	63.7	-	-	91.7	**95.4**
benchvise	**99.9**	91.0	79.7	**99.7**	93	98.2	86.6	**98.6**	-	-	97.9	98.2
camera	86.9	60.9	**98.0**	**99.6**	94	98.5	65.6	89.7	-	-	94.8	**96.9**
can	95.5	84.4	**99.3**	**99.5**	93	94.0	90.2	96.1	-	-	97.6	**98.2**
cat	79.3	65.0	**96.4**	**99.8**	97	92.0	90.7	84.3	-	-	98.3	**98.6**
driller	96.4	76.3	**99.7**	**99.3**	87	97.2	97.3	**98.6**	-	-	**98.6**	98.5
duck	52.6	43.8	**89.3**	**98.2**	92	91.5	50.0	63.9	-	-	88.5	**92.7**
eggbox*	99.2	96.8	**99.5**	99.8	**100**	99.6	99.7	99.8	-	-	**100.0**	**100.0**
glue*	95.7	79.4	**99.7**	**100.0**	**100**	92.5	93.5	99.4	-	-	98.6	**100.0**
holepuncher	82.0	74.8	**97.4**	**99.9**	92	92.1	57.9	93.2	-	-	92.4	**93.6**
iron	98.8	83.1	**100.0**	**99.7**	97	98.7	85.0	**98.4**	-	-	96.9	96.9
lamp	99.3	82.0	**99.8**	**99.8**	95	96.5	82.1	98.7	-	-	98.8	**99.1**
phone	92.4	45.0	**98.4**	**99.5**	93	97.2	94.4	97.5	-	-	98.3	**98.8**
MEAN	86.3	72.4	**97.4**	**99.4**	94	95.6	82.1	90.9	86.1	96.4	96.3	**97.5**

TABLE II: The accuracy in terms of ADD(S) results for the O-LM dataset. Symmetric objects are noted with *. We highlight the best performance in bold for each group.

INPUTS	RGB			RGB-D			Depth-Only				
METHODS	PVNet [4]	Pix2Pose [5]	Keypoint [33]	Point-to-Keypoint [34]	FFB6D [6]	KPD [32]	CloudAAE [8]	OVE6D [18] (w Mask R-CNN)	OVE6D [18] (w GT Masks)	OURS (w/o GT Mask)	OURS (w GT Masks)
ape	15.8	**22.0**	-	**51.6**	47.2	19.5	-	-	-	50.3	**59.8**
can	**63.3**	44.7	-	75.6	**85.2**	78.4	-	-	-	84.1	**88.8**
cat	16.7	**22.7**	-	28.7	**45.7**	28.2	-	-	-	39.0	**46.7**
driller	25.2	**44.7**	-	66.9	**81.4**	75.1	-	-	-	88.9	**95.1**
duck	**65.7**	15.0	-	36.7	**53.9**	38.6	-	-	-	53.0	**59.4**
eggbox*	**50.2**	25.2	-	47.1	**70.2**	51.2	-	-	-	28.5	**90.3**
glue*	**49.6**	32.4	-	**71.9**	60.1	52.1	-	-	-	58.4	**88.0**
holepuncher	39.7	**49.5**	-	45.7	**85.9**	59.0	-	-	-	79.8	**88.8**
MEAN	**40.8**	32.0	33.7	52.6	**66.2**	50.3	58.9	56.1	70.9	60.3	**77.1**

TABLE III: The average accuracy in terms of ADD(S) results for the YCBV dataset. We highlight the best performance in bold for each group.

INPUTS	RGB				RGB-D				Depth-Only				
METHODS	PoseCNN [30]		PVNet [4]	EPOS [35]	PVN3D [7]		DenseFusion [15]		CloudAAE [8] (w ICP)	OURS (w/o GT Mask)		OURS (w GT Masks)	
METRICS	ADD	ADDS	ADDS	ADD	ADD	ADDS	ADD	ADDS	ADD	ADD	ADDS	ADD	ADDS
MEAN	75.2	61.3	73.4	**78.3**	**95.4**	92.6	90.9	83.9	**93.5**	71.8	89.4	73.1	91.4

TABLE IV: Results of ablation study. We validate the efficacy of different aspects of SwinDePose.

Aspect	Description	Average ADD(S) w/o GT Masks	Average ADD(S) w GT Masks
Images Feature Extraction	w/o Any Encoder	42.8	51.6
	w Conv Encoder	94.3	95.7
Point Clouds Feature Extraction	w/o Any Encoder	51.4	59.3
	w PointNet	80.4	90.1
Normal Vector Angles Images Replacement	w Depth Images	94.1	95.7
	w Normal Vectors	81.2	86.2
Full Model	SwinDePose	96.3	97.5

a Swin-T module for image feature extraction. The results show the difficulty in accurately estimating 6D poses solely based on point cloud embeddings. The use of CNNs significantly improved the performance of pose estimation, while the Swin-T model produced slightly more accurate results on the LM dataset. However, when we evaluated the performance of CNNs on the O-LM dataset, the average ADD(S) score obtained was only 53.03, which is significantly worse than the performance of Swin-T (60.3). This highlights the superior robustness to occlusion achieved by incorporating the Swin-T module.

Effect of the Point Clouds Feature Extraction. To validate the effectiveness of RandLA-Net for learning point cloud embedding, firstly, we employed no module for point cloud feature extraction (w/o Any Encoder). Then, we added PointNet [36] (w PointNet) to extract features. Lastly, we replaced PointNet with RandLA-Net. The results in Table IV show that removing the point cloud embeddings significantly harms the performance. Additionally, PointNet demonstrates its capability in enhancing pose estimation performance. Moreover, the substitution of PointNet with RandLA-Net yields superior accuracy in pose estimation. We hypothesize that the randomized aggregation method in RandLA-Net allows for more robust feature learning and is less sensitive to individual point errors than PointNet.

Effect of the Normal Vector Angles Image. To validate the effectiveness of normal vector angles images, initially, depth images were fed into SwinDePose, which were subsequently replaced by normal vectors. Finally, we changed them into normal vector angles images. The results reveal that neither using depth images nor normal vectors surpasses using normal vector angles images. This finding demonstrates the usage of normal vector angle images is beneficial in improving pose estimation accuracy.

V. CONCLUSIONS

We introduce SwinDePose, a novel fusion network for learning representations from a single depth image that maximizes the information present in the scene for 6D pose estimation. We developed an effective module that converts depth images into normal vector angles images, explicitly incorporating more geometric information into the fusion network. Our approach achieves superior results on the LM, and O-LM datasets, while being competitive on YCBV. Furthermore, our proposed fusion network can be applied to 6D pose estimation based on RGB-D images, and we anticipate further research in this area.

Acknowledgement

This work was partially supported by NSF Award CNS1625843. We would also like to thank the insightful suggestions of Shuo Zhang in manuscript refinement, code acceleration, and model construction.

References

[1] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED Framework: Object Recognition And Pose Estimation For Manipulation. *The international journal of robotics research*, 30(10):1284–1306, 2011.

[2] Mary B Alatise and Gerhard P Hancke. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access*, 8:39830–39846, 2020.

[3] Ying Kin Yu, Kin Hong Wong, and Michael Ming Yuen Chang. Pose Estimation for Augmented Reality Applications using Genetic Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1295–1301, 2005.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Ge Gao, Mikko Lauri, Yulong Wang, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. 6D Object Pose Regression via Supervised Learning on Point Clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3643–3649. IEEE, 2020.

[10] Xingyu Liu, Gu Wang, Yi Li, and Xiangyang Ji. CATRE: Iterative Point Clouds Alignment for Category-Level Object Pose Refinement. In *European Conference on Computer Vision*, pages 499–516. Springer, 2022.

[11] Stefan Hinterstoisser, Vincent Lepetit, Naresh Rajkumar, and Kurt Konolige. Going Further with Point Pair Features. In *European conference on computer vision*, pages 834–848. Springer, 2016.

[12] Somar Boubou, Tatsuo Narikiyo, and Michihiro Kawanishi. Differential Histogram of Normal Vectors for Object Recognition with Depth Sensors. In *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 162–167. IEEE, 2016.

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[14] 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.

[15] 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.

[16] Xiaoke Shen and Ioannis Stamos. simcrosstrans: A simple cross-modality transfer learning for object detection with convnets or vision transformers. *arXiv preprint arXiv:2203.10456*, 2022.

[17] Jiayi Ma, Linfeng Tang, Fan Fan, Jun Huang, Xiaoguang Mei, and Yong Ma. SwinFusion: Cross-domain long-range learning for general image fusion via swin transformer. *IEEE/CAA Journal of Automatica Sinica*, 9(7):1200–1217, 2022.

[18] 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.

[19] Apoorva Beedu, Huda Alamri, and Irfan Essa. Video based Object 6D Pose Estimation using Transformers. *arXiv preprint arXiv:2210.13540*, 2022.

[20] Baba C Vemuri, Amar Mitiche, and Jake K Aggarwal. Curvature-Based Representation of Objects from Range Data. *Image and vision computing*, 4(2):107–114, 1986.

[21] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Depth Kernel Descriptors For Object Recognition. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 821–826. IEEE, 2011.

[22] Allan Zelener and Ioannis Stamos. CNN-Based Object Segmentation in Urban Lidar With Missing Points. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 417–425. IEEE, 2016.

[23] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified Perceptual Parsing for Scene Understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

[24] Dorin Comaniciu and Peter Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss For Dense Object Detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[27] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-Squares Fitting of Two 3D Point Sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987.

[28] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal Templates For Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[29] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *European conference on computer vision*, pages 536–551. Springer, 2014.

[30] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. 2018.

[31] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. RNNPose: Recurrent 6-DoF Object Pose Refinement with Robust Correspondence Field Estimation and Pose Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14880–14890, 2022.

[32] Lounes Saadi, Bassem Besbes, Sebastien Kramm, and Abdelaziz Bensrhair. Optimizing RGB-D Fusion For Accurate 6DoF Pose Estimation. *IEEE Robotics and Automation Letters*, 6(2):2413–2420, 2021.

[33] Shaobo Zhang, Wanqing Zhao, Ziyu Guan, Xianlin Peng, and Jinye Peng. Keypoint-Graph-Driven Learning Framework for Object Pose Estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1065–1073, 2021.

[34] Weitong Hua, Jiaxin Guo, Yue Wang, and Rong Xiong. 3D Point-To-Keypoint Voting Network For 6D Pose Estimation. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 536–541. IEEE, 2020.

[35] 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.

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017.