DGECN++: A Depth-Guided Edge Convolutional Network for End-to-end 6D Pose Estimation via Attention Mechanism

Tuo Cao, Wenxiao Zhang, Yanping Fu, Shengjie Zheng, Fei Luo* and Chunxia Xiao* *Senior Member, IEEE*

***Abstract*—Monocular object 6D pose estimation is a fundamental yet challenging task in computer vision. Recently, deep learning has been proven to be capable of predicting remarkable results in this task. Existing works often adopt a two-stage pipeline with establishing 2D-3D correspondences and utilizing a PnP/RANSAC or differentiable PnP algorithm to recover 6 degrees-of-freedom (6DoF) pose parameters. However, most of them hardly consider the geometric features in 3D space, and ignore the topological cues when performing differentiable PnP algorithms. To this end, we present an improved end-to-end monocular 6D pose estimation method (DGECN++) that incorporates depth estimation and a geometric-aware learnable PnP network. Our method is based on keypoints. First we detect the 2D keypoints that correspond to the 3D model. We then integrate differentiable PnP/RANSAC algorithm to create an end-to-end pipeline for 6D pose estimation. We focuses on the following three key aspects: 1) We utilize the estimated depth information to guide the process of extracting 2D-3D correspondences and refine the results using a cascaded differentiable PnP/RANSAC algorithm that incorporates geometric information. 2) We leverage the uncertainty of the estimated depth map to enhance the accuracy and robustness of the predicted 6D pose. 3) We propose a differentiable Perspective-n-Point (PnP) algorithm based on edge convolution and self-attention to explore the topological relationships between 2D-3D correspondences. Experimental results demonstrate that our proposed network surpasses existing methods in terms of both effectiveness and efficiency.**

***Index Terms*—6D pose estimation, graph CNN, end-to-end network, attention mechanism.**

I. Introduction

OBJECT pose estimation is an important task in computer vision. It involves estimating the 6 degrees of freedom (6DoF or 6D) parameters for the location and orientation of an object in an image or a series of video frames. It has wide applications in AR [1]–[4], robotic vision [5]–[7] and 3D reconstruction [8], [9]. In additional, [7] leverages the aircraft's inherent rigid structural characteristics and incorporates arrow-like directional properties to construct a 3D skeleton model for aircraft that possesses reconstruction capabilities, simplicity, and directional attributes. Due to the influence of various factors, including noises, occlusion, and illumination variations, accurate 6D pose estimation is still challenging. 6D pose estimation can rely on RGB image [1], [2], [5], [6], [10], [11] or RGB image accompanying a depth image [12]–[15]. The regression-based estimation and the keypoints-based estimation are two major strategies for 6D pose estimation. Traditional regression-based methods mainly used the template matching technique [16]–[18]. CNN networks have shown significant robustness to environmental variations. Some methods [1], [19], [20] introduced the CNN network to directly regress the 6D pose parameters from a single RGB image. [15] employ data augmentation techniques to enhance the practical applicability of network in real-world scenarios. By exclusively training on synthetic data, this approach reduces the need for labor-intensive manual data annotation efforts. The keypoints-based methods usually consist of two stages. The first stage predicts the 2D locations of the the 3D model keypoints in RGB images. Then, the second stage predicts the 6D pose parameters by the Random Sample Consensus (RANSAC) based Perspective-n-Point (PnP) method from the 2D-3D correspondences.

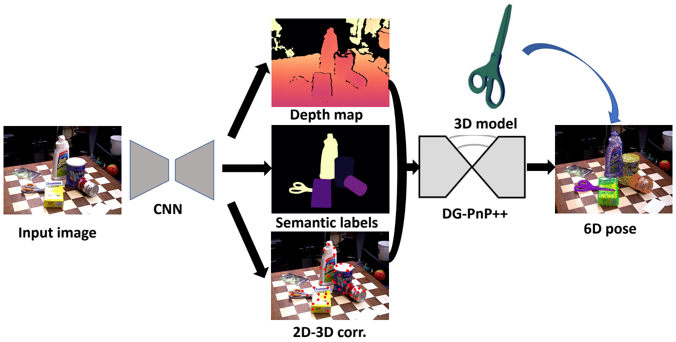


Fig. 1. **Pipeline of our DGECN++**. Given an RGB image as input, our DGECN++ performs simultaneous segmentation and depth map prediction. Once the 2D-3D correspondences are established, we replace the traditional RANSAC/PnP algorithm with a learnable DG-PnP module to accurately regress the 6D pose.

Although many representative works [21]–[25] have proven the validity of the two-stage pipeline, there are still some limitations on it. Firstly, RANSAC-based PnP is very time-cost when the 2D-3D correspondences are dense. Secondly, most two-stage neural networks do not optimize the loss functions

This work was supported by the National Natural Science Foundation of China under Grant (No. 61972298 and No. 62372336) and CAAI-Huawei MindSpore Open Fund. (*Corresponding Author: Chunxia Xiao and Fei Luo*)

Tuo Cao, Wenxiao Zhang, Shengjie Zheng, Fei Luo and Chunxia Xiao are with School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: {maplect, 2020282110184, luofei, cxxiao}@whu.edu.cn, wenxxiao.zhang@gmail.com)

Yanping Fu is with School of Computer Science and Technology, Anhui University, Anhui 230039, China (e-mail: ypfu@ahu.edu.cn)

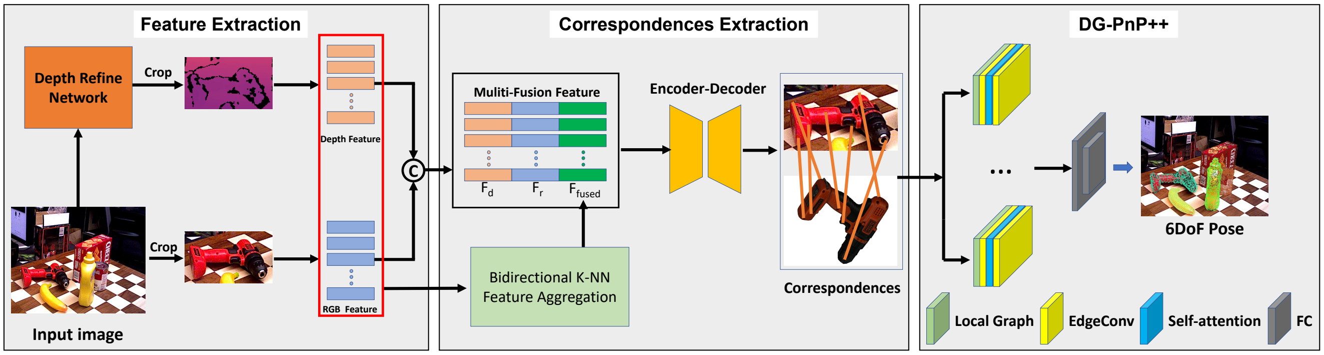


Fig. 2. **Overview of our DGECN++ architecture.** Our framework comprises three main components: 1) a feature extraction network that combines depth and RGB features, 2) a deep voting-based network for extracting 2D-3D correspondences, and 3) a learnable PnP network called DG-PnP++ for estimating the 6D pose of objects. Here F_r, F_d, and F_{fused} represent RGB features, depth features, and local features, respectively.

directly for the ultimate 6D estimation. Thus they are not trained in an end-to-end manner. Finally, the separation of the two stages may accumulate significant errors and decrease the ultimate estimation effect.

To overcome the above limitations, we propose a Depth-Guided Edge Convolutional Network via attention mechanism (DGECN++) to jointly tackle 2D-3D correspondence extraction and 6D pose estimation. In DGECN++ network, we design a depth-guided approach to leverage the geometric constraints of rigid objects to effectively establish 2D-3D correspondences, and a novel Dynamic Graph PnP (DG-PnP++) to evaluate the properties of the correspondence set and discover its potential for dealing with complex textures.

A preliminary version of this work (DGECN) has been published in CVPR 2022 [26]. In this paper, we have made several significant improvements to enhance the performance of our original method. Firstly, we have incorporated a more effective fusion module that combines depth and texture features, leading to a notable improvement in accuracy. Secondly, we have introduced several important modifications to the DGECN framework to enhance the robustness of the learning process. Specifically, we have extended the DG-PnP module into an attention-based Dynamic Graph Edge Convolutional network, which greatly reduces the influence of noise points. Additionally, we have introduced a novel self-attention guideline to handle depth estimation in uncertain areas, further strengthening the robustness of our approach. Through extensive experiments, we demonstrate that these contributions significantly enhance the capability of extracting geometric features from monocular images compared to the original DGECN. Based on the fundamental architecture of DGECN and further improvements made in this work, the whole work becomes a new method, so we call it as DGECN++.

The contributions are reformulated as follows:

- We introduce a depth-guided network that enables direct learning of the 6D pose from a monocular image, without needing extra information. Additionally, we propose a Depth Refinement Network (DRN) to enhance the quality of the estimated depth map.
- We investigate the characteristics of 2D-3D correspondence sets and unveil that constructing a graph from the distributions of 2D keypoints assists in the learning of 6D pose parameters. Additionally, we introduce a simple but effective learnable PnP network for directly predicting the 6D pose from 2D-3D correspondences.
- We integrate attention mechanism into the 6D pose estimation where the depth feature and pose parameters can be jointly optimized with the depth map, uncertainty map and 2D-3D matching.

Particularly, we have not only significantly improved our baseline in on common benchmark datasets including LINEMOD, YCB-Video and Occluded LINEMOD, but also added the new evaluation for HomebrewedDB. Our DGECN++ is able to produce more reliable results under challenging scenes, *i.e.* Occluded LINEMOD and YCB-Video, which have many symmetric objects and significant occlusion. Moreover, our method can be directly applied to RGB-D based 6D pose estimation works just through replacing depth estimation module to the corresponding depth maps.

II. Related work

A. RGB based 6D Pose Estimation

As deep learning shows strong ability in object detection, recognition and other fields, many CNN 6D pose estimation based methods have emerged. These methods can be roughly divided into two classes, direct methods and correspondence-based methods. Direct methods usually directly estimate the 6D pose in a single shot. Kendall.*et al.* [20] first introduced CNN into this field, where they employed a network based on GoogleNet [27] to directly learn the 6D camera pose. This problem is still challenging due to the variety of objects as well as the complexity of a scene caused by clutters and occlusions between objects. To address this issue, PoseCNN [1] estimated the 3D translation of an object by localizing its center in the image and predicting its distance from the camera. However, this problem is still difficult due to the non-closed property to addition of rotation matrix. Some works [19], [28] utilized the $\mathbb{SO}(3) or \mathbb{SE}(3)$ to make the rotation space differentiable. Instead, correspondence-based methods find correspondence between image plane and 3D space and recover 6D poses by RANSAC-based Perspective-n-Point(PnP). PVNet [2] and Seg-Driven [10] conducted segmentation coupled with voting for each correspondence to make the estimation more robust.

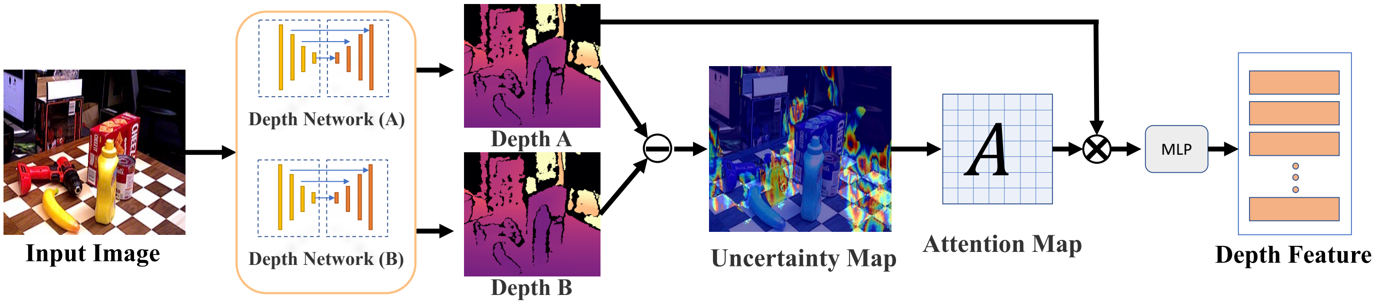


Fig. 3. **Depth Refinement Network.** This module consists of two separate depth estimation networks that produce depth maps D_A and D_B respectively. We calculate the disparity between these depth maps and identify regions where the disparity exceeds a predefined threshold as uncertain areas. To address the uncertainty in these regions, we employ a cross-attention mechanism and generate depth features.($\otimes, \ominus$) are matrix multiplication and subtraction, respectively.

EPOS [21] made use of surface fragments to account for ambiguities in pose estimation. Pix2Pose [6] used a network based on GAN to predict the 3D coordinates of each object pixel without textured models. Oberwegeret *al.* [29] output pixel-wise heatmaps of keypoints to address the issue of occlusion. NVR-Net [11] introduced a novel pose representation that effectively disentangles rotations from translations, enabling robust pose prediction through a Convolutional Neural Network (CNN). Additionally, it incorporates viewpoint rectification techniques to mitigate ambiguity in pose estimation. TexPose [30] introduced a surfel-conditioned adversarial training loss and incorporated a synthetic texture regularization term. These components were designed to effectively address pose errors and mitigate segmentation imperfections during the texture learning process.

B. RGB-D based 6D Pose Estimation

Recently, consumer-level RGB-D cameras have shown advancements in autonomous driving, AR/VR, and 3D reconstruction by providing additional depth information. Many approaches are now attempting to leverage this distance information to enhance performance in challenging scenarios, including poor lighting conditions, heavy occlusion, and weak texture scenes.

Deep learning-based approaches have been exploring its potential in this field, by incorporating depth information as an additional input. For example, MCN [31] and DenseFusion [14] combine 3D point and 2D information into a color-depth feature, enabling the learning of 6D pose from this latent space. PVN3D [12] extends PVNet [2] to include 3D keypoints and leverage depth information to exploit geometric constraints of rigid objects. PR-GCN [32] enhances depth representation using a point refinement module to estimate 6D pose. In contrast to the aforementioned methods, our approach is based on RGB images, where we learn depth values to extract geometric features without relying on real depth labels. HFF6D [15] introduces an innovative Subtraction Feature Fusion (SFF) module, which incorporates an attention mechanism to exploit feature subtraction during fusion. This module explicitly emphasizes the feature distinctions between consecutive frames, thereby enhancing the reliability of relative pose estimation in complex and challenging scenarios. Hai *et al.* [33] computed a pose-induced flow based on the displacement of 2D reprojection between the initial pose and the currently estimated pose, which embeds the target's 3D shape implicitly.

C. Correspondence Learning in 6D Pose Estimation

To address the limitations of surrogate correspondence learning, researchers have proposed end-to-end approaches that enable gradient backpropagation from pose estimation to intermediate representations. One such approach is the dense correspondence network proposed by Brachmann and Rother [34], which incorporates learnable 3D points, BPnP [35] for predicting 2D keypoint locations, and BlindPnP [36] for learning the weight matrix associated with unordered 2D/3D point sets. It is important to note that these methods require surrogate regularization loss to ensure convergence, as the optimal pose estimation involves numerical instability and non-differentiable operations. Within the probabilistic framework, these methods can be seen as a Laplace approximation approach.

In addition to point correspondence, the RePOSE [37] method introduces a feature-metric correspondence network trained by backpropagating the PnP solver, such as Levenberg-Marquardt. While this approach serves as a local regularization technique in our framework, it is insufficient in addressing pose ambiguity. EPro-PnP [38] introduced a probabilistic PnP layer for pose estimation, which produces a distribution of poses with differentiable probability density on the $\mathbb{SE}(3)$ manifold. The intermediate variables in this layer, including the 2D-3D coordinates and their corresponding weights, are learned by minimizing the KL divergence between the predicted and target pose distributions.

Overall, these end-to-end approaches provide valuable insights and advancements in handling surrogate correspondence learning, enabling gradient-based optimization and enhancing pose estimation in various applications.

D. Graph Convolution Network (GCN)

Owing to the higher representation power enabled by graph structures, Graph Convolutional Networks (GCN) have shown remarkable performance in various tasks such as image captioning [39], text-to-image generation, and human pose estimation [40]. In the field of 3D computer vision, Wald et al [41]. introduced a pioneering learning method that generated a semantic scene graph from a 3D point cloud. DGCNN [42] utilized a GCN-based network for extracting features from point clouds. Another notable work is Superglue [43], which employed GCN to match two sets of local features by simultaneously searching for correspondences and filtering out non-matchable points. GCNs typically perform message passing or information propagation between nodes. At each layer, a node aggregates information from its neighbors, and this aggregated information is then used to update the node's representation. Different from traditional GCNs, we construct a local graph from a set of 2D corresponding points using the K-nearest neighbors (K-NN) algorithm. Then, we perform custom edge convolution operations on the local graph to regress poses.

III. Methodology

In this section, we present our depth-guided 6D pose regression network, which aims to estimate the 6D pose from monocular images. The overview of our method is illustrated in Figure 2. Traditional keypoint localization adopts a voting-based architecture that does not fully leverage depth information. To address this limitation, we focus on three directions to enhance this strategy:

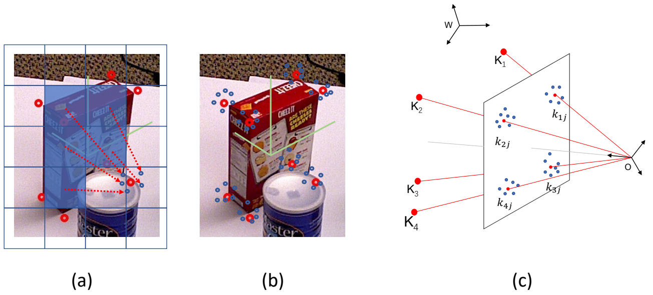
1) We exploit the uncertainty of the estimated depth map in scenes involving 6D object estimation. By refining the depth map, we mitigate the impact of noise during the depth estimation process.
2) Before directly inputting RGB images into the CNN for establishing 2D-3D correspondences, we first predict the depth map and introduce a Bidirectional K-NN Feature Aggregation (BKFA) block to fuse features across different domains.
3) We introduce a learnable DG-PnP++ module to replace the conventional RANSAC/PnP in the two-stage 6D pose estimation pipeline.

We begin with providing the necessary background information. Subsequently, we describe our network architecture, which incorporates depth information to enhance the accuracy of 6D pose estimation.

A. Problem Formulation

Given an image I, our objective is to detect objects and estimate their 6D pose. Specifically, we aim to estimate the rotation $\mathbf{R} \in \mathbb{SO}(3)$ and translation $\mathbf{t} = (t_x, t_y, t_z) \in \mathbb{R}^3$ that transform the object from its object world coordinate system to the camera world coordinate system.

Figure 2 provides an overview of our proposed method. We start by learning depth information using an unsupervised depth estimation network. Similar to the methods GDR-Net [44] and PVNet [2], we locate each object in the image using the FCN method [45] for segmentation. Based on the segmentation results, we crop the region of interest from the depth map and RGB image. These cropped regions are then fed into the Bidirectional K-NN based Feature Aggregation (BKFA) module to extract local features. Simultaneously, we employ ResNet50 [46] to extract 2D features from the image.



(a) (b) (c)

Fig. 4. **2D-3D correspondences**. **(a)** Each semantic grid cell predicts the 2D keypoints' locations corresponding to the object to which it pertains. **(b)** Ground truth 2D correspondences (shown in red) along with their corresponding hypotheses (shown in blue). **(c)** Projections of 2D correspondences onto the camera plane. The camera and object coordinate systems are represented by O and W, respectively.

To incorporate appearance features, geometry information, and local features, we utilize a dense fusion module. This module performs fusion and combines these different types of features. Subsequently, the fused feature is fed into a 2D-3D correspondences prediction network to establish the 2D-3D correspondences. Finally, we utilize our proposed differentiable DG-PnP to directly regress the associated 6D object pose from the established 2D-3D correspondences.

Our framework is a keypoint-based method. Given an image I and a set of 3D models $M = \{M_i | i = 1, ..., N\}$, our task is to estimate the unknown rigid transformation $\{\mathbf{R}, \mathbf{t}\}$. For ease of presentation, we assume there is a single target object in the image, which we denote as O. As illustrated in Figure 4, our goal is to predict the potential 2D locations in I corresponding to the 3D keypoints of the model M. Subsequently, we aim to recover the 6D pose parameters from these correspondences using a network.

B. Depth-Guided Edge Convolutional Network

Inspired by recent works [12], [14], [47], [48] that utilize RGB-D data and point clouds, we incorporate depth information to enhance the robustness and accuracy of 2D-3D correspondences. However, these methods typically require LIDAR or other sensors to obtain precise depth information. In addition, obtaining accurate depth information from a pre-captured RGB image is often challenging. To address this, we employ a network to predict depth as an additional feature to guide the estimation of 2D-3D correspondences. With the advancements in monocular depth estimation, several methods [49]–[51] have emerged. However, these methods are primarily designed for estimating depth in large-scale scenes and may not be directly suitable for estimating depth maps in the context of 6D pose estimation. Therefore, in our approach, We start by pretraining two depth estimation networks, determining the uncertain regions by comparing the errors of the two generated depth maps. We then utilize a self-attention mechanism to optimize the estimated depth map within these uncertain regions.

As illustrated in Fig 5, the DRN consists of two distinct depth estimation networks, which generate depth maps D_A

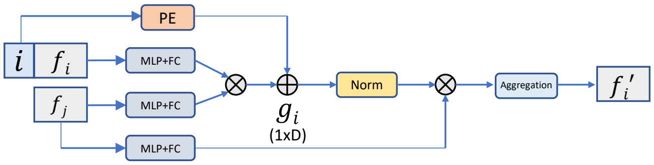


Fig. 5. **Detail of the self-attention feature enhance layer**.

and D_B respectively. We then compute the difference between these depth maps and identify regions where the difference exceeds a certain threshold as uncertain areas. We address the uncertainty in certain areas using a cross-attention mechanism, where we consider the Depth as the *query* and the Uncertainty Map (UM) as the *key* and *value* in the multi-head cross-attention module. The computation of the attention map can be represented as follows:

$$A^{(m)} = \text{Norm}(\Phi_1(\text{Depth})\Phi_2^T(\text{UM})), \tag{1}$$

where $A^{(m)}$ denotes the attention map in the m-th head. The function $Norm$ represents the normalization function, and $\Phi()$ corresponds to the MLP. The multi-head cross-attention can then be expressed as:

$$D = [A^{(1)}\Phi_3(\text{Depth}); ...; A^{(m)}\Phi_3(\text{Depth})]W. \tag{2}$$

In the above equation, D represents the result of the multi-head cross-attention, and W is a learnable parameter. Consequently, D can be utilized to generate the depth feature F_d by adding it to $\Phi_3(\text{Depth})$:

$$F_d = \Phi_3(\text{Depth}) + D. \tag{3}$$

*1) **Feature extraction**:* This stage consists of two streams: one for depth feature extraction and the other for color feature extraction. Previous works [1], [2], [10], [14] have addressed multiple object segmentation by employing existing detection or semantic segmentation algorithms. Similarly, we adopt FCN [45] for segmenting the input image. We use ResNet50 for color embedding and regarding 3D feature extraction, DGECN [26] introduce KFA module for more sufficient RGB-D fusion,. By incorporating both local and global information from appearance and geometry features, we can achieve improved feature representations. To accomplish this, we introduce the Bidirectional KFA module as an extension of the KFA module. Let p_i represent a pixel in the RGB image, and $D_i = \{d_j|j = 1...k\}$ denote the depth set of the k-nearest neighbors of p_i. We utilize a nonlinear function $F_{p_i} = f(D_i, \theta_i)$ with a learnable parameter θ_i to aggregate the local feature of p_i. Similarly, by replacing p_i with d_i, we obtain $F_{d_i} = f(P_i, \theta_i)$. Next, we concatenate the depth-to-pixel feature F_{p_i} with the pixel-to-depth feature F_{d_i} and employ a MLP to obtain the local fused feature:

$$F_{fused} = MLP(F_{p_i} \oplus F_{d_i}), \tag{4}$$

where $\oplus$ represents the concatenation operation. As depicted in Figure 2, the resulting feature is denoted as $F = (F_r, F_d, F_{fused})$.

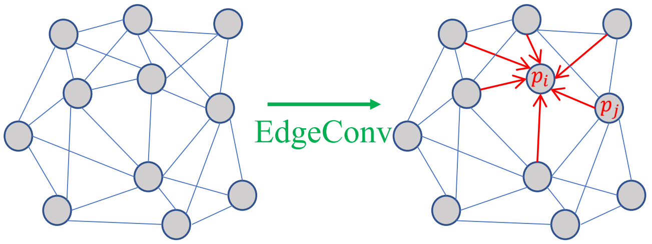


Fig. 6. **Local graph and edge convolution**.

*2) **2D keypoint localization**:* The 3D keypoints are chosen from the 3D object model, similar to methods proposed in [2], [12]. While some methods [5], [10] select the eight corners of the 3D bounding box, these points are virtual and may result in 2D correspondences outside the image. This can lead to significant errors, especially for objects near the image boundary. To address this, we select keypoints on the actual object surface. Following the approach in [2], we employ the farthest point sampling (FPS) algorithm to choose keypoints on the object surface. Subsequently, we employ a network based on [10] for detecting 2D correspondences.

*3) **Learning 6D pose from 2D-3D correspondences**:* As illustrated in Figure 4, we are given a set of n 3D keypoints $K = \{K_i|i = 1, ..., n\}$, where each K_i corresponds to a set of 2D locations $k = \{k_{ij}|j = 1, ..., m\}$ in the image. Our objective is to design a network that can learn the rigid transformation $(\mathbf{R}, \mathbf{t})$ from the established 2D-3D correspondences. Previous approaches such as DSAC [52], Single-Stage [53], and GDR-Net [44] have addressed this problem using different techniques. However, they either rely on sparse correspondences or dense correspondence maps.

To overcome these limitations, we propose a network based on Graph Convolutional Networks (GCNs) to directly regress the 6D pose from the 2D-3D correspondences. The network, denoted as $\mathcal{M}$, is described by the following equation:

$$(\mathbf{R}, \mathbf{t}) = \mathcal{M}(K, k|\Theta), \tag{5}$$

where Θ represents the parameters of the proposed DG-PnP module.

By reevaluating the characteristics of 2D-3D correspondences, we observe that the structure of these correspondences resembles that of a graph. As depicted in Figure 4, instead of treating individual points as inputs, we consider the 2D correspondence cluster as a graph and input it into our DG-PnP module.

*4) **Self-attention Feature Enhance Layer**:* In our experiments, we observed that each point in the 2D cluster contributes differently to the overall results. To address this, we employ a self-attention mechanism to enhance the features. The self-attention mechanism is suitable for point cloud-like data structures due to its permutation-invariant nature. The detailed structure of our self-attention feature enhancement is illustrated in Figure 5.

Let $f \in \mathbb{R}^{N\times d}$ be the input to the self-attention feature enhancement layer, where f_i and f_j represent the features of the i-th and j-th points in the 2D correspondences, respectively.

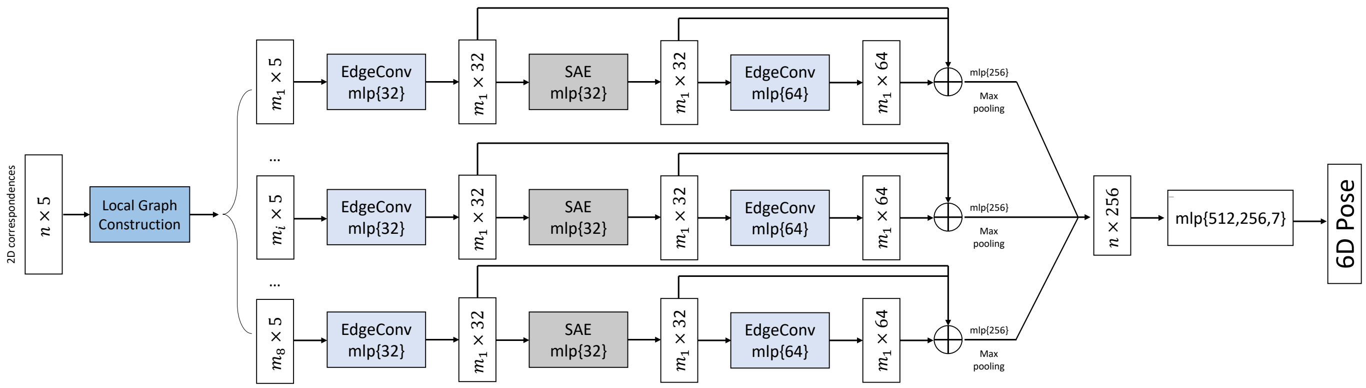


Fig. 7. **The architecture of our DG-PnP++.** The DG-PnP++ model takes a set of 2D correspondences points $n = \{m_1, m_2, ..., m_8\}$ as input. For each 2D correspondence cluster m_i, an EdgeConv layer calculates an edge feature set of size k and aggregates features within each set to compute EdgeConv responses for the corresponding points. Subsequently, a Self-attention enhance (SAE) layer is applied to enhance features by fusing their global feature information. The output features from the last EdgeConv layer are then globally aggregated to form a 1D global descriptor, which is used to regress the 6D pose parameters.

We begin by encoding the position of the i-th point using a Positional Encoder (PE). Then, both f_i and f_j pass through a Multi-Layer Perceptron (MLP), and the resulting output is denoted as g_i:

$$g_i = PE(i) + FC(MLP(f_i)) \cdot FC^T(MLP(f_j)), \quad (6)$$

where FC represents a fully connected layer. Next, we utilize self-attention to obtain g_i, which is then combined with f_j using self-attention. Finally, we apply an aggregation operation to obtain f_i':

$$f_i' = \mathrm{Norm}(g) \cdot FC^T(MLP(f_j)). \quad (7)$$

5) ***Local Graph Construction:*** As shown in Figure 6, we define $\mathcal{P} = \{p_i | i = 1...m\}$ as a cluster of 2D correspondences. To construct the local graph $\mathcal{G} = (\mathcal{P}, \mathcal{E})$, we employ a k-nearest neighbor (k-NN) approach. Here, $\mathcal{P}$ represents the vertices, and $\mathcal{E} = p_i \leftrightarrow p_j$ represents the edges. Subsequently, we compute edge features by aggregating the neighborhoods of p_i within $\mathcal{P}$.

6) ***Edge-convolution:*** Different from a graph convolutional network (GCN), our edge convolution is a variant of a CNN. Given a 2D correspondence cluster with m pixels and X-dimensional features denoted as $f = \{f_i | i = 1, ..., m\}$, we compute the local graph feature using our graph operation:

$$f_i' = \sum_{j=1}^{m} \lambda_j g_{\theta_i}(f_i, f_j), \quad (8)$$

where λ_j is a hyperparameter determined by the distance between k_i and k_j. g_θ represents a non-linear function with learnable parameters θ. We adopt an asymmetric edge function proposed in [42]:

$$g_{\theta_i}(f_i, f_j) = \mathrm{RELU}(\alpha_i \cdot (f_i - f_j) + \beta_i \cdot f_i), \quad (9)$$

where $\theta_i = (\alpha_i, \beta_i)$ and $\Theta = \{\theta_i | i = 1, ..., m\}$ in Eq. 5. In our approach, we consider the 3D coordinates and RGB information of k_i as features f_i, and the 3D coordinates can be obtained by transforming the depth using camera intrinsic parameters. Hence, in our network, $X = 6$.

C. Loss Function and Pose Estimation

To train our network, We employ four distinct loss functions: $\mathcal{L}_d$, $\mathcal{L}_s$, $\mathcal{L}_k$, and $\mathcal{L}_p$. The collective loss function is defined as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_d + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_k + \lambda_4 \mathcal{L}_p, \quad (10)$$

where λ_1, λ_2, λ_3, and λ_4 are the weight coefficients.

$\mathcal{L}_d$ represents the depth loss, Referring to [54], [55], the photometric reconstruction loss function $\mathcal{L}_{\mathrm{ph}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\mathrm{ph}}(\boldsymbol{I}_t, \boldsymbol{I}_{t'\to t}) = {} & \frac{\alpha}{2}(1 - \mathrm{SSIM}(\boldsymbol{I}_t, \boldsymbol{I}_{t'\to t})) \\ & + (1-\alpha)\left\|\boldsymbol{I}_t - \boldsymbol{I}_{t'\to t}\right\|_1,\end{aligned}$$

Here, $\alpha = 0.85$ and SSIM() represents the structural similarity measured computed over a 3×3 pixel window [56]. The re-projected image $\boldsymbol{I}_{t'\to t}$ is defined as:

$$\boldsymbol{I}_{t'\to t} = \boldsymbol{I}_{t'}\langle \mathrm{proj}(\boldsymbol{D}_t, \boldsymbol{T}_{t\to t'}, \boldsymbol{K})\rangle,$$

Where $\langle . \rangle$ denotes the sampling operator, $\boldsymbol{T}_{t\to t'}$ represents the camera relative pose, and the camera intrinsic parameter matrix $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$ is identical for all images. The proj() function calculates the 2D coordinates of the projected depths $\boldsymbol{D}_t$ in image $\boldsymbol{I}_{t'}$ as follows:

$$\mathrm{proj}(\boldsymbol{D}_t, \boldsymbol{T}_{t\to t'}, \boldsymbol{K}) = \boldsymbol{K}\boldsymbol{T}_{t\to t'}\boldsymbol{D}_t(p_t)\boldsymbol{K}^{-1}p_t,$$

Where p_t represents a pixel coordinate. To encourage neighboring pixels to have similar depths, an edge-aware depth smoothness loss $\mathcal{L}_{\mathrm{ds}}$, weighted by image gradients, is employed to improve predictions around object boundaries:

$$\mathcal{L}_{\mathrm{ds}} = |\partial_{\mathrm{x}} \boldsymbol{D}_t^{\mathrm{mn}}| \mathrm{e}^{-|\partial_{\mathrm{x}} \boldsymbol{I}_t|} + |\partial_{\mathrm{y}} \boldsymbol{D}_t^{\mathrm{mn}}| \mathrm{e}^{-|\partial_{\mathrm{y}} \boldsymbol{I}_t|},$$

Here, ∂_{x} and ∂_{y} are gradient operations on the x-axis and y-axis, respectively. $\boldsymbol{D}_t^{\mathrm{mn}} = \boldsymbol{D}_t / \overline{\boldsymbol{D}_t}$ represents the mean-normalized inverse depth. The final loss is computed as the weighted sum of the image photometric reconstruction loss $\mathcal{L}_{\mathrm{p}}$ and the smoothness loss $\mathcal{L}_s$:

$$\mathcal{L}_d = \mathcal{L}_{\text{ph}} + \mu \mathcal{L}_{\text{ds}},$$

where $\mu = 0.01$ represents the weighting for the smoothness term.

$\mathcal{L}_s$ is the segmentation loss, which is used to guide the segmentation task and extract the target object from the image. We adopt the Focal Loss as suggested in [57].

$\mathcal{L}_k$ is the keypoint matching loss, which ensures accurate 2D-3D correspondences. As depicted in Figure 4, our objective is to predict the 2D keypoints in the image, and the loss function is defined as:

$$\mathcal{L}_k = \frac{1}{M} \sum_{i=1}^{n} \sum_{j=1}^{m} ||kp_{ij} - kp_i^*||, \tag{11}$$

where kp_i^* represents the ground truth 2D keypoint location, n is the number of 3D keypoints, m is the number of 2D correspondences for kp_i, and $M = m \times n$ is the total number of 2D correspondences predicted by our network in the image.

$\mathcal{L}_p$ is the final pose estimation loss, which ensures accurate estimation of the 6DoF pose parameters. Inspired by PoseCNN [1] and DeepIM [58], we define $\mathcal{L}_p$ as:

$$\mathcal{L}_p = \frac{1}{n} \sum_{i=1}^{n} \| (\mathbf{R}^* \mathbf{p}_i + \mathbf{t}^*) - (\mathbf{R} \mathbf{p}_i + \mathbf{t}) \|, \tag{12}$$

where $\mathbf{R}^*$ and $\mathbf{t}^*$ are the estimated rotation matrix and translation vector, while $\mathbf{R}$ and $\mathbf{t}$ are the ground-truth values.

Our network is a multi-task network that performs various calculations including depth map estimation, segmentation mask generation, 3D-2D correspondences extraction, and 6DoF pose parameter estimation, similar to the current state-of-the-art methods. In a more general scenario where multiple target objects are present in the image, our network can estimate the poses of these objects simultaneously, as demonstrated in the experimental section.

IV. Experiments

In this section, we conduct a comprehensive set of experiments to validate the effectiveness of DGECN++ on various widely-used benchmark datasets. To facilitate a direct comparison with traditional PnP methods and learning-based PnP approaches, we replicate several experiments following the methodology outlined in [44], [53] using a synthetic sphere dataset to verify the proposed DG-PnP. Furthermore, we conduct an ablation study to analyze the impact and effectiveness of each component in our proposed method.

A. Implementation Details

1) Network Architecture: We feed DGECN with a RGB image and directly output 6D pose. After a cross-domain feature fusion block, we leverage PVNet as backbone to estimate 2D-3D correspondences from the multi-fusion feature of size 256×256. Finally, DG-PnP directly estimates the 6D pose from the estimated 2D-3D correspondences. In Figure 7, we illustrate the detailed network architecture of our proposed DG-PnP++. It takes $n = m_1 + m_2 + ... + m_8$ 2D locations as input, where n is the number of input points, m_i is the number of 2D correspondences in a cluster. The dimension of output pose is 7, including quaternion and translation.

2) Training Strategy: Our network is trained end-to-end using Ranger optimizer on a single GTX3090 GPU. We use a batch size of 24 and a base learning rate of 1e-3 and divided by 10 after processing 50%, 75%, and 90% of the total number of data samples. For 2D localization we utilize FCN on LMO and FCOS on YCB-V. The detectors are trained using the identical training samples employed for pose estimation. We employ an SGD optimizer with specific settings, including a learning rate of 0.001, a momentum value of 0.9, and a weight decay factor of 10^{-4}. We set $\lambda_{1-4} = 1$ in Equation 10.

B. Datasets

1) Synthetic Sphere Dataset: Similar to the approach used in Single-Stage [53], we generate synthetic 2D-3D correspondences using a virtual calibrated camera. The synthetic images have a resolution of 640×480, a focal length of 800, and the principal point is located at the center of the image. However, since our network utilizes color information and extracts local features, including location and color, we introduce a gradient background to the synthetic dataset. The remaining parameter settings are consistent with those of Single-Stage, as depicted in Figure 9.

2) YCB-V Dataset: This dataset, proposed by Calli et al. [1], [63], consists of 21 YCB objects with diverse shapes and textures. It includes 92 RGB-D videos, wherein a subset of objects is captured and annotated with 6D pose and instance semantic masks. The dataset presents challenges such as varying lighting conditions, significant image noise, and occlusions. Following the approach of PoseCNN [1], we divide the dataset into 80 videos for training and select a set of 2,949 keyframes from the remaining 12 videos for testing purposes.

3) LM-O Dataset: The dataset used in this study [64] serves as a widely accepted benchmark for object 6D pose estimation. It consists of 13 videos featuring 13 low-textured objects, along with annotations for 6D pose and instance masks. LM-O presents several challenges, including scenes with high complexity, texture-less objects, and variations in lighting conditions. To address these challenges, we adopt similar approaches as previous works and augment our training set with synthesized images, following the methodology presented in PoseCNN [1].

4) HomebrewedDB Dataset: This dataset, called HomebrewedDB, was recently introduced for evaluating 6D pose estimation [65]. In our study, we specifically utilize the sequence that includes three objects shared with LINEMOD [66]. This choice allows us to showcase the capability of estimating the poses of identical models in different environmental settings.

C. Evaluation metrics

For comparison, we evaluate our method with two common metrics: the average distance (ADD) [1] and the 2D reprojection error (REP) [10].

The Average Distance of Distances (ADD) metric calculates the average distance between the 3D model points

TABLE I

Ablation Study. Results for different versions of our model with comparison to some baseline models. We evaluate the impact of the DGECN, and DG-PnP. (s) denotes symmetric objects, we report the Average Recall (%) of ADD(-S) on LM-O dataset. The overall best results are presented in bold, while the second best results are underlined.

2D-3D extractor	PnP type	Ape	Can	Cat	Driller	Duck	Eggbox^s	Glue^s	Holepun	Mean
DGECN++(**Ours**)	DG-PnP++(Ours)	**56.1**	80.3	**30.5**	**78.3**	**55.2**	**62.3**	**70.6**	**68.6**	**62.5**
	DG-PnP	54.3	75.9	22.4	77.5	51.2	57.8	66.9	63.2	58.7
	PointNet-like PnP	44.4	71.3	18.5	71.6	48.6	51.3	59.1	60.3	53.1
	Patch-PnP	51.2	74.6	21.6	73.4	48.5	56.9	65.1	61.4	56.6
	RANSAC-based PnP	41.3	66.5	14.3	65.4	44.1	48.9	55.4	56.2	49.0
	BPnP	46.2	73.3	19.5	72.4	46.2	52.1	61.4	56.2	53.4
PVNet	DG-PnP++(Ours)	26.3	71.2	24.6	73.2	28.5	58.1	51.6	48.5	47.8
	DG-PnP	23.4	68.9	23.2	72.2	27.8	55.1	53.2	47.2	46.4
	PointNet-like	19.2	65.1	18.9	69.0	25.3	52.0	51.4	45.6	43.3
	Patch-PnP	14.4	55.3	14.9	68.2	22.1	45.9	49.4	41.3	38.9
	RANSAC-based PnP	15.8	63.3	16.7	65.7	25.2	50.3	49.6	36.1	40.8
	BPnP	21.4	45.3	12.7	64.3	21.4	42.1	44.5	38.7	36.3
SegDriven	DG-PnP++(Ours)	19.9	53.2	16.6	58.2	21.8	32.6	49.3	42.1	36.7
	DG-PnP	17.5	51.4	15.9	57.9	20.6	31.8	43.2	39.6	34.7
	PointNet-like	14.8	45.5	12.1	54.6	18.3	30.2	45.8	37.4	32.3
	Patch-PnP	9.8	36.9	14.6	57.3	11.6	28.3	42.3	32.4	28.4
	RANSAC-based PnP	12.1	39.9	8.2	45.2	17.2	22.1	35.8	36.0	27.0
	BPnP	15.6	47.8	14.5	51.3	14.8	30.5	26.4	32.1	29.1
GDR-Net	DG-PnP++(Ours)	40.5	**81.2**	28.8	73.1	50.3	58.3	51.5	56.9	55.1
	DG-PnP	37.5	78.5	26.8	70.6	42.9	56.8	50.4	56.4	52.5
	PointNet-like PnP	17.9	65.3	18.6	62.8	31.5	48.6	36.7	49.2	41.3
	Patch-PnP	39.3	79.2	23.5	71.3	44.4	58.2	49.3	58.7	53.0
	RANSAC-based PnP	20.9	67.5	23.9	66.1	34.9	53.4	42.3	54.3	45.4
	BPnP	35.5	74.2	21.5	67.4	36.9	51.4	45.8	51.1	48.0

TABLE II

Quantitative comparison on known categories of LM-O dataset with state-of-the-art RGB methods with the metric as ADD(-S), (R) stands for Refinement. All methods trained with $real + syn$ data. P.E. means whether the method is trained with 1 pose estimator for the whole dataset or 1 per object (N objects in total).

Method	PoseCNN	PVNet	Single-Stage	HybridPose	GDR-Net	SO-Pose	DeepIM^R	DPOD^R	DGECN	DGECN++(Ours)	
P.E.	1	N	N	N	1	1	1	1	N	1	N
Ape	9.6	15.8	19.2	20.9	41.3	46.3	**59.2**	-	50.3	50.1	52.1
Can	45.2	63.3	65.1	75.3	71.1	**81.1**	63.5	-	75.9	73.8	**76.3**
Cat	0.9	16.7	18.9	24.9	23.5	18.2	26.2	-	26.4	25.4	**27.5**
Driller	41.4	65.7	69.0	70.2	54.6	71.3	55.6	-	77.5	77.6	**78.3**
Duck	19.6	25.2	25.3	27.9	41.7	43.9	52.4	-	54.2	**55.5**	55.2
Eggbox^s	22.0	50.2	52.0	52.4	40.2	46.6	**63.0**	-	57.8	59.1	62.3
Glue^s	38.5	49.6	51.4	53.8	59.5	63.3	**71.7**	-	66.9	64.1	66.6
Holepun	22.1	36.1	45.6	54.2	52.6	62.9	52.5	-	60.2	58.3	**60.6**
Mean	24.9	40.8	43.3	47.5	47.4	54.3	55.5	47.3	58.7	58.0	**59.9**

TABLE III

Evaluation with state-of-the-art RGB methods on YCB-V. Ref. stands for Refinement. P.E. means whether the method is trained with 1 pose estimator for the whole dataset or 1 per object (N objects in total).

Method	Ref.	P.E.	ADD(-S)	REP-5px	AUC of ADD-S
PoseCNN		1	21.3	3.7	75.9
GDR-Net		N	60.1	-	91.6
SO-Pose		N	56.8	-	90.9
PVNet		N	-	47.4	73.4
SegDriven		1	39.0	30.8	-
Single-Stage		N	53.9	48.7	-
DeepIM	✓	1	-	-	88.1
CosyPose	✓	1	-	-	89.8
DGECN		1	60.6	50.3	90.9
Ours		1	67.1	60.5	92.5
		N	**69.5**	**63.2**	**93.1**

transformed using the predicted pose and those obtained with the ground truth pose. If the distance is less than 10% of the model's diameter, we consider the estimated pose to be correct. For symmetric objects, we use the ADD(-S) metric,

TABLE IV

Ablation on Depth Map. ✓ denotes test with depth map and ✗ denotes test without depth map.

Corr. Extractor	DG-PnP	ADD	AUC of ADD-S
✓	✓	58.7	90.9
✓	✗	53.2	83.5
✗	✓	50.6	81.3
✗	✗	41.3	75.3

which measures the deviation to the closest model point. Let's denote the predicted pose as $[\mathbf{R}^*, \mathbf{t}^*]$ and the ground truth pose as $[\mathbf{R}, \mathbf{t}]$. The ADD metric is calculated as:

$$\mathrm{ADD} = \frac{1}{m} \sum_{x \in \mathcal{O}} \|(Rx + t) - (R^* x + t^*)\|. \tag{13}$$

The ADD-S metric is calculated as:

$$\mathrm{ADD-S} = \frac{1}{m} \sum_{x_1 \in \mathcal{O}} \min_{x_2 \in \mathcal{O}} \|(Rx_1 + t) - (R^* x_2 + t^*)\|, \tag{14}$$

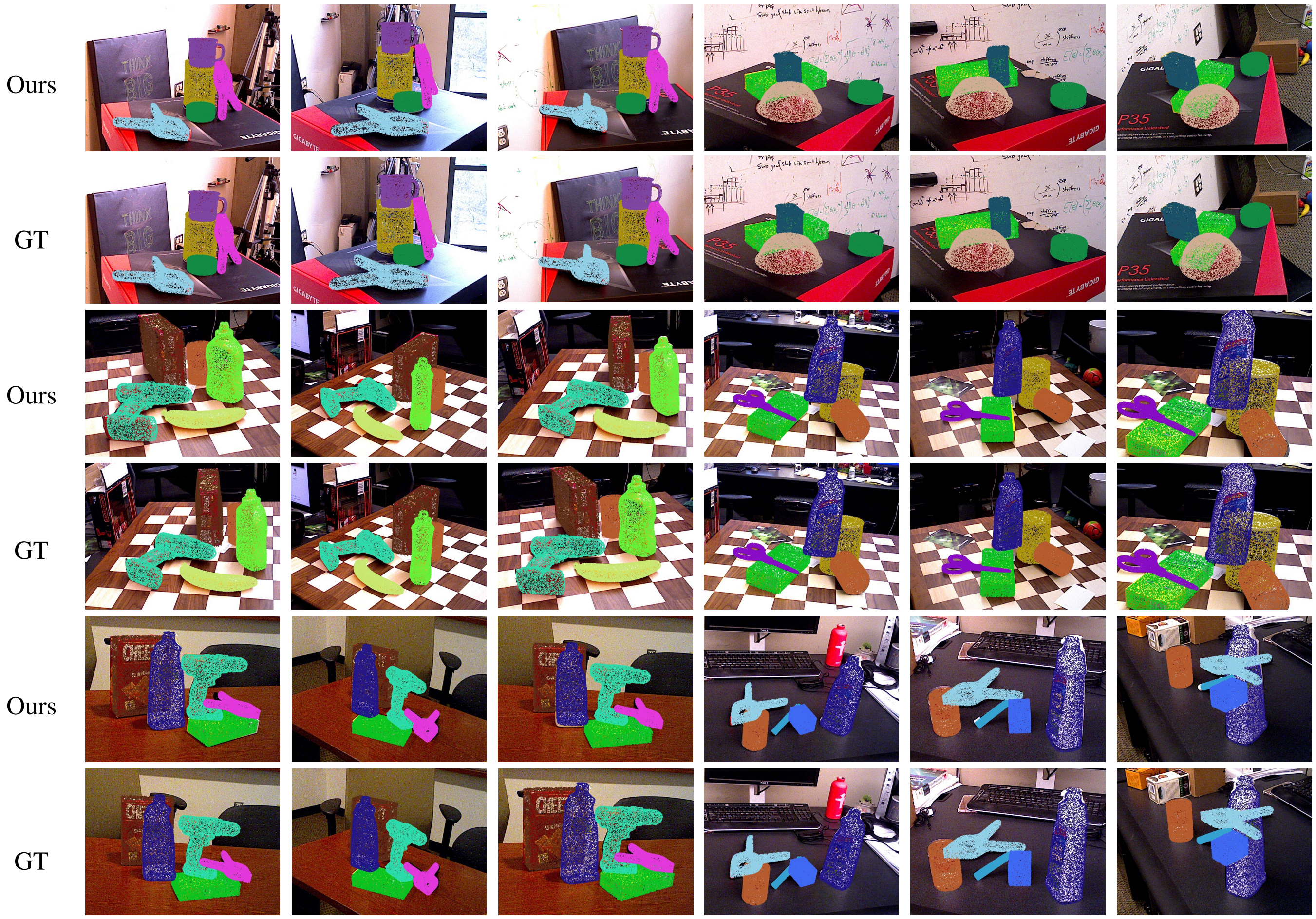


Fig. 8. **Qualitative results on YCB-V dataset.** Here we show visualizations of results on YCB-V dataset. Points on different meshes in the same scene are in different colors which projected back to the image after being transformed by the predicted pose.

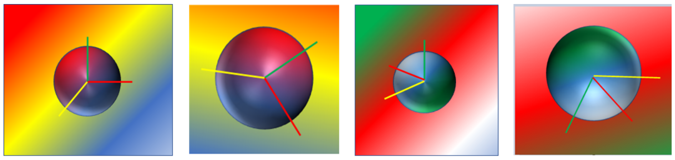


Fig. 9. **Synthetic data.** We generate synthetic data following the approach described in [53], with the addition of incorporating backgrounds into the scenes.

where x represents a vertex among a total of m vertices on the object mesh $\mathcal{O}$.

The Reprojection Error (REP) metric calculates the average distance between the projections of the 3D model points based on the estimated pose and the ground truth pose. If the REP value is below 5 pixels, we consider the estimated pose to be accurate.

When evaluating on YCB-Video, we further compute the AUC (area under curve) of ADD-S/ADD(-S) by varying the distance threshold from 0cm to 10cm as in PoseCNN. Thereby, ADD-S uses the symmetric metric for all objects, while ADD(-S) only uses the symmetric metric for symmetric objects. For Cropped LINEMOD, we report the average angle error following PixelDA. For each metric, we use the symmetric version for symmetric objects, which we denote by a superscript (s).

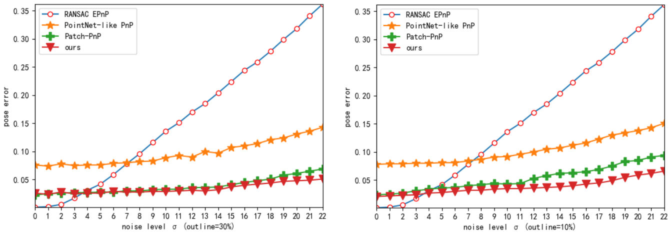


Fig. 10. **Comparison with PnP variants**. We conducted comparisons between our method and three existing approaches: EPnP [59], PointNet-like PnP [53], and Patch-PnP [44]. Our method consistently outperforms PointNet-like PnP in terms of accuracy. Furthermore, as the noise level increases, our method exhibits superior accuracy and robustness compared to EPnP. The pose error is evaluated using the ADD metric.

D. *Comparison with State-of-the-arts*

We compare with the state-of-the-art works on YCB-V and LM-O datasets. It is worth mentioning that we also make a comparison with the RGB-D based methods to verify the effectiveness of our depth estimation network.

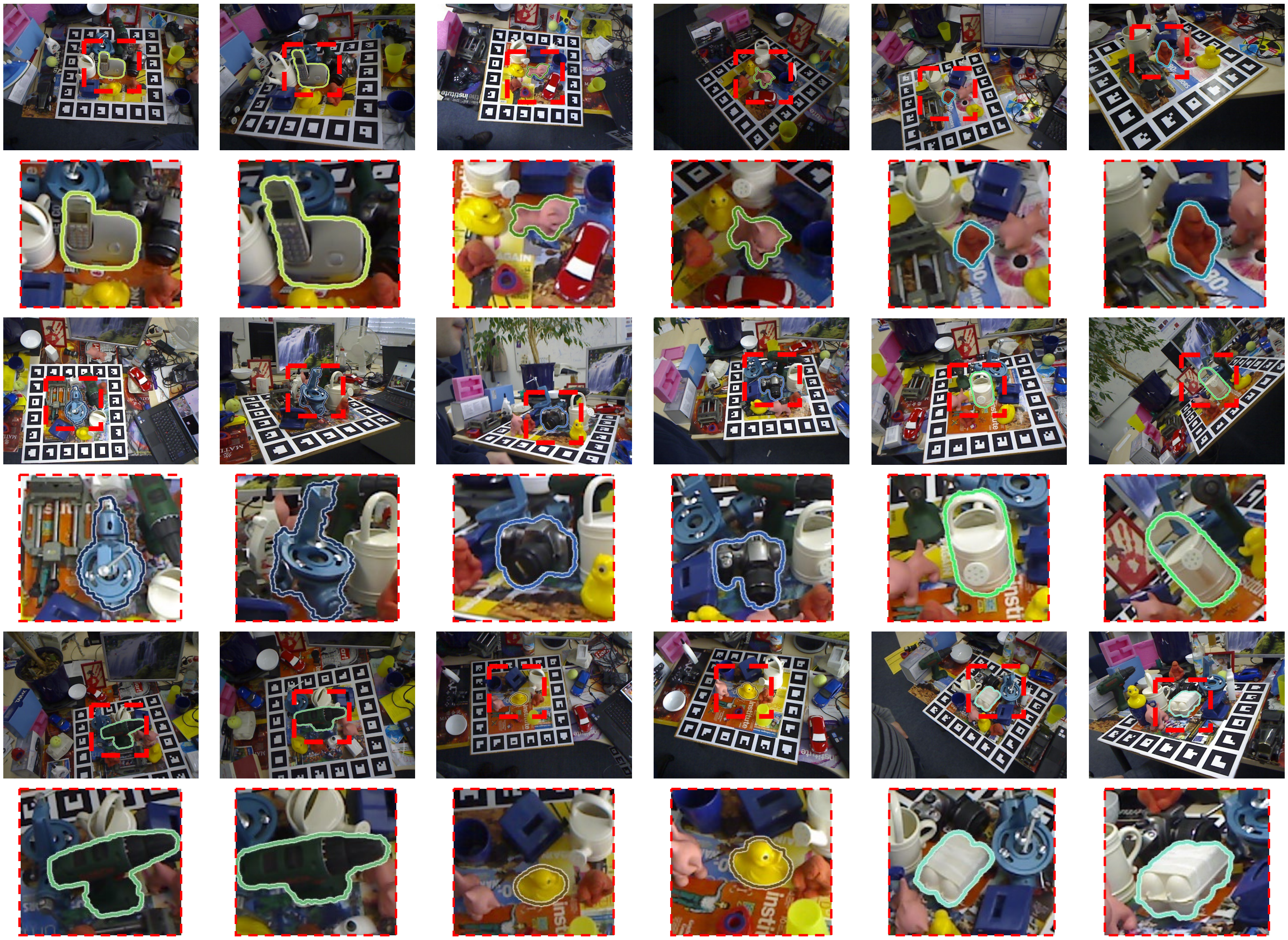


Fig. 11. **Qualitative results on LM-O.** Here, The colorful silhouettes represent the estimated 6D poses.

TABLE V
EVALUATION WITH STATE-OF-THE-ART RGB-D METHODS ON YCB-V. (*) DENOTES REPLACING DEPTH ESTIMATION NETWORK WITH GROUND TRUTH LABELS.

Method	ADD(-S)	REP-5px	AUC of ADD-S
Implicit ICP [60]	64.7	-	-
SSD-6D ICP [61]	79.0	-	91.6
PointFusion [62]	-	73.7	73.4
DenseFusion [14]	86.2	30.8	-
PVN3D [12]	53.9	99.4	-
DGECN [26]	60.6	50.3	90.9
DGECN*	82.1	99.2	91.5
Ours	67.1	60.5	92.5
Ours*	**85.3**	**99.8**	**95.5**

1) Performance on HomebrewedDB: We compare our method with DPOD, YOLO6D and SSD6D, along with the refined version (SSD6D+Ref.), on three objects from the HomebrewedDB dataset, which is also used in LINEMOD. We strictly follow the experimental setup of HomebrewedDB, where our models are trained on real LineMOD data and evaluated on a new sequence in HomebrewedDB, which includes three LineMOD objects: a benchvise, a drill, and a phone. However, direct methods for solving 6D pose estimation implicitly learn the camera intrinsics, which hampers their performance when faced with a new camera. In contrast, approaches based on 2D-3D correspondences, such as PnP, are more robust to camera changes as they can simply use the new intrinsics for pose estimation. By employing contour-based pose refinement and rendering with the new intrinsics, SSD6D+Ref. enables easy adaptation and even outperforms DPOD and other approaches for the Bvise object. As shown in Table IX, among the methods compared, our approach consistently outperforms others on both synthetic and real data. Specifically, on the real dataset, we observe a significant improvement of at least 12% compared to the performance achieved by DPOD [67], and outperforms approaches like DPOD and SSD6D+Ref. on the synthetic by at least 10% and 30%, respectively.

2) Performance on LM-O and LM dataset: Table II presents a comparison of DGECN++ with state-of-the-art monocular methods on the Occlusion LINEMOD dataset. Our DGECN++ achieves comparable performance to methods such as [44], [58], [68], while outperforming [2], [53]. It is important to mention that the Occlusion LINEMOD dataset poses additional challenges due to strong occlusions frequently occurring

TABLE VI
EFFECT OF THE DESIGN ON DGECN++. EVALUATE ON LM DATASET.

Self-attetion Feature Enhance	Depth Refinement Network	K-NN Feature Aggregation	ADD	AUC of ADD-S
✗	✗	✗	46.3	78.4
✗	✓	✗	47.6	81.2
✗	✗	✓	52.8	83.7
✓	✗	✗	48.3	80.5
✓	✓	✗	53.8	85.5
✓	✗	✓	52.9	82.9
✗	✓	✓	54.9	87.6
✓	✓	✓	59.9	92.5

TABLE VII
DETAILED RESULTS ON YCB-V W.R.T. ADD(-S). (S) DENOTES SYMMETRIC OBJECTS. THE OVERALL BEST RESULTS ARE PRESENTED IN BOLD, WHILE THE SECOND BEST RESULTS ARE UNDERLINED.

Method	PoseCNN	SegDriven	Single-Stage	GDR-Net	DGECN	DGECN++**(Ours)**
002 master chef can	3.6	33.0	-	41.5	$\underline{45.3}$	**50.6**
003 cracker box	25.1	44.6	-	$\underline{83.2}$	77.5	**85.4**
004 sugar box	40.3	75.6	-	91.5	$\underline{94.8}$	**97.3**
005 tomato soup can	25.5	40.8	-	65.9	$\underline{71.2}$	**78.5**
006 mustard bottle	61.9	70.6	-	90.2	89.9	**93.6**
007 tuna fish can	11.4	18.1	-	44.2	$\underline{54.3}$	**59.5**
008 pudding box	14.5	12.2	-	2.8	$\underline{16.7}$	**22.6**
009 gelatin box	12.1	59.4	-	61.7	$\underline{62.2}$	**69.7**
010 potted meat can	18.9	33.3	-	64.9	$\underline{65.8}$	**71.6**
011 banana	30.3	16.6	-	64.1	$\underline{78.9}$	**81.6**
019 pitcher base	15.6	90.0	-	$\underline{99.0}$	98.5	**100**
021 bleach cleanser	21.2	70.9	-	73.8	$\underline{82.1}$	**84.6**
024 bowl^S	12.1	30.5	-	$\underline{37.7}$	23.5	**40.3**
025 mug	5.2	40.7	-	61.5	$\underline{63.5}$	**65.8**
035 power drill	29.9	63.5	-	$\underline{78.5}$	77.2	**81.3**
036 wood block^S	10.7	27.7	-	59.5	$\underline{62.3}$	**66.5**
037 scissors	2.2	17.1	-	3.9	$\underline{18.3}$	**23.6**
040 large marker	3.4	4.8	-	7.4	$\underline{8.1}$	**12.3**
051 large clamp^S	28.5	25.6	-	**69.8**	55.6	$\underline{66.6}$
052 extra large clamp^S	19.6	8.8	-	90.0	90.1	**92.3**
061 foam brick^S	54.5	34.7	-	**71.9**	38.6	$\underline{65.8}$
Average	21.3	39.0	53.9	60.1	$\underline{60.6}$	**67.1**

on many objects. To ensure a fair evaluation, we compare our methodology with state-of-the-art methods using synthetic data exclusively within the BOP setup, see Table VIII. Our baseline method, DGECN++, demonstrates a significant performance advantage over all other methods. Notably, it surpasses the current top-performing method, CosyPose, from the BOP leader board, by a substantial margin. Specifically, Ours achieves an impressive accuracy of 68.3%, surpassing CosyPose's 62.6% by 5.7%.

3) Performance on YCB-V: Table VII presents the evaluation results for the YCB-V dataset, demonstrating the performance of our model. Our model is comparable to state-of-the-art methods such as [44], [69], and it even outperforms the refinement-based method proposed in [58]. Figure IV showcases qualitative results on the YCB-V dataset. In general, our observations align with those made for the other datasets. Specifically, the incorporation of geometric information, either from RGB or RGB-D, contributes to improved performance compared to the associated baselines. Furthermore, Table V displays a comparison with RGB-D based methods. Notably, in certain scenes, our proposed method even surpasses RGB-D based approaches without the availability of ground truth depth maps. For the AUC of ADD(-S), our method achieves 67.1%, while DenseFusion achieves 82.6%, and for the AUC of ADD-S, our method achieves 91.9% compared to SSD6D+ICP's 91.6%. Moreover, with the utilization of the attention mechanism and the utilization of real depth data in place of the depth estimation module, our method surpasses all compared RGB-D based methods in performance.

TABLE VIII
Comparison with state-of-the-art methods on LMO and YCB-V under BOP metrics. We provide results for AR_{VSD}, AR_{MSSD} and AR_{MSPD} on LMO and YCB-V. Mean AR represents the overall performance on these two datasets as the average over all AR scores. The overall best results are presented in bold, while the second best results are underlined.

Method	Ref.	LMO			Mean AR	YCB-V			Mean AR
		AR_VSD	AR_MSSD	AR_MSPD		AR_VSD	AR_MSSD	AR_MSPD	
CosyPose	✓	<u>0.480</u>	0.606	0.812	<u>0.632</u>	<u>0.772</u>	<u>0.842</u>	**0.850**	**0.821**
EPOS		0.389	0.501	0.750	0.547	0.626	0.677	0.783	0.695
PVNet		0.428	0.543	0.754	0.575	-	-	-	-
CDPN		0.445	<u>0.612</u>	0.815	0.624	0.396	0.570	0.631	0.532
GDR-Net		-	-	-	-	0.584	0.674	0.726	0.661
SO-Pose		0.442	0.581	<u>0.817</u>	0.613	0.652	0.731	0.763	0.715
DGECN		0.458	0.593	0.816	0.622	0.663	0.726	0.775	0.721
DGECN++		**0.542**	**0.672**	**0.844**	**0.683**	**0.793**	**0.853**	<u>0.797</u>	<u>0.814</u>

TABLE IX
Pose estimation results in terms of ADD 10%metric on HomebrewedDB dataset. The best method is marked in bold.

Method	Supervision	Object			Mean
		Bvise	Drill	Phone	
Ours	Real GT	**63.5**	**70.6**	**43.2**	**59.1**
YOLO6D		15.3	6.5	0.1	7.3
DPOD		57.2	62.8	33.1	51.0
Ours	Synthetic	**77.5**	**72.4**	**45.9**	**65.3**
SSD6D+Ref.		59.4	25.1	29.3	37.9
DPOD		70.9	66.4	35.6	57.6

E. Ablation study

In this section, we aim to discuss the following research questions:

1. How does DG-PnP++ compare to handcrafted PnP methods and other learnable PnP approaches?

2. Does the incorporation of learned depth information enhance the accuracy of the final pose estimation?

3. Is DGECN++ backbone effective in combination with various PnP variants?

By investigating these questions, we can gain insights into the comparative performance of DG-PnP, the impact of learned depth on pose estimation, and the effectiveness of DGECN in conjunction with different PnP variations.

1) Comparison to PnP Variants: For the training phase, we utilize a dataset comprising 20,000 synthetic images, while 2,000 images are reserved for testing purposes. During the training process, we introduce random 2D noise with a variance σ ranging from 0 to 15 and incorporate outliers at rates of 10% and 30%. The comparison conducted on synthetic data is of paramount importance as it allows for a direct evaluation of DG-PnP against PnP variants, while mitigating the influence of keypoint detection methods. Figure 5 showcases the results obtained at different noise levels, comparing DG-PnP with EPnP [59], PointNet-like PnP [53], and Patch-PnP [44]. While handcrafted PnP methods demonstrate higher accuracy under minimal noise conditions, learnable PnP methods exhibit increased robustness to noise and achieve superior accuracy as the noise level increases. Notably, DG-PnP exhibits remarkable robustness and accuracy, surpassing PointNet-like PnP and performing comparably to Patch-PnP. This can be attributed to DG-PnP++ and Patch-PnP considering both geometric and topological features, enabling them to achieve superior performance.

2) Ablation on Depth Map and DRN: As mentioned earlier, depth information plays a crucial role in 6D pose regression. However, in our experiments, we trained our DGECN++ model without utilizing depth estimation. Since depth information is utilized in both correspondence extraction and DG-PnP++, we conducted an ablation study to examine its impact. The results, shown in Table IV, clearly demonstrate that DGECN++ achieves significantly improved robustness when depth prediction is incorporated. Table VI further illustrates the impact of the Depth Refinement Network (DRN). Notably, when the depth refinement network is not utilized, there is a significant decrease of 7.0 in the Average Distance Difference (ADD) metric and a decrease of 9.6 in the Area Under Curve (AUC) of the ADD-S metric.

3) Effect of the Design and Each Component in DGECN++: Table I presents a comprehensive analysis of the effectiveness of each component in our proposed method. To evaluate the impact of different components, we combine them with various state-of-the-art methods. For DGECN++, we substitute the DG-PnP++ module in our architecture with different PnP variants [26], [35], [44], [53]. The results demonstrate that DGECN++ achieves competitive performance when compared to different PnP methods. Notably, it even outperforms the combination of Single-Pose with the PointNet-like PnP. Regarding DG-PnP++, we replace the PnP variants in certain two-stage methods with DG-PnP++. This analysis provides insights into the contribution of DG-PnP++ within these frameworks and its impact on overall performance.

We evaluate the effectiveness of each component in DGECN++ as shown in Table VI. By selectively adding and removing individual components, we can demonstrate the efficacy of each component. In particular, the incorporation of the self-attention mechanism in the original version noticeably enhances the accuracy of 6D pose estimation.

By conducting these experiments and evaluations, we gain a deeper understanding of the effectiveness and potential advantages of each component in our proposed method.

TABLE X
ABLATION STUDY UNDER BOP SETUP ON LMO AND YCBV DATASET.

Row	Method	LMO			Mean AR	YCB-V			Mean AR	FPS
		AR_VSD	AR_MSSD	AR_MSPD		AR_VSD	AR_MSSD	AR_MSPD		
A0	DGECN++	0.542	0.672	$\underline{0.844}$	0.683	$\underline{0.793}$	0.853	0.797	0.814	30
B0	A0: Sparse corr.→ Dense corr.	0.549	$\underline{0.689}$	**0.856**	**0.698**	**0.812**	**0.881**	0.806	**0.833**	15
B1	B0: DG-PnP++→ DG-PnP	0.512	0.638	0.779	0.643	0.783	0.844	0.765	0.797	17
B2	B0: DG-PnP++→ CNN	0.432	0.511	0.669	0.537	0.619	0.721	0.512	0.617	10
C0	A0: → dynamic graph for feature extraction	**0.553**	0.678	0.841	0.691	0.790	$\underline{0.885}$	$\underline{0.812}$	$\underline{0.829}$	28
C1	B0: → dynamic graph for feature extraction	$\underline{0.551}$	**0.703**	0.832	$\underline{0.695}$	0.775	0.872	**0.818**	0.816	14

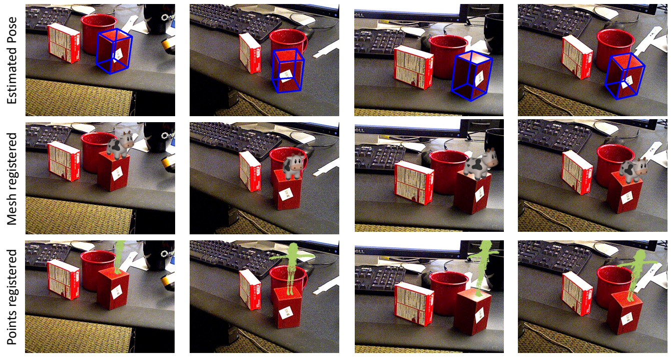


Fig. 12. **Application.** Real-time 3D registration of point clouds and meshes.

TABLE XI
RUNTIME ANALYSIS. WE PROVIDE RUN-TIME STATISTICS FOR THE KEY STEPS IN OUR METHODOLOGY AND CONTRAST THEM WITH THE BASELINE'S PERFORMANCE.

Method	Dadaset	Detector	Depth Pred.	Feature Fusion	Pose Reg.	FPS
DGECN	LM	10ms	7ms	2ms	3ms	40
Ours	LM	10ms	7ms	2ms	4ms	40
	YCB-V(1 obj.)	10ms	7ms	2ms	4ms	40
	YCB-V(8 obj.)	15ms	10ms	3ms	5ms	30

4) Ablation Study of Different Designs of Architecture: We have conducted an ablation study on replacing sparse correspondences with dense correspondence, and a dynamic graph again for feature extraction in DG-PnP++. Table IX shown that The utilization of dense correspondences connectivity undeniably improves the accuracy of our method. Nevertheless, given our reliance on k-nearest neighbor (knn) search for graph convolution, adopting dense connections would lead to a substantial escalation in computational requirements, rendering real-time performance unachievable. While graph feature extraction can indeed trade less time loss for higher accuracy.

5) Ablation Study of Different Detectors: On the Occluded LINEMOD test set of the BOP dataset, we conducted an experiment by switching the synthetically trained object detector from FCN (with metrics: AP: 68.3, AP50: 91.5, AP75: 76.8, Speed: 32.4 ms/img) to the slightly less accurate but significantly slower Faster R-CNN with a ResNet101 backbone (with metrics: AP: 66.9, AP50: 89.1, AP75: 74.8, Speed: 77.5 ms/img). Surprisingly, this change resulted in only a marginal 1.4% drop in performance for DGECN++. As a result, we continue to use FCN as the base detector in all other experiments due to its superior accuracy and efficiency.

F. Runtime Analysis

All our experiments are implemented using PyTorch [70]. We test our method on a PC with an Intel E5-2630 CPU and a GTX 3090 GPU. Given a 640 × 480 image, using the FCN detector, our approach takes ≈ 32 ms for 8 objects, which include ≈ 15 ms for detection, for ≈ 13 ms for depth estimation and correspondence extraction and ≈ 5 ms for DG-PnP to estimate 6D pose. Table XI provides an overview of the computational times for each step of our method, with particular emphasis on object detection and depth estimation, which consume the majority of the processing time.

G. Applications

Our method is suitable for various applications since it is real-time and effective.

Objects Grasp. Robotic grasping is a critical capability that allows robots to interact with their surroundings by manipulating and securely grasping objects. It serves as a foundational skill for robots to perform a wide range of tasks, including pick-and-place operations, assembly tasks, and object manipulation.

The primary objective of robotic grasping is to enable a robot to achieve secure and reliable grasps on objects with varying shapes, sizes, and material properties. This entails the robot perceiving the objects in its environment, devising suitable grasp strategies, and executing the grasps with precision and control. While our method primarily focuses on grasping rigid objects, its principles and techniques can be adapted for other object types as well.

Virtual Reality. Three-dimensional registration technology holds significant importance in the realm of virtual reality (VR). It serves the purpose of accurately aligning and integrating virtual objects with the real world, facilitating a seamless interaction between the virtual and real environments to provide an immersive experience for users.

Traditional 3D registration methods typically rely on speciali zed equipment such as depth sensors (*e.g.* , RGB-D cameras) or laser scanners to capture the geometric structure and texture information of the real world. These sensors enable the acquisition of object shapes, positions, and surface characteristics within the environment. In contrast, our proposed solution leverages a single consumer-grade color camera, as illustrated in Figure 12, to achieve similar results.

V. CONCLUSION

In this work, we have introduced DGECN++, a novel end-to-end depth-guided network designed to learn from RGB images without real depth annotations. The main idea behind our approach is to leverage geometric and topological information to jointly address 2D keypoint detection and 6D pose estimation. We further explore the use of graph structures to model the distribution of keypoints more effectively in the context of 2D-3D correspondences. Additionally, we propose a dynamic graph PnP approach to learn the 6D pose, which replaces the traditional handcrafted PnP method. As a result, our approach offers real-time performance, high accuracy, and robustness for monocular 6D object pose estimation.

Additionally, we share common constraints with other model-based pose estimation methods, necessitating the use of CAD models to ensure precise 2D-3D correspondences. As we look to future research directions, we strongly believe that our reformulation has the potential to significantly impact self-supervision pipelines, particularly for unseen objects, by enabling joint optimization of both geometry and texture information. Additionally, exploring the generalization capability of our approach to novel object instances or categories would provide valuable insights into its robustness and applicability in real-world settings with diverse and evolving object classes. We firmly believe that model-free pose estimation represents a promising and pivotal direction for future advancements in the field.

References

[1] 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. 1, 2, 5, 7

[2] 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. 1, 2, 3, 4, 5, 10

[3] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, 2001. 1

[4] F. Tang, Y. Wu, X. Hou, and H. Ling, "3d mapping and 6d pose computation for real time augmented reality on cylindrical objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2887–2899, 2020. 1

[5] M. Rad and V. Lepetit, "Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3828–3836. 1, 5

[6] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7668–7677. 1, 3

[7] D. Fu, S. Han, B. Liang, and W. Li, "The 6d pose estimation of the aircraft using geometric property," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 7, pp. 3358–3368, 2023. 1

[8] Y. Fu, Q. Yan, J. Liao, and C. Xiao, "Joint texture and geometry optimization for rgb-d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5950–5959. 1

[9] Y. Fu, Q. Yan, J. Liao, H. Zhou, J. Tang, and C. Xiao, "Seamless texture optimization for rgb-d reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, 2021. 1

[10] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3385–3394. 1, 2, 5, 7

[11] G. Feng, T.-B. Xu, F. Liu, M. Liu, and Z. Wei, "Nvr-net: Normal vector guided regression network for disentangled 6d pose estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023. 1, 3

[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. 1, 3, 4, 5, 10

[13] K. Wada, E. Sucar, S. James, D. Lenton, and A. J. Davison, "Morefusion: multi-object reasoning for 6d pose estimation from volumetric fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 540–14 549. 1

[14] 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. 1, 3, 4, 5, 10

[15] J. Liu, W. Sun, C. Liu, X. Zhang, S. Fan, and W. Wu, "Hff6d: Hierarchical feature fusion network for robust 6d object pose tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 11, pp. 7719–7731, 2022. 1, 3

[16] 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. 1

[17] 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. 1

[18] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 9, pp. 850–863, 1993. 1

[19] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753. 1, 2

[20] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946. 1, 2

[21] T. Hodan, D. Barath, and J. Matas, "Epos: estimating 6d pose of objects with symmetries," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 703–11 712. 1, 3

[22] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7678–7687. 1

[23] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, and F. Tombari, "Explaining the ambiguity of object detection and 6d pose from visual data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6841–6850. 1

[24] D. Rozumnyi, J. Kotera, F. Sroubek, and J. Matas, "Sub-frame appearance and 6d pose estimation of fast moving objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6778–6786. 1

[25] J. Shao, Y. Jiang, G. Wang, Z. Li, and X. Ji, "Pfrl: Pose-free reinforcement learning for 6d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 454–11 463. 1

[26] T. Cao, F. Luo, Y. Fu, W. Zhang, S. Zheng, and C. Xiao, "Dgecn: A depth-guided edge convolutional network for end-to-end 6d pose estimation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Apr 2022. 2, 5, 10, 12

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9. 2

[28] H.-W. Hsu, T.-Y. Wu, S. Wan, W. H. Wong, and C.-Y. Lee, "Quatnet: Quaternion-based head pose estimation with multiregression loss," *IEEE Transactions on Multimedia*, p. 1035–1046, Apr 2019. [Online]. Available: http://dx.doi.org/10.1109/tmm.2018.2866770 2

[29] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134. 3

[30] H. Chen, F. Manhardt, N. Navab, and B. Busam, "Texpose: Neural texture learning for self-supervised 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 4841–4852. 3

[31] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 254–269. 3

[32] G. Zhou, H. Wang, J. Chen, and D. Huang, "Pr-gcn: A deep graph convolutional network with point refinement for 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2793–2802. 3

[33] Y. Hai, R. Song, J. Li, and Y. Hu, "Shape-constraint recurrent flow for 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 4831–4840. 3

[34] E. Brachmann and C. Rother, "Learning less is more - 6d camera localization via 3d surface regression," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2018.00489 3

[35] B. Chen, A. Parra, J. Cao, N. Li, and T.-J. Chin, "End-to-end learnable geometric vision by backpropagating pnp optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8100–8109. 3, 12

[36] D. Campbell, L. Liu, and S. Gould, *Solving the Blind Perspective-n-Point Problem End-to-End with Robust Differentiable Geometric Optimization*, Jan 2020, p. 244–261. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-58536-5_15 3

[37] S. Iwase, X. Liu, R. Khirodkar, R. Yokota, and K. Kitani, "Repose: Fast 6d object pose refinement via deep texture rendering," *International Conference on Computer Vision*, Jan 2021. 3

[38] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li, "Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 2781–2790. 3

[39] X. Dong, C. Long, W. Xu, and C. Xiao, "Dual graph convolutional networks with transformer and curriculum learning for image captioning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2615–2624. 3

[40] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2272–2281. 3

[41] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3d semantic scene graphs from 3d indoor reconstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[42] 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 (tog)*, vol. 38, no. 5, pp. 1–12, 2019. 4, 6

[43] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947. 4

[44] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 16 611–16 621. 4, 5, 7, 9, 10, 11, 12

[45] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440. 4, 5

[46] 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. 4

[47] W. Zhang and C. Xiao, "Pcan: 3d attention map learning using contextual information for point cloud based retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 436–12 445. 4

[48] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *European Conference on Computer Vision*. Springer, 2020, pp. 512–528. 4

[49] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019, pp. 3827–3837. 4

[50] M. Ramamonjisoa, M. Firman, J. Watson, V. Lepetit, and D. Turmukhambetov, "Single image depth prediction with wavelet decomposition," pp. 11 089–11 098, June 2021. 4

[51] J. Watson, M. Firman, G. Brostow, and D. Turmukhambetov, "Self-supervised monocular depth hints," in *ICCV*, 2019, pp. 2162–2171. 4

[52] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac - differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5

[53] Y. Hu, P. Fua, W. Wang, and M. Salzmann, "Single-stage 6d object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2930–2939. 5, 7, 9, 10, 12

[54] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6

[55] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 6

[56] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, p. 600–612, Apr 2004. 6

[57] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988. 7

[58] 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. 7, 10, 11

[59] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009. 9, 12

[60] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 699–715. 10

[61] 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. 10

[62] 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*, 2018, pp. 244–253. 10

[63] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517. 7

[64] 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

[65] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0. 7

[66] 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. 7

[67] S. Zakharov, I. Shugurov, and S. Ilic, "Dpod: 6d pose object detector and refiner," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1941–1950. 10

[68] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "So-pose: Exploiting self-occlusion for direct 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 396–12 405. 10

[69] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 574–591. 11

[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019. 13

Tuo Cao received his BSc degree from the school of Electronic Engineering at University of Electronic Science and Technology of China(UESTC) in 2015 and received his MSc degree for the School of Electronic Information at Wuhan University in 2018. Currently, he is working toward his PhD degree in the School of Computer, Wuhan University, China. His research interests are 3D vison, SLAM and object pose estimation.

Wenxiao Zhang is now a research fellow at Singapore University of Technology and Design. He obtained his Ph.D. degree in the School of Computer Science, Wuhan University, China. Before that he received his M.E. degree from Huazhong University of Science and Technology and B.E. degree from Shandong Normal University 2016 and 2014, respectively. His research interests include point cloud analysis, such as point cloud completion and point cloud based retrieval for place recognition.

Yanping Fu received his Ph.D. degree at School of Computer Science Department of Wuhan University in June 2020, and received the B.Sc. degree from Changchun University in 2008, and the M.Sc. degree from the Yanshan University in 2011. Currently, he is a lecturer in the School of Computer Science and Technology, Anhui University, China. His research is focused on 3D reconstruction, texture mapping and image processing.

Shengjie Zheng received his B.S. degree in software engineering from the School of Computer Science, Dalian Maritime University, Dalian, in 2020. He is working toward his M.S. degree in computer science and technology at the School of Computer Science, Wuhan University, Wuhan. His research interests are monocular depth estimation and 3D vision.

Fei Luo received his Ph.D. degree in computer science and technology from Wuhan University, Wuhan, in 2011. He worked as a research assistant at the School of Computer Engineering of Nanyang Technological University, Singapore, in 2009. From 2011 to 2013, he worked as a postdoctor at Human Polymorphism Study Center, Paris, France. He is now an assistant professor at the School of Computer Science, Wuhan University, Wuhan. His research interests include computer vision, computer graphics and data mining

Chunxia Xiao received his B.S. and M.S. degrees in mathematics from Hunan Normal University, Changsha, in 1999 and 2002 respectively. He received his Ph.D. degree in applied mathematics from the State Key Lab of CAD&CG of Zhejiang University, Hangzhou, in 2006. He became an assistant professor at Wuhan University in 2006, and became a professor in 2011. From October 2006 to April 2007, he worked as a postdoctor at the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. During February 2012 to February 2013, he visited University of California, Davis. Currently, he is a professor at the School of Computer Science, Wuhan University, Wuhan. His main interests include computer graphics, computer vision, virtual reality and augmented reality. He is a member of CCF and IEEE.