PAPER

6D pose estimation of object based on fused region-level feature in cluttered scenes

IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **34** (2023) 075402 (12pp) https://doi.org/10.1088/1361-6501/acc603

6D pose estimation of object based on fused region-level feature in cluttered scenes

Xiangpeng Liu, Huiping Duanmu, Kang An*, Wancheng Wang, Yaqing Song, Qingying Gu, Bo Yuan and Danning Wang

College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, People's Republic of China

E-mail: ankang526@foxmail.com

Received 15 December 2022, revised 5 March 2023
Accepted for publication 21 March 2023
Published 18 April 2023

Abstract

RGB-based methods have certain disadvantages because they lack 3D information and cannot cope with the pose estimation problem in cluttered scenes. Therefore, we propose a region-level pose estimation network that uses RGB-D images. We first extracted the objects' color features and geometric features by convolutional neural networks (CNN) and PointNet, respectively, and then performed feature fusion. The fused features were fed into a region-level feature extraction network to obtain the region-level features, which extracted the local geometry features from the point cloud and learned the point set's semantic information. We used the output of the region-level feature extraction network to perform region-level pose estimation, then selected the pose with the highest confidence level as the output and iteratively optimized the pose to obtain the final results. The experimental results showed that the proposed solution performed well on the LINEMOD data set, which verified the effectiveness of the proposed method in the pose estimation problem and the algorithm's robustness in severely cluttered scenes.

Keywords: object pose estimation, point cloud, region-level feature, feature fusion, iterative optimization

(Some figures may appear in colour only in the online journal)

1. Introduction

The application of several techniques are based on the results of 6D pose estimation, including robotic manipulation [1], autonomous driving [2], and augmented reality [3]. In most realistic scenes, the pose estimation algorithm needs to be able to deal with objects of various shapes and textures, and have sufficient robustness under the circumstances of cluttered scenes, sensory noise, and changing illumination conditions. RGB-based methods [4–6] are extremely sensitive to occlusion and cannot be applied to the pose estimation problem in cluttered scenes. The emergence of the RGB-D camera has enabled great progress in pose estimation methods for textureless objects, which are more accurate than the methods based solely on RGB data [7]. Compared with RGB images, RGB-D images contain depth information. By converting this information into point clouds, we can obtain geometric information describing the spatial geometric structure of objects, which can lead to cursory position information on objects and is more suitable for cluttered scenes. Therefore, we can use RGB-D images to obtain the color information and geometric information of the object, then fuse them, and finally design a region-level feature extraction network to extract the region-level information for pose estimation. This method makes full use of the color and geometric information of the object and performs region-level estimation, which can effectively cope with cluttered scenes. The fusion of features and the design of the

* Author to whom any correspondence should be addressed.

region-level feature extraction network are the two most critical technical points in this method.

For texture-rich objects, traditional 2D–3D correspondence methods are commonly used for pose estimation [8]. Three-dimensional models of the objects are projected from different perspectives and several RGB image templates are obtained. At the same time, the 2D–3D correspondence between the 3D points of the model and the 2D pixels of the RGB images is established. After capturing the RGB image of an object from a single perspective, the feature points, such as scale-invariant feature transform (SIFT) [9], speeded-up robust features (SURF) [10], and Oriented FAST and Rotated BRIEF (ORB) [11], can be extracted from the image, and this image is matched with template images to select the template image with the highest matching ratio. Afterwards, the Perspective-n-Point (PNP) [12] algorithm is applied to calculate the current pose according to the established 2D–3D correspondence. These methods become less effective in textureless scenarios.

With the development of deep learning and its extensive application in the fields of image classification and object detection, researchers have begun to introduce deep learning into the field of pose estimation. These approaches include two main categories. The first category uses RGB images as the input data and directly obtains the 6D pose of the object with convolutional neural networks (CNNs), such as PoseNet [13], PoseCNN [14], etc. However, these methods need the hyperparameters to be accurately selected in the loss function to realize more effective detection. The other category takes RGB-D images as the data source, and estimates the 6D pose of the object with the fused RGB-D data. Taking depth information as the fourth channel of RGB images, Michel *et al* [15] fused RGB-D data and processed the data with a CNN. Wang *et al* [16] put forward the DenseFusion network, which first converted a depth map into a point cloud, extracted the geometric features with the PointNet [17] network, then fused the color features with the geometric features, and finally estimated the object's pose. Although this method takes account of both the color information and the depth geometry information of the image, it shows poor ability to extract the regional features because feature extraction based on PointNet is limited to a single point of the point cloud data. The performance of pose estimation with these features is not satisfactory in severely cluttered scenes.

In this study, we proposed a pixel-level feature fusion network and a region-level feature extraction network that can thoroughly utilize the color information and depth geometry information from RGB-D images and obtain a region-level feature representation for region-level pose estimation. The method first uses a CNN to extract the color features of the image, then fuses the color data with the point cloud data processed by PointNet. We used a region-level feature extraction network to obtain region-level features for region-level pose estimation. This region-level feature extraction network can form a graph structure by establishing the topology between points and learning the semantic information of the point set by dynamically updating the graph structure between the layers to obtain region-level features with strong expressive power. The inspiration behind this method mainly came from two sources. One is that humans can perceive objects through colors and 3D information. The other is that we can distinguish an object's pose by its local features [18]. On the basis of these two facts, we fused the color information and 3D point cloud information to create a network with a real 3D scene. The real 3D scene was divided into several regions to extract the features of each region, with the aim of carrying out region-level pose estimation. We then outputted the region-level pose prediction with the highest confidence. Finally, a pose optimization module was used to improve the accuracy. In this scheme, the model we used can make full use of objects' color, depth information, and local features, which is of great significance for pose estimation in severely cluttered scenes. The experimental results on the ModelNet40 [19], ShapeNet [20], Stanford 3D indoor scene (S3DIS) [21], and LINEMOD [22] data sets demonstrated that our network performs well at local feature extraction and pose estimation.

In summary, the contributions of this work are as follows:

(a) We proposed a network which can fuse the color features and depth point cloud features of RGB-D images. The network merges the color features and the depth point cloud features into a single point cloud, which can yield dense and more efficiently fused features.
(b) The fusion features were divided into regions and then region-level feature extraction was performed to obtain region-level features to cope with the problem of pose estimation in severely cluttered scenes. A region-level feature extraction network was proposed, which can learn the semantic information of the fused features better and obtain more expressive region-level features than PointNet++. We utilized this scheme to carry out pose estimation.
(c) The region level feature extraction proposed in this study was verified to outperform PointNet++ and dynamic graph CNN (DGCNN) on the ModelNet40, ShapeNet, and S3DIS data sets, and excellent results were obtained on the LINEMOD data set.

2. Related work

2.1. RGB-based methods

The most popular RGB-based methods [4, 6, 22–26] attempt to build 2D–3D correspondences between the 2D image pixels and 3D mesh vertexes, which are then leveraged to calculate the 6D pose through a perspective–n–point method [12]. The 2D–3D correspondences are either sparse or dense. For instance, pixel-wise voting network (PVNet) [23] regresses pixel-wise vectors pointing to the key points and uses these vectors to vote for the key points' locations. HybridPose [25] utilizes a hybrid intermediate representation to express different geometric information in the input image, including the key points, edge vectors, and symmetry correspondences. Hu *et al* [27] introduced a deep architecture that takes a group of candidate correspondences for each 3D key points as input, directly returning to the 6D pose.

2.2. Voting-based methods

Where there is occlusion in complex scenes, only local information about the object can be acquired, which limits the accuracy of pose estimation. The voting-based methods aim to obtain an object's pose estimation results by voting on the pose of each image patch, and these methods perform well in severely cluttered scenes and occlusions. Drost *et al* proposed the point pair feature (PPF) [28] algorithm. The traditional point descriptor-based method relies on local information around a point; however, the PPF method uses point pairs of features to create global descriptors and then uses a speedy voting scheme for local matching of the model. Point pair features contain information about the distance and normals of two arbitrary 3D points, making PPF a very effective method for 6D pose estimation. Vidal *et al* proposed a flexible approach that can handle textured and untextured objects, which possesses a learnable intermediate structure that is able to conduct dense 3D object coordinate label matching [29]. In 2019, Wang proposed the DenseFusion [16] neural network. DenseFusion uses a heterogeneous network to process the RGB images and point clouds separately to capture the color features and point cloud features, then performs pixel-level feature fusion. Each fused feature gives a prediction and its confidence level; ultimately, the prediction with the highest confidence level is selected as the pose estimation output.

2.3. RGB-D-based methods

The classical pose estimation methods using RGB-D data as input extract features by hand and then perform the associated grouping and hypothesis validation [7, 30, 31]. Manual feature extraction cannot perform well under the circumstances of severe occlusions, cluttered scenes, and environmental changes, so models based on CNNs for direct object pose prediction emerged [14, 19, 32–38]. They either estimated the 6D pose from the color data only or fused depth images as an additional channel of color images [33] before making predictions, and subsequently applied the depth information in the pose optimization stage. DenseFusion [16] processed color and depth information using different networks, and then combined the two at the pixel level to perform pixel-level pose estimation, thus achieving the most advanced performance at that time.

Inspired by DenseFusion, we estimated the 6D poses of objects by fusing the color and depth information. The difference is that we extracted the color and depth information with a CNN and PointNet, respectively. We then merged the color features into the point cloud and used our proposed region-level feature extraction network, the design of which was based on PointNet++ [39] and DGCNN, for processing the point cloud generated in the previous step to extract more expressive region-level features. Next, we combined these region-level features with the global features to perform region-level pose estimation. We demonstrate that the novel region-level feature fusion and extraction scheme proposed in this study outperforms DenseFusion.

3. Proposed methods

In this study, we proposed a 6D pose estimation method. Given an image, the task of 6D pose estimation is to estimate the orientation and translation of the object in 3D space. The specific output is represented by $[R|t]$, where R denotes the 3D rotation and t represents the 3D translation of the object relative to the camera's coordinate system. We assumed that the problem of adverse conditions can be solved by fusing the color and depth information of the image, followed by region-level feature extraction. The key techniques are fusing the features and extracting the region-level features properly and effectively. We combined the extracted color information into the corresponding point clouds and then performed region-level segmentation on the resulting point clouds, after which we used the proposed region-level feature extraction network for capturing the region-level features to implement the key techniques. Our extracted region-level features contained both color and geometric information. The region-level information has a powerful capability for feature representation, which is conducive to solving the problem of estimating an object's pose in the case of textureless objects and heavily cluttered scenes. After obtaining the estimation results with the highest confidence, we used a pose estimation optimization module [16] to improve the accuracy of the results.

3.1. Overview of the architecture

Figure 1 illustrates the overall architecture of the proposed method. The architecture contains three main parts: the first part was used to obtain the masked bounding boxes of each known object category with a semantic segmentation network. We then cropped the color image patch and the object point cloud of each object according to the results of segmentation. In the second part, the color features were extracted from the CNN and the geometric features from the PointNet network. Afterwards, pixel-level feature fusion was executed. We processed the fused point clouds with the network based on PointNet++ and DGCNN to obtain multiple region-level features and one global feature. After that, these region-level features were combined with the global features separately. The combined features were used for pose estimation, outputting the most confident estimation result. Finally, we utilized a pose refining module to optimize the result.

3.2. Semantic segmentation

In the first part of our method, we used a semantic segmentation network to detect the objects of interest in the image. The output of this network can provide more information about the target and can handle the occlusion problem effectively. Here, we used the segmentation network in PoseCNN [14]. The segmentation network has an encoder–decoder architecture that takes an image as input and outputs a segmentation map of $N+1$ channels. The first channel describes the background, and the remaining N channels describe the N known classes of objects.

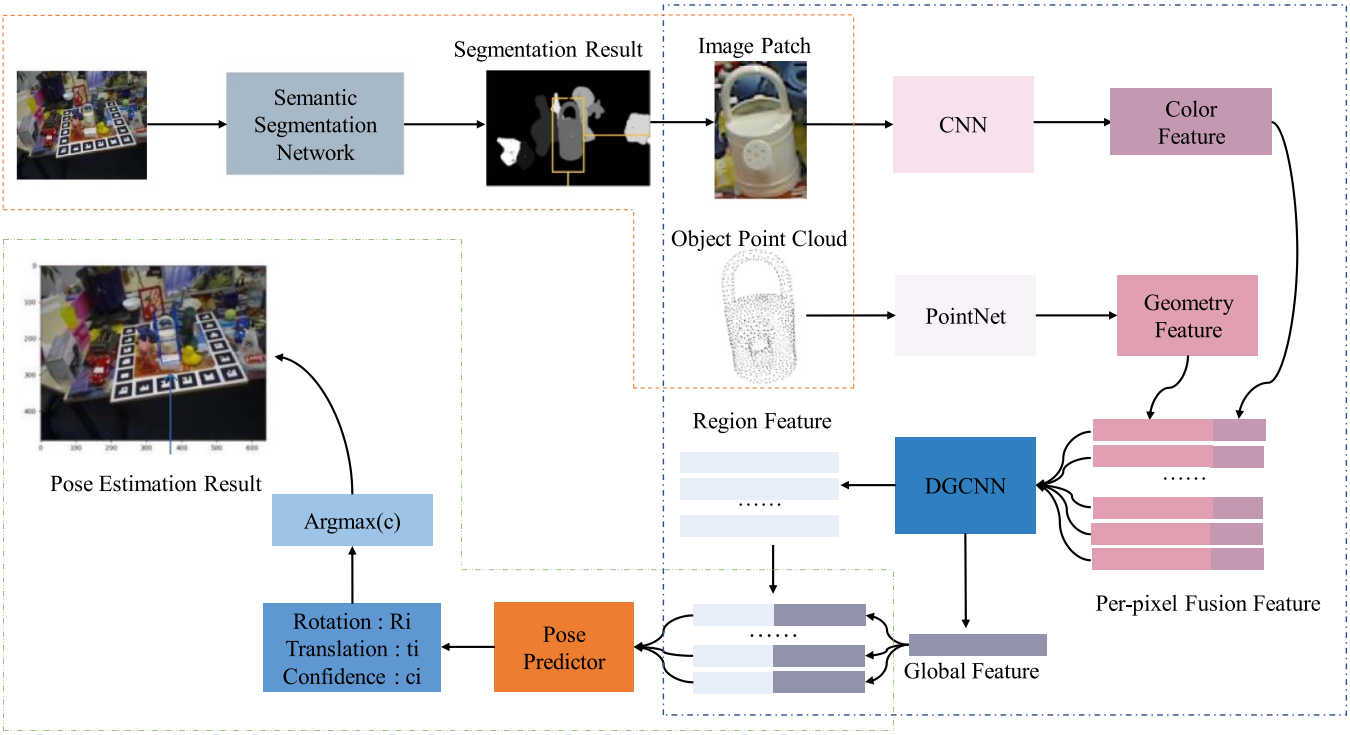

Figure 1. Overview of the proposed 6D pose estimation model.

3.3. *Feature extraction and fusion*

3.3.1. *Region-level feature extraction*

3.3.1.1. Color feature extraction. Each channel is a binary mask where active pixels depict objects of each of the N possible known classes. Based on the mask and the external rectangle which we calculated by the edges of the mask, the original image was cropped into an image patch with a size of $h \times w \times 3$. An image patch only contained one object. The RGB image contained more information than the depth map. Due to the sparsity of point cloud, the point cloud processing network was not effective enough in extracting the features. The color map could not be fully utilized by directly merging the color map into the depth map and feeding it into the point cloud processing network. Therefore, we designed an encoder–decoder CNN for pre-feature extraction of the color image patches. The network mainly consisted of ResNet18 and pyramid scene parsing network (PSPNet). As we also fused the color features into the point cloud, the size of the color feature maps had to be the same as that of the original image patch. Consequently, we used an upsampling module after PSPNet to scale up the obtained feature maps, as shown in figure 2. The cropped image block was fed into the color feature extraction network, and then we obtained the feature maps with a size of $h \times w \times D_{\text{rgb}}$.

3.3.1.2. Geometric feature extraction. The depth image was cropped on the basis of the mask obtained through semantic segmentation, and then the cropped result was transformed into a point cloud with a size of $N \times 3$, where N represents the number of points in the point cloud. To use the geometric

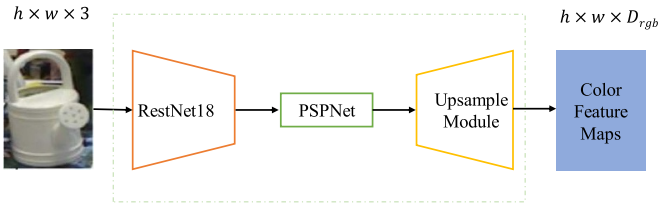

Figure 2. Feature extraction network.

information in the depth map more thoroughly, we extracted the geometric features of the point cloud using PointNet and obtained feature maps with a size of $N \times D_{\text{pc}}$. PointNet can be used to process point clouds and thus to obtain dense point-by-point features, which are conducive to the fusion of the pixel-level color features and geometric features.

3.3.2. Pixel-level feature fusion. After the above feature extraction process, we obtained dense color features and the point cloud's geometric features. Because of occlusion of the objects in cluttered scenes and semantic segmentation errors, these dense features may contain the points of other objects or backgrounds. Direct global fusion of the color features and geometric features can degrade the performance of the estimation. Therefore, we used pixel-level fusion, which can effectively fuse dense color and geometric features, which is especially suitable for estimating an object's pose in the case of occlusions and classification errors. There were N points in the point cloud. On the basis of the correspondence between the color pixel points of the original color image and the point cloud, we selected N pixel points from the color features and

fused them with the point cloud for feature fusion. The pixel-level fusion network outputted a point cloud with a size of $N \times (D_{\text{pc}} + D_{\text{rgb}})$.

3.3.3. Region-level feature extraction. In cluttered scenes, objects may be occluded; therefore, the ability to extract the features of objects is easily affected, which leads to inaccurate estimation of the object's pose. In this case, we can use multi-scale receptive fields to extract different region-level features. The small-scale perceptual field can extract the region-level features of the object, and the large-scale field can extract the global features. The different region-level features obtained are then combined with the global features and fed to the positional estimation network for positional estimation. In this way, the network can estimate an object's pose by extracting the local features in a severely cluttered scene, which can effectively solve the pose estimation problem in complex environments. Therefore, designing an effective region-level feature extraction network is of top priority.

To extract more effective region-level features, we proposed a custom region-level feature extraction network based on PointNet++ and DGCNN for processing point clouds with a size of $N \times (D_{\text{pc}} + D_{\text{rgb}})$ generated after feature fusion. If the dimension of the feature is too high, the network may learn some meaningless messages. Then the performance of the network will be weakened. Here we chose D_{pc} and D_{rgb} as both 64. PointNet++ consists of several hierarchies [39], each of which has three parts: the sampling layer (sampling), the grouping layer (grouping), and the PointNet [17] layer. The sampling and grouping layers are used for integrating the local information, and PointNet is used as the feature extractor to obtain features from the local area of the point set. PointNet++ divides the point set into overlapping local regions to extract the region-level features. However, PointNet uses pooling to extract global features and thus the representation of the output is not strong.

We introduced DGCNN to extract the features more efficiently. DGCNN [40] is a network with an integrated convolutional module (EdgeConv) as the core. It models the relationship between points in the point cloud so that the network can learn the local and global features while learning the information about each point. EdgeConv first calculates the edge features e_{ij} based on the points X_i and X_j in the point cloud, and then performs an aggregation operation on these edge features to solve x_i'.

The edge features (e_{ij}) are calculated as follows:

$$\begin{cases} h_\theta : \mathbb{R}^F \times \mathbb{R}^F \to \mathbb{R}^{F'} \\ e_{ij} = \mathrm{h}_\theta(X_i, X_j) \end{cases} \tag{1}$$

where h_θ is a nonlinear activation function composed of the learnable parameters θ. In this study, we applied the following expression of the function h_θ:

$$\mathrm{h}_\theta(X_i, X_j) = \tilde{h}_\theta(X_i, X_j - X_i). \tag{2}$$

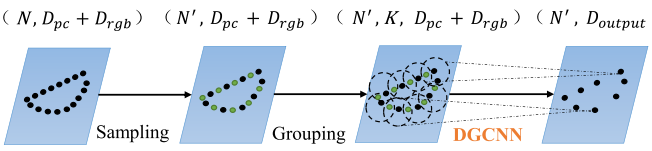

Figure 3. Local feature extraction of the point cloud.

x_i' is obtained by adding a channel-level symmetric aggregation operation o.

$$x_i' = o_{j:(i,j)\in\varepsilon} h_\theta(X_i, X_j). \tag{3}$$

DGCNN [40] can dynamically update the graph structure between the layers to better learn the semantic information of the point set. The EdgeConv module is able to extract the features of the local shape of the point cloud effectively and maintain an invariant alignment. Therefore, we used the EdgeConv module in DGCNN instead of the PointNet structure in PointNet++, as shown in figure 3.

Taking a point set of size $N \times (D_{\text{pc}} + D_{\text{rgb}})$ as input, the sampling layer uses the farthest point sampling algorithm to select N' points from the input points. These N' points are the points furthest from each other (in the metric 3D distance instead of $(D_{pc} + D_{rgb}) - dimension$ distance) and they define the center of the local regions. The grouping layer constructs the local region sets by the ball query method, which finds all the points within the radius of the query point (region center) and selects the k nearest points from the query point to generate the local region's point sets. If the number of points is less than k, in-ball points are reused to supplement them. K-nearest neighbors (KNN) is not used to find a fixed number of neighboring points because the local neighborhood of the query sphere ensures a fixed regional scale. The local area features are more generalized and universal in space, which is significant for extracting the local area's features. After grouping, we can acquire a point set with a size of $N' \times K \times (D_{\text{pc}} + D_{\text{rgb}})$, as shown in figure 3. The DGCNN layer can encode the N' local region point set as a feature vector with a size of $N' \times D_{\text{output}}$. Therefore, the size of the output data is $N' \times D_{\text{output}}$, while preserving the 3D coordinates of the N' center of mass. In our proposed structure, the local regions that have been sampled and grouped are divided again in the DGCNN layer to extract the higher-level features. We use two abstraction levels to help us extract various levels of region-level features, which facilitate the successive aggregation of the local point cloud's regions into larger regions. The penultimate layer of this network outputs N_{patch} local feature vectors with a size of D_{local}, i.e. the original point cloud is partitioned into N_{patch} local regions, and the last layer outputs global feature vectors with a size of D_{global}. The improved model has a stronger capability of extracting region-level features than PointNet++.

We can then merge these N_{patch} feature vectors with the global feature vectors with a size of D_{global} and finally obtain N_{patch} hybrid feature vectors with a size of $D_{\text{local}} + D_{\text{global}}$. The final feature vector contains both the global and local features and has excellent ability to represent the features.

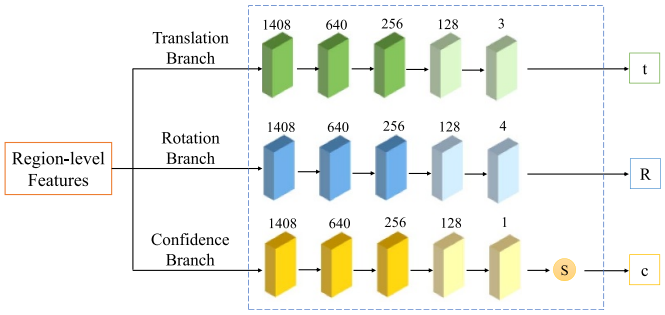

Figure 4. Structure of the pose estimation network.

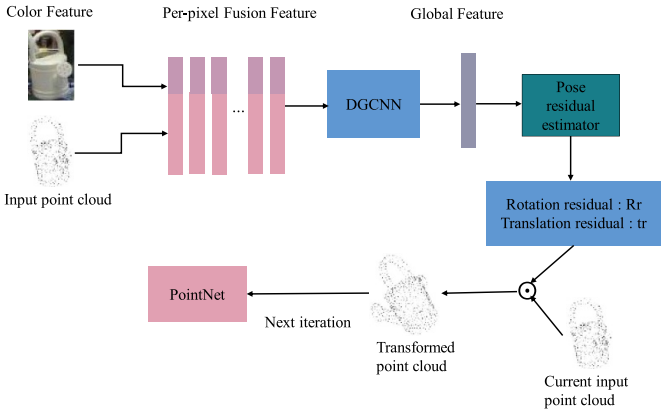

Figure 5. Optimization of pose estimation.

3.4. Pose estimation and refinement

After the feature fusion operation, we fed the obtained features into the pose estimation network. As shown in figure 4, the pose estimation network has three main branches, namely predict rotation, translation, and confidence. We chose the one with the highest confidence as the output of the network [16].

After obtaining the pose estimation results, we proceeded to optimization of the result. Since the traditional iterative closest point (ICP) [41] algorithm cannot meet the requirements of real-time application, here, we used a neural network-based iterative optimization module that can optimize the pose estimation results quickly and robustly. The goal of the network was to iteratively reduce the error of the network's pose estimation and improve the prediction results of pose estimation, as shown in figure 5. The original point cloud of the object was transformed using the pose obtained from the backbone network. The geometric features of the inverted point cloud were extracted using PointNet and fused with the original color features, and the fused features were used to estimate the residuals of the poses. After k iterations, we merged the obtained pose residuals with the initial positional prediction results to obtain the final pose prediction results. The calculation formula is as follows:

$$\hat{p} = [R_k|t_k] \cdot [R_{k-1}|t_{k-1}] \cdots [R_0|t_0]\,. \tag{4}$$

3.5. Loss function

Having defined the overall network structure, we turned our attention to the learning objectives of the network. We defined the pose estimation loss as the distance between the sampled points on the model of the object in the ground truth pose and the corresponding points on the same model transformed by the predicted poses. Specifically, an asymmetric object at this distance is expressed by the following formula:

$$L_i^p = \frac{1}{M}\sum_{j=1}^{M}\left\|(Rx_j + t) - (\hat{R}_i x_j + \hat{t}_i)\right\|. \tag{5}$$

For symmetric objects, the formula is

$$L_i^p = \frac{1}{M}\sum_{j=1}^{M}\min_{0<k<M}\left\|(Rx_j + t) - (\hat{R}_i x_j + \hat{t}_i)\right\| \tag{6}$$

where M denotes the number of sampling points; x_j denotes the j-th point of M; $p = [R|t]$, where R and t represent the true rotation and translation; and $\hat{R}_i$ and $\hat{t}_i$ represent the rotation and translation generated from the fused embedding of the i-th dense pixel.

We wanted our network to learn to balance the confidence among the predictions for the dense pixels. To that end, we weighted the loss per dense pixel with the dense pixels' confidence, and added a second confidence regularization term. The final loss function of this network is shown in the following formula:

$$L = \frac{1}{N}\sum_{i=1}^{N}(L_i^p c_i - w\log(c_i)) \tag{7}$$

where N is the number of randomly sampled features of dense pixels from the p elements of the segment, w is a balancing hyperparameter, and c_i is the confidence of each feature vector. Intuitively, low confidence will result in low pose estimation loss but would incur a high penalty from the second term, and vice versa.

4. Experimental

In the experiments, we aimed to answer two questions. (a) Does the proposed region-level feature network produce good results when extracting features? (b) Can the designed pose estimation network deal with the estimation problem in severely cluttered scenes?

For the first question, the ModelNet [19], ShapeNet [20], and S3DIS [21] data sets were applied in the tests, which correspond to three scenarios of classification, part segmentation, and scene segmentation, respectively. For the second question, we tested our model on the LINEMOD [22] data set, which is one of the most widely used data sets, and thus we could compare our method with existing methods.

4.1. Data sets

ModelNet data set: the data set is a 3D image classification data set containing 40 classes of CAD models, including 9843 training data and 2468 test data.

ShapeNet data set: the data set is a part segmentation data set containing 16 881 3D models in 16 categories, and most object models are divided into two to five parts.

S3DIS data set: the data set is a scene segmentation data set containing 3D scans of six regions containing 271 rooms. The room types include conference rooms, photocopy rooms, foyers, restaurants, and toilets. Each point in the room is marked as one of the 13 object types (tables, chairs, flooring, walls, etc).

LINEMOD data set: there are 13 objects that are textureless in this data set, and it contains most of the challenges faced in 6D pose estimation, including textureless and severely cluttered scenes.

4.2. Evaluation metrics

Regarding the classification test on the ModelNet data set, we used the average accuracy (*AA*) and overall accuracy (*OA*) rate for our evaluation. The calculation formulas are as follows:

$$AA = \frac{SACP}{QC} \tag{8}$$

$$OA = \frac{PC}{TN} \tag{9}$$

where *SACP* is the sum of the accuracy of each category of prediction, *QC* stands for the quantity of categories, *PC* denotes the predicted number of correct detection, and *TN* is the total number.

On the ShapeNet data sets, Mean Intersection over Union (*MIOU*) was used to evaluate the effect of the local point feature extraction network on the part segmentation task. The Intersection over Union (*IOU*) of the predicted part and the true part of each object in a certain class was calculated first, then the *MIOU* was obtained by averaging the *IOU* of each class of objects. The formula for the *MIOU* is

$$MIOU = \frac{1}{k}\sum_{i=1}^{k}\frac{ROI_{\mathrm{p}i} \cap ROI_{\mathrm{G}i}}{ROI_{\mathrm{p}i} \cup ROI_{\mathrm{G}i}} \tag{10}$$

where $ROI_{\mathrm{p}i}$ represents the i-th predicted part of the element, $ROI_{\mathrm{G}i}$ denotes the i-th real (true) graphic, and k is the number of object categories.

On the S3DIS data sets, *MIOU* was used to evaluate the results of the local point feature extraction network for the scene segmentation task.

On the LINEMOD data set, we used the average distance measurement ADD [42] and 2D reprojection [23] for the measurement. ADD indicates the average distance between the corresponding point in the 3D models of the real pose and the estimated pose, namely

$$ADD = \frac{1}{m}\sum_{x\in M}\left\|(Rx+t)-(\hat{R}x+\hat{t})\right\| \tag{11}$$

where M represents the collection of 3D model points, m is the number of points in the collection, R denotes the 3D rotation, t represents the 3D translation of the object relative to the camera's coordinate system, and $\hat{R}$ and $\hat{t}$ are the estimated rotation and transition. For symmetrical objects, when objects are in different poses, they may appear the same in the image when they are photographed from the same angle. In order to solve this problem, we employed the distance of nearest point to calculate the average distance [42]:

$$ADD-S = \frac{1}{m}\sum_{x_1\in M}\min_{x_2\in M}\left\|(Rx_1+t)-(\hat{R}x_2+\hat{t})\right\|. \tag{12}$$

If the average distance is less than the predefined threshold, the estimated 6D pose is considered to be correct. In general, the threshold is set to 10% of the 3D model's diameter.

The 2D reprojection measurement is calculated as

$$\mathrm{proh.2D} = \frac{1}{m}\sum_{x\in M}\left\|K(Rx+t)-K(\hat{R}x+\hat{t})\right\| \tag{13}$$

where K is the camera's intrinsic parameters. When the 2D reprojection is used, the estimated pose is considered to be correct if the average distance between the 2D projection of the 3D model for the real pose and the estimated pose is less than five pixels.

4.3. Implementation

Three experiments that verified the effect of regional features extraction were conducted using an AMD Ryzen 74 800 H with Radeon Graphics 2.90 GHz and a NVIDIA GeForce RTX 2060 GPU platform, and PyTorch was used as the deep learning framework. The following configuration was designated for all three experiments: the number of epochs was set to 250, the batch size was set to 32, and the learning rate was set to 0.001. Adam was selected as the optimizer of the algorithm. The dynamic momentum was set to 0.9, and the learning decay rate was 0.7. For the part segmentation test with ShapeNet, we used the data set division method proposed by Chang *et al* [43]. For the scene segmentation test with S3DIS, the 3D scanning results of the room were divided into blocks with an area of 1 m × 1 m, and the network was trained to predict the type of each block. The number of dimensions for each input point was set to 9, including the *XYZ* coordinates, the RGB color features, and the normalized position. In total, 4096 block inputs were selected randomly during the training process. In the test, all blocks were trained and tested via the k-fold strategy.

For the pose estimation test with the LINEMOD data set, we trained the network on a computer with an Intel Core i7-9700 K@3.60 GHz CPU and a single NVIDIA GeForce RTX 1080TI GPU. The following configuration was used for all three experiments: the number of epochs was set to 500, the batch size was set to 4, the learning rate was set to 0.0001, the decay margin was set to 0.016, and the refining margin was set to 0.013. Adam was selected as the optimizer of the

Table 1. Comparison of the models' classification performance using the ModelNet data set.

Model	Input	Average accuracy (%)	Overall accuracy (%)
PointNet	Point	86.2	89.2
PointNet++	Point	90.7	91.9
DGCNN	Point	90.2	92.9
Proposed	Point	90.9	93.8

Table 2. Comparison of the scene segmentation results using the S3DIS data set.

Model	Mean IOU (%)	Overall accuracy (%)
PointNet	20.1	53.2
PointNet++	47.6	78.5
DGCNN	56.1	84.1

algorithm. The datas et was divided according to the settings in [16], in which the ratio between the training and testing images was 15:85. For the LINEMOD data set, 2373 and 13 404 images were used for training and testing, respectively. We first trained the semantic segmentation network using cross-entropy loss function. Then, we optimized the rest of the main network using the loss function described in equation (7). The color feature extraction network output a 64-dimensional color feature map, and the geometric feature extraction network also output a 64-dimensional geometric feature map, which means that each pixel was represented by a 64-dimensional feature vector. We fused the two 64-dimensional vectors and sent them to a region-level feature extraction network based on PointNet++ and DGCNN, which had three set abstraction levels and finally output eight region-level feature vectors of dimension 1664, including 128-dimensional local region features, 512-dimensional local region features, and 1024-dimensional global features. After that, the pose estimation output the pose estimation results and confidence of each feature vector. The pose optimization module was not trained with the main network because of the problem of convergence. Therefore, we had to train the main network until convergence first, then set the main network as fixed and started training the pose optimization module.

4.4. Classification experiment using the ModelNet data set

In table 1, the region-level feature extraction network proposed in this study is compared with previous models. The experiments were performed on the ModelNet data set. PointNet, PointNet++, DGCNN, and the proposed region-level feature extraction network in this study all used point cloud as input. Among these models, the *AA* and *OA* of the proposed region-level feature extraction network achieved the best results. Compared with the PointNet model, the proposed model was better by 4.9% in terms of the *AA* and 4.6% in terms of the *OA*. With PointNet++ as the comparison network, the *AA* of the proposed model was better by 2% and the *OA* increased by 1.9%. Compared with the DGCNN model, the proposed model increased by 0.7% in terms of the *AA*, and the *OA* increased by 0.9%.

4.5. Part segmentation experiment using the ShapeNet data set

In this experiment, the performances of the proposed region-level feature extraction network and other models on ShapeNet data set were compared, and the results are listed in table 2. The *MIOU* of the proposed model was the highest. Our method achieved an accuracy of 85.6%, outperforming the other five methods, and even outperforming DGCNN and PointNet++ by 0.4% and 0.5%, respectively.

4.6. Scene segmentation experiment using the S3DIS data set

The scene segmentation tests were conducted on the S3DIS data set, and the performance of the proposed region-level feature extraction network was compared with that of other models. The corresponding results are shown in table 3. *OA* of the proposed approach reached up to 84.8%, which is 31.6%, 6.3%, and 0.7% higher than PointNet, PointNet ++, and DGCNN, respectively. Among these models, the *MIOU* of the proposed region-level feature extraction network achieved 56.7%, which was similar to that of the other models. The *MIOU* of the proposed model was 36.6%, 9.1%, and 0.6% higher than that of PointNet, PointNet++, and DGCNN, respectively.

4.7. Pose estimation experiment using the LINEMOD data set

In table 4, the ADD-S measurement results of the proposed network and the other approaches on the LINEMOD data set are compared. Two groups of tests were conducted based on the input type, namely RGB and RGB-D images. On the RGB-D data set, the method proposed in this article achieved an accuracy of 96.5%, which is 7.9% higher than the result of the PoseCNN model and 2.2% higher than that of the DenseFusion model.

The results of the 2D projection measurement test of the proposed pose estimation approach and other benchmark methods on LINEMOD data set are listed in table 5.

As in the last experiment, we compared the performance of the models based on the RGB and RGB-D images. With the RGB-D image as the data source, the accuracy of the proposed approach reached 97.82%, which is 4.22% higher than that of DenseFusion.

It is worth noting that the proposed method performed better than DenseFusion in both the ADD-S measurement tests and 2D projection measurement tests. This verifies the effectiveness of the proposed model.

Images of the pose estimation results are shown in figure 6. The green bounding box represents the ground truth pose, and the blue bounding box shows the predicted pose.

Table 3. Parts segmentation test using the ShapeNet data set.

Object category	# Shapes	PointNet (%)	PointNet++ (%)	Kd-Net (%)	PCNN (%)	DGCNN (%)	Proposed (%)
Mean	/	83.7	85.1	82.3	85.1	85.2	85.6
Aero	2690	83.4	82.4	80.1	82.4	84	84.1
Bag	76	78.7	79	74.6	80.1	83.4	86.3
Cap	55	82.5	87.7	74.3	85.5	86.7	86.6
Car	898	74.9	77.3	70.3	79.5	77.8	78.8
Chair	3758	89.6	90.8	88.6	90.8	90.6	90.6
Earphone	69	73	71.8	73.5	73.2	74.7	76.7
Guitar	787	91.5	91	90.2	91.3	91.2	92.1
Knife	392	85.9	85.9	87.2	86	87.5	87.4
Lamp	1547	80.8	83.7	81	85	82.8	85.3
Laptop	451	95.3	95.3	94.9	95.7	95.7	96.3
Motor	202	65.2	71.6	57.4	73.2	66.3	71.7
Mug	184	93	94.1	86.7	94.8	94.9	95.2
Pistol	283	81.2	81.3	78.1	83.3	81.1	82.4
Rocket	66	57.9	58.7	51.8	51	63.5	61.6
Skateboard	152	72.8	76.4	69.9	75	74.5	79.1
Table	5271	80.6	82.6	80.3	81.8	82.6	83.2

Note: PCNN: pulse coupled neural networks.

Table 4. Comparison of the pose estimation results using the LINEMOD data set.

	RGB			RGB-D				
Object category	BB8 [24] (%)	PVNet [23] (%)	HybridPose [25]	PoseCNN [14] (%)	AAE [44] (%)	SSD-6D [20] (%)	DenseFusion [16] (%)	Proposed (%)
Ape	40.4	43.6	63.1	77	24.35	65	92.3	94.5
Bench vise	91.8	99.9	99.1	97.5	89.13	80	93.2	93.9
Cam	55.7	86.8	90.4	93.5	82.1	78	94.4	96.4
Can	64.1	95.4	98.5	96.5	70.82	86	93.1	96.5
Cat	62.6	79.3	89.4	82.1	72.18	70	96.5	98.3
Drill	74.4	96.4	98.5	95	44.87	73	87	94.8
Duck	44.3	52.5	65.0	77.7	54.63	66	92.3	96.6
Eggbox	57.8	99.1	100.0	97.1	96.62	100	99.8	99.7
Glue	41.2	95.6	98.8	99.4	94.18	100	100	98.5
Hole punch	67.2	81.9	89.7	52.8	51.25	49	92.1	96.0
Iron	84.7	98.8	100.0	98.3	77.86	78	97	95.9
Lamp	76.5	99.3	99.5	97.5	86.31	73	95.3	96.5
Phone	54	92.4	94.9	87.7	86.24	79	92.8	96.9
Average	62.7	86.2	91.3	88.6	71.58	79	94.3	96.5

Note: BB8: 8 corners of the bounding box. AAE: Augmented Autoencoders. SSD: single-shot detection

Table 5. Comparison of the 2D projection measurement results using the LINEMOD data set.

	RGB			RGB-D	
Object category	Brachmann [45] (%)	BB8 [24] (%)	Tekin [22] (%)	DenseFusion [16] (%)	Proposed (%)
Ape	33.2	95.3	92.10	96.85	99.71
Bench vice	64.8	80	95.06	88.26	94.95
Cam	38.4	80.9	93.24	93.82	98.04
Can	62.9	84.1	97.44	96.06	97.74
Cat	42.7	97	97.41	96.11	99.3
Drill	61.9	74.1	79.41	84.84	95.54
Duck	30.2	81.2	94.65	98.5	98.97
Eggbox	49.9	87.9	90.33	99.34	100
Glue	31.2	89	96.53	95.46	100
Hole punch	52.8	90.5	92.86	87.91	94.96
Iron	80.0	78.9	82.94	93.97	96.93
Lamp	67.0	74.4	76.87	92.32	96.93
Phone	38.1	77.3	86.07	92.99	98.46
Average	50.2	83.9	90.37	93.6	97.82

Figure 6. Illustration of pose estimation results using the LINEMOD data set.

5. Conclusion

In this study, we proposed a new 6D pose estimation algorithm which is based on RGB-D images. We used a CNN and a PointNet network to extract the color information and depth information of the object and fuse them at the pixel level. Then the region-level feature extraction network was proposed based on PointNet ++ and DGCNN to process the fused point clouds. After that, we obtained the global features and several partial features. Through integration of the global and local features, multiple fused features were obtained, which were used for estimating the object's pose. The result with the highest confidence was chosen and iteratively optimized to acquire the final result of pose estimation for the object. By means of the fusion approach we proposed, the network can fully utilize color, depth, and local and global information. Experiments using three data sets verified the effectiveness of the proposed model, and we also achieved better results than the state-of-the-art algorithms on benchmark data sets. The experimental results also showed that this method can effectively cope with the pose estimation problems in complex backgrounds and severely cluttered scenes. The fusion of color and depth information and the combination of local and global information has injected new vitality into the problem of pose estimation. In the future, we will further update our model to improve its accuracy and real-time performance for the task of pose estimation.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Funding

This research was funded by the Pudong New Area Science & Technology Development Fund (PKX2021-R07), the Natural Science Foundation of Shanghai (20ZR1440500), and Shanghai Normal University (SK202123).

Conflict of interest

The authors declare no conflicts of interest.

ORCID iDs

Xiangpeng Liu https://orcid.org/0000-0003-3540-848X
Huiping Duanmu https://orcid.org/0000-0001-5556-2164

References

[1] Wang H, Wang H and Zhuang C 2021 6D pose estimation from point cloud using an improved point pair features method *2021 7th Int. Conf. on Control, Automation and Robotics (ICCAR) (23–26 April 2021)* pp 280–4
[2] Lee S and Moon Y K 2022 Camera pose estimation using voxel-based features for autonomous vehicle localization tracking *2022 37th Int. Technical Conf. on Circuits/ Systems, Computers and Communications (ITC-CSCC) (5–8 July 2022)* pp 185–8
[3] Tang F, Wu Y, Hou X and Ling H 2019 3D mapping and 6D pose computation for real time augmented reality on cylindrical objects *IEEE Trans. Circuits Syst. Video Technol.* **30** 2887–99

[4] Park K, Patten T and Vincze M 2019 Pix2Pose: pixel-wise coordinate regression of objects for 6D pose estimation *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV) (27 October2 November 2019)* pp 7668–77

[5] Li Z, Wang G and Ji X 2019 CDPN: coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV) (27 October–2 November 2019)* pp 7678–87

[6] Cheng J, Liu P, Zhang Q, Ma H, Wang F and Zhang J 2021 Real-time and efficient 6-D pose estimation from a single RGB image *IEEE Trans. Instrum. Meas.* **70** 2515014

[7] Kehl W, Milletari F, Tombari F, Ilic S and Navab N 2016 Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation *2016 14th European Conf. on Computer Vision (ECCV) (11–14 October 2016)* pp 205–20

[8] Vacchetti L, Lepetit V and Fua P V 2004 Stable real-time 3D tracking using online and offline information *IEEE Trans. Pattern Anal. Mach. Intell.* **26** 1385–91

[9] Lowe D G 1999 Object recognition from local scale-invariant features *Proc. of 7th IEEE Int. Conf. on Computer Vision (20–27 September 1999)* vol 2 pp 1150–7

[10] Bay H, Tuytelaars T, Van G L, Leonardis A, Bischof H and Pinz A 2006 SURF :speeded up robust features *9th European Conf. on Computer Vision (ECCV) (7–13 May 2006)* vol 3951 pp 404–17

[11] Mur-Artal R, Montiel J M M and Tardos J D 2015 ORB-SLAM: a versatile and accurate monocular SLAM system *IEEE Trans. Robot.* **31** 1147–63

[12] Lepetit V, Moreno-Noguer F and Fua P 2009 EPnP: an accurate O(n) solution to the PnP problem *Int. J. Comput. Vis.* **81** 155–66

[13] Kendall A, Grimes M and Cipolla R 2015 PoseNet: a convolutional network for real-time 6-dof camera relocalization *Proc. of IEEE Int. Conf. on Computer Vision (ICCV) (13–16 December 2015)* pp 2938–46

[14] Xiang Y, Schmidt T, Narayanan V and Fox D 2017 PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes (arXiv:1711.00199)

[15] Michel F, Kirillov A, Brachmann E, Krull A, Gumhold S, Savchynskyy B and Rother C 2017 Global hypothesis generation for 6D object pose estimation *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (21–26 July 2017)* pp 462–71

[16] Wang C, Xu D, Zhu Y, Martin-Martin R, Lu C, Li F and Savarese S 2019 DenseFusion: 6D object pose estimation by iterative dense fusion *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (16–20 June 2019)* pp 3343–52

[17] Qi C R, Su H, Mo K and Guibas L J 2017 PointNet: deep learning on point sets for 3D classification and segmentation *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (21–26 July 2017)* pp 652–60

[18] Stevsic S, Christen S and Hilliges O 2020 Learning to assemble: estimating 6D poses for robotic object -object manipulation *IEEE Robot. Autom. Lett.* **5** 1159–66

[19] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X and Xiao J 2015 3D shapenets: a deep representation for volumetric shapes *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (7–12 June 2015)* pp 1912–20

[20] Yi L, Kim V G, Ceylan D, Shen I C, Yan M, Su H, Lu C, Huang Q, Sheffer A and Guibas L 2016 A scalable active framework for region annotation in 3D shape collections *ACM Trans. Graph.* **35** 1–12

[21] Armeni I, Sener O, Zamir A R, Jiang H, Brilakis I, Fischer M and Savarese S 2016 P3D semantic parsing of large-scale indoor spaces *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (26 June–1 July 2016)* pp 1534–43

[22] Tekin B, Sinha S N and Fua P 2018 Real-time seamless single shot 6D object pose prediction *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (18–22 June 2018)* pp 292–301

[23] Peng S, Zhou X, Liu Y, Lin H and Bao H 2019 PVNet: pixel-wise voting network for 6DoF object pose estimation *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (16–20 June 2019)* pp 4561–70

[24] Rad M and Lepetit V 2017 BB8: a scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)* pp 3828–36

[25] Song C, Song J and Huang Q 2020 HybridPose: 6D object pose estimation under hybrid representations *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 431–40

[26] Zakharov S, Shugurov I and Ilic S 2019 DPOD: 6D pose object detector and refiner *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)* pp 1941–50

[27] Hu Y, Fua P, Wang W and Salzmann M 2020 Single-stage 6D object pose estimation *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 2930–39

[28] Drost B, Ulrich M, Navab N and Ilic S 2010 Model globally, match locally: efficient and robust 3D object recognition *2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (13–18 June 2010)* (https://doi.org/10.1109/CVPR.2010.5540108)

[29] Vidal J, Lin C Y and Marti R 2018 6D pose estimation using an improved method based on point pair features *2018 4th Int. Conf. on Control, Automation and Robotics (ICCAR) (20–23 April 2018)* pp 405–9

[30] Wohlhart P and Lepetit V 2015 Learning descriptors for object recognition and 3D pose estimation *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (7–12 June 2015)* pp 3109–18

[31] Tejani A, Tang D, Kouskouridas R and Kim T K 2014 Latent-class hough forests for 3D object detection and pose estimation *13th European Conf. on Computer Vision (ECCV) (6–12 September 2014)* pp 462–77

[32] Liu Y, Zhou L, Zong H, Gong X , Wu Q, Liang Q and Wang J 2019 Regression-based three-dimensional pose estimation for texture-less objects *IEEE Trans. Multimedia* **21** 2776–89

[33] Li C, Bai J and Hager G D 2018 A unified framework for multi-view multi-class object pose estimation *Proc. of European Conf. on Computer Vision (ECCV)* pp 254–69

[34] He Y, Sun W, Huang H, Liu J, Fan H and Sun J 2020 PVN3D: a deep point-wise 3D keypoints voting network for 6DoF pose estimation *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (14–19 June 2020)* pp 11632–41

[35] He Y, Huang H, Fan H, Chen Q and Sun J 2021 FFB6D: a full flow bidirectional fusion network for 6D pose estimation *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (19–25 June 2021)* pp 3003–13

[36] Zhou G, Wang H, Chen J and Huang D 2021 PR-GCN: a deep graph convolutional network with point refinement for 6D pose estimation *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV) (11–17 October 2021)* pp 2793–802

[37] Jiang X, Li D, Chen H, Zheng Y, Zhao R and Wu L 2022 Uni6D: a unified CNN framework without projection breakdown for 6D pose estimation *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR) (19–24 June 2022)* pp 11174–84

[38] Zhang Y, Liu Y, Wu Q, Zhou J, Gong X and Wang J 2022 EANet: edge-attention 6D pose estimation network for

texture-less objects *IEEE Trans. Instrum. Meas.* **71** 1–13
[39] Qi C R, Li Y, Hao S and Guibas L J 2017 PointNet++: deep hierarchical feature learning on point sets in a metric space *Advances in Neural Information Processing Systems 30 (NIPS 2017)* ed I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan and R Garnett
[40] Wang Y, Sun Y, Liu Z, Sarma S E, Bronstein M M and Solomon J M 2019 Dynamic graph CNN for learning on point clouds *ACM Trans. Graph.* **38** 1–12
[41] Besl P J and Mckay N D 1992 Method for registration of 3D shapes *Sensors Fusion IV: Control Paradigms and Data Structures* vol 1611 pp 586–606
[42] Hinterstoisser S, Lepetit V, Ilic S, Holzer S, Bradski G, Konolige K and Navab N 2012 Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes *11th Asian Conf. on Computer Vision (ACCV) (5–9 November 2012)* pp 548–62
[43] Chang A X, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H *et al* 2015 ShapeNet: an information-rich 3D model repository (arXiv:1512.03012)
[44] Sundermeyer M, Marton Z C, Durner M, Brucker M and Triebel R 2018 Implicit 3D orientation learning for 6D object detection from RGB images *Proc. of European Conf. on Computer Vision (ECCV) (8–14 September 2018)* pp 699–715
[45] Brachmann E, Michel F, Krull A, Yang M Y, Gumhold S and Rother C 2016 Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (26 June–1 July 2016)* pp 3364–72