End to End Robust Point-Cloud Alignment Using Unsupervised Deep Learning

Xuzhan Chen[1,2], Youping Chen[1], and Homayoun Najjaran[2(✉)]

[1] School of Mechanical Science and Engineering,
Huazhong University of Science and Technology, Wuhan, China
[2] School of Engineering, The University of British Columbia, Vancouver, Canada
homayoun.najjaran@ubc.ca

Abstract. The point-cloud alignment methods help robots to map their environment, recognize target objects and estimate rigid-body object poses from the 3D vision sensor data. In this paper, we propose a robust and computationally efficient approach for point-cloud alignment. Unlike the feature descriptor-based pose classifiers or regression methods, the proposed method can process an unordered point cloud by mapping it uniquely onto a particular 2D space determined based on the point cloud from the object. The model training is fully unsupervised and relies on optimizing the projection results based on a loss function. Specifically, the proposed 2D mapping enables the model to recognize objects with a simple linear classifier to increase computational efficiency. Then, the proposed method calculates the object pose in the continuous space rather than classifying the point cloud into discrete pose labels. The experiments and comparison with a well-established descriptor-based point-cloud alignment method show that the proposed method has a good performance and is robust to missing points of the point cloud. The higher performance in recognition and pose estimation precision make the method suitable for industrial robotic and automation applications.

Keywords: Unsupervised learning · Point cloud alignment · Object recognition · Object pose estimation

1 Introduction

Aligning point clouds collected by 3D sensors such as scanning LiDARs and RGB-D cameras to the standard models has potential for frontier robot applications such as object grasping, 3D scene registration, and robot navigation. To successfully align point clouds, the algorithm needs to i) recognize the object from multiple potential candidates, and ii) estimate the rigid body pose from the input point cloud. However, point-cloud alignment is still an open research topic since with a large number of candidate objects there will be a large number of similar and confusing features so that the algorithm needs to be robust to noise and missing points.

In this paper, we focus on the point cloud alignment topic. Typically, the shapes of all candidates are known. Hence, it is reasonable to assume that the CAD models or 3D

T. McDaniel et al. (Eds.): ICSM 2019, LNCS 12015, pp. 158–168, 2020.
https://doi.org/10.1007/978-3-030-54407-2_14

scans of the target objects are given. Our goal is then to recognize the object category and estimate the object pose based on a point cloud, simultaneously.

The intuitive methods to align point cloud builds on the shape descriptors which encode local geometry into a feature vector. Then, the corresponding points are paired based on the feature vector similarity, and the relative 6 degree-of-freedom pose is solved with respect to the rigid body translation determined by point pair matching. One of the problems is that point pairing process requires high-dimension searching, and the search time grows fast with the increasing number of candidate object features. Thus, the point pairing-based method is not well-suited for big data applications. Another problem is that the descriptor- and matching-based point cloud alignment methods highly depend on the quality of the point cloud and repeatable local features. Thus, poor quality point clouds can compromise the performance and leads to incorrect object recognition results. Also, local similarities between different objects can cause difficulties for point-cloud alignment.

Inspired by human recognition i.e., manipulating an object until the most obvious perspective is achieved, this paper introduces Deep Point Cloud Mapping Network (DPC-MN) that is designed as an end-to-end solution to obtain an optimal unique representation for the object point cloud regardless of its pose. Then, the points can be automatically paired based on their unique representation shown in Fig. 1. The DPC-MN point-cloud alignment offer two advantages i) the intra-class differences caused by various poses are omitted so recognition of the object category can be more robust, and ii) pairing is accelerated since the high-dimension searching process is removed.

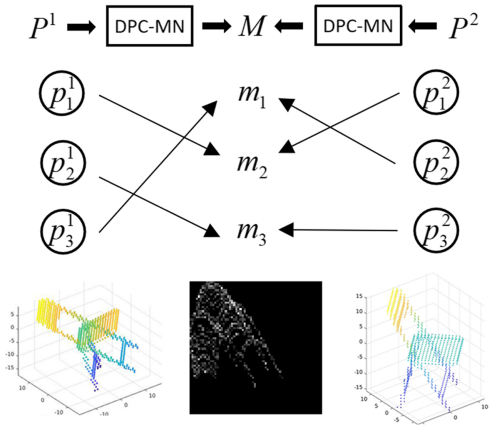


Fig. 1. The DPC-MN model pairs points of the point clouds of an object taken from different poses. P1 and P2 are the point clouds, M is a unique representation of the object, and p is a point in the point cloud.

One of the novelties of the DPC-MN model is that the point-cloud alignment is end-to-end processed by the deep learning technique and fundamentally different from the feature descriptor cascade. Aligning point clouds via the deep learning-based method

is also more robust than the descriptor cascade since the feature extraction ability is learned from data instead of using a prior knowledge.

Another novelty is that the DPC-MN model can be trained unsupervised, i.e. the pose labels are not required for pose estimation. Almost similar to reinforcement learning, the proposed method learns a proper action i.e., mapping the point cloud into a unique 2D view. Then, the degree of self-occlusion of the mapping is used as the optimization objective that enables the convergence of the model training. In the end, the performance of the proposed method is also boosted by the use of machine learning and GPU accelerating technique.

The contribution of this paper can be summarized as,

1). The DPC-MN model is proposed. In this way, the point-cloud alignment is end-to-end processed by the deep learning technique and fundamentally different from the feature descriptor cascade.
2). The DPC-MN model can be trained unsupervised, i.e. the pose labels are not required for pose estimation.

The rest of the paper is organized as follows. Section 2 reviews the previous work related to pose estimation based on the 3D data. Section 3 shows the analysis of the deep point cloud mapping network model. In Sect. 4, the proposed model is verified using a 3D shape dataset available online. Section 5 summarizes the concluding remarks of this work.

2 Literature Review

2.1 Descriptor-Based Recognition and Pose Estimation

Descriptors provide a means to quantify the local and global information of the point cloud. Rusu et al. [1] proposed the VFH global shape descriptor which is based on the histogram of the object normals and invariant to object rotation. Aldoma et al. [2] introduced the CVFH feature descriptor which solved the mass center shifting problem by pre-clustering the object point cloud. Most global feature descriptors such as VFH and CVFH are invariant to the object rotation. The rotation invariant property improves recognition accuracy but also blurs the features compromising the object pose estimation. The CVFH is extended to pose estimation by adopting an additional camera roll histogram [2].

In contrast to the global feature descriptors, local feature descriptors emphasize the local geometry of the object surface. Li et al. [3] used a cascade of 3D key point detection, key point description, and key point matching. Their method can align CAD models in the dataset with the point cloud. The cascade is known as the standard framework for pose estimation with local features [4, 5].

One of the most powerful local feature descriptors for pose estimation is the point pair feature (ppf) [6]. Drost et al. [5] proposed the ppf constructed by repeatedly sampling two points from the point cloud and calculating four elements of the feature vector. Each pair of a ppf can estimate a rigid body transformation from the source and the target point cloud, and a voting strategy can be used to find the most likely pose among the calculated

poses. Choi et al. [7, 8] extended the idea of ppf to the point-point pair, the point-surface pair, the surface-surface pair, and the color point pair to improve the results. In addition to designing a better feature, Hinterstoisser et al. [9] improved the voting scheme of pose estimation by introducing smart sampling. The ppf methods are dominant among the local feature-based pose estimation methods. However, the computational load of sampling and voting procedures grows fast as the number of object points increases.

2.2 Processing the Raw Point Cloud with Deep Learning

The proposed method offers two key features: i) the model uses raw point clouds, instead of latticing the shape, and ii) it can be trained via unsupervised learning. Neural networks capable of processing raw point clouds have recently drawn a lot of attention. Our previous work [10] proposed a point convolution network which recognizes objects from point cloud via the defined point convolution operation. Qi et al. [11] proposed the PointNet which yields high performance on both the object recognition and segmentation domains. Wang et al. [12] proposed the O-CNN model that leverages on the Oct-tree data structure of the point cloud. Klokov et al. [13] came up with the kd-tree based raw-point network.

However, all of these works [10–13] are based on supervised learning which requires a large labeled dataset and aims to generalize the object recognition to unseen objects within known categories. The proposed method in this paper is based on unsupervised learning which doesn't require labeled dataset to train the model. Furthermore, the proposed method focuses on object pose estimation using neural networks which have received far less coverage in the published literature. Descriptors provide a means to quantify the local and global information of the point cloud. Rusu et al. [1] proposed the VFH global shape descriptor which is based on the histogram of the object normals and invariant to object rotation. In contrast to the global feature descriptors, local feature descriptors emphasize the local geometry of the object surface. Li et al. [3] used a cascade of 3D key point detection, key point description, and key point matching. Their method can align CAD models in the dataset with the point cloud. The cascade is known as the standard framework for pose estimation with local features [4, 5].

One of the most powerful local feature descriptors for pose estimation is the point pair feature (ppf) [6]. Drost et al. [5] proposed the ppf constructed by repeatedly sampling two points from the point cloud and calculating four elements of the feature vector. Each pair of a ppf can estimate a rigid body transformation from the source and the target point cloud, and a voting strategy can be used to find the most likely pose among the calculated poses. Choi et al. [7, 8] extended the idea of ppf to the point-point pair, the point-surface pair, the surface-surface pair, and the color point pair to improve the results.

2.3 Point Cloud Recognition Using Deep Learning

The proposed method offers two key features: i) the model uses raw point clouds, instead of latticing the shape, and ii) it can be trained via unsupervised learning. Neural networks capable of processing raw point clouds have recently drawn a lot of attention. Our previous work [10] proposed a point convolution network which recognizes objects from point cloud via the defined point convolution operation. Qi et al. [11] proposed the

PointNet which yields high performance on both the object recognition and segmentation domains. Wang et al. [12] proposed the O-CNN model that leverages on the Oct-tree data structure of the point cloud. Klokov et al. [13] came up with the kd-tree based raw-point network. However, all of these works [10–13] are based on supervised learning which requires a large labeled dataset and aims to generalize the object recognition to unseen objects within known categories.

3 Method

3.1 Deep Point Cloud Mapping Network Architecture (DPC-MN)

The mapping function g is formulated by the neural network. Given a point cloud P, the mapping function will be invariant to the permutation of P. Inspired by the ideas of NiN [18] and PointNet [11], we adopted a 1×1 convolution kernel to process and extract the features of the point cloud. However, the extracted features are used to generate the mapping matrix instead of classifying the object. The architecture of the DPC-MN is shown in Fig. 2. The network aims to learn an appropriate projection matrix based on the features of the whole point cloud. The average pooling is used to generate the global feature of the point cloud because it is a symmetric operation and invariant to the set order of the point cloud.

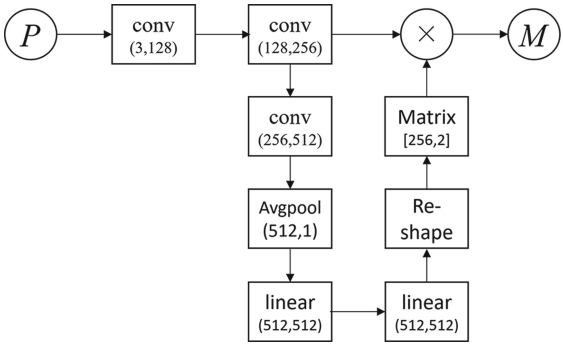


Fig. 2. The architecture of DPC-MN. The 1×1 convolution is noted by conv. (m, n) means the input feature dimension is m and the output feature dimension is n. The linear means the output feature is calculated by a fully connected layer. All layers are activated by the ReLU non-linear function. The Reshape block reshapes the 512-dimension feature into the 256×2 matrix and the mark $\times$ means to multiply the 256 dimensions "point cloud" with the 256×2 matrix. All of the layers are activated by the ReLU non-linear function.

3.2 Loss Function

The goal of the loss function is to map point cloud in different poses into the same 2D representation. In this way, the recognition of point cloud is simplified since the variation caused by the poses is removed. Three principles are designed to obtain the idealized

properties of the mapping function: 1) neighborhood points in the point cloud are mapped into a tight area in 2D space, 2) the overlap of mapped shape must be minimum. 3) point clouds in different poses are mapped into the same 2D representation.

The anchor point cloud P_{a} is the original data collected from the 3D sensor or rendered from the CAD model. Based on the first principle, the positive point cloud P_{p} is generated by shifting each point in the anchor point cloud P_{a} within a small random δ. The negative point cloud P_{n} is generated by shuffling the anchor point cloud P_{a} so the operation is equivalent to randomly select 2 points for numbers of times. Based on the first and second principles, the function L_{m} is shown in (1).

$$L_m = \frac{1}{N}\sum \max\left(\left[\varepsilon + \left\|M_a - M_p\right\|_2 - \|M_a - M_n\|_2\right], 0\right) \tag{1}$$

where ε is a margin value, and M_{a}, M_{p}, M_{n} are the output generated by applying the proposed DPC-MN to P_{a}, P_{p}, and P_{n}, respectively.

Based on the third principle, the L_{p} is defined as,

$$L_p = \frac{1}{N}\sum \|M_a - M_i\|_2 \tag{2}$$

The loss function f is,

$$f = L_m(M_a, M_p, M_n) + \lambda \cdot L_p(M_a, M_i) \tag{3}$$

where λ is a hyperparameter that adjusts the weight of Lm and Lp.

3.3 Object Recognition and Pose Estimation

First, the model output M is regularized into a 64 × 64 2D grid based on the (u, v) values in M. The (u, v) value is rounded into closest integer number and taken as 2D coordinates of the bin in the grid. Then, the value of the bin will accumulate *one unit* for every (u, v) assigning to the bin. Based on the steps, the M can be converted into a 2D grid-based representation. The recognition runs on the 2D grid representation with a linear classifier. The classifier is defined by,

$$y_i = \frac{e^{\sum_{j=1}^{j=k} w_{ij}x_j}}{\sum_i e^{\sum_{j=1}^{j=k} w_{ij}x_j}} \tag{4}$$

where j is the index for the pixel in the 2D grid, i indicates the category, x is the 2D grid representation, and w_{ij} is trainable parameters for the recognition. y_i is the output score for the category i. The linear classifier can be easily trained because of the simple one-layer linear structure.

The pose of the object is calculated by finding the corresponding points in two point clouds. Because the point cloud is an $N \times 3$ matrix, we use left matrix multiplication instead of the standard form. The optimized solution of R is given in (11). The Single

Value Decomposition (SVD) method is used to ensure that R is an orthogonal matrix. The solution is based on the least squares method.

$$\hat{R} = \arg\min(\left\| \hat{S}_{pi} - S_{pi} \right\|_2) = (S_{pj}^T \cdot S_{pj})^{-1} \cdot S_{pj}^T \cdot S_{pi} \tag{5}$$

$$\tilde{R} = U \cdot V^T, where(U, S, V^T) = svd(\hat{R}) \tag{6}$$

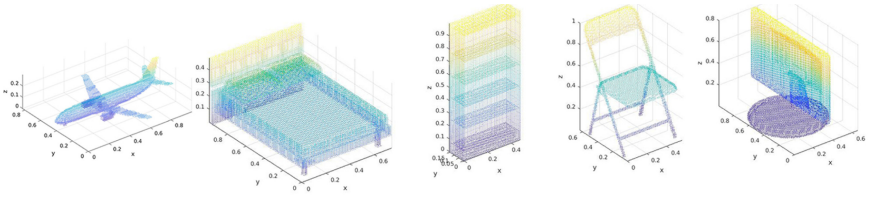
4 Experiment Results

A. Experiment Configuration

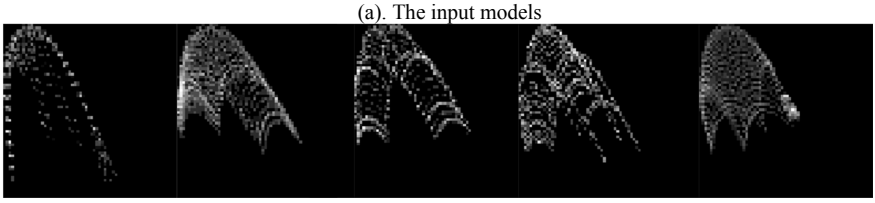
The whole model ran on a workstation with an Nvidia GTX1070 GPU, E5 CPU, and 27 GB memory. The training and testing CAD models were taken from the ModelNet40 dataset [19]. Firstly, the point cloud is augmented by rotating about *x, y, and z* axes. The rotation angle is uniformly generated from 0 to $\pi/2$. For training of the network, the angle is incremented by $\pi/24$. For testing of the network, the angle is incremented by $\pi/10$. Thus, the training and testing datasets have no intersections.

The model is optimized by the SGDM solver. The learning rate is 0.001 and the momentum is 0.90. The training batch size is 16 and the maximum training epoch is 300 based on our configuration. The loss weight λ is 0.3. The training time for each epoch was about 380 s, and the model took around 40 epochs to converge.

The input models and the mapped result M for each input model are visualized in Fig. 3(a) and Fig. 3(b), respectively. The view is randomly selected from the test set because the mapped views for objects in different poses are completely identical.



(a). The input models



(b). The mapped results

Fig. 3. The input models and mapped results

4.1 Recognizing Objects from Different Poses

The classifier is trained on the workstation with a GPU accelerator. Because all of the poses are mapped into a unique view, the model took only 2 epochs to converge and the linear classifier took 0.1 s to finish one epoch. Thus, the training time for classification was negligible.

The point cloud sparsity is a common problem for 3D sensors. The point cloud sparsity may be caused by the low resolution of the sensor or the inappropriate measuring distance. The recognition experiment is designed to simulate the sparsity of 3D sensors and verifies the robustness of the recognition algorithm. To simulate the sparsity, 10% to 90% points were randomly selected from the original complete point cloud. Figure 4 shows the downsampling point cloud for a single pose, but downsampling has been applied to the entire test dataset.

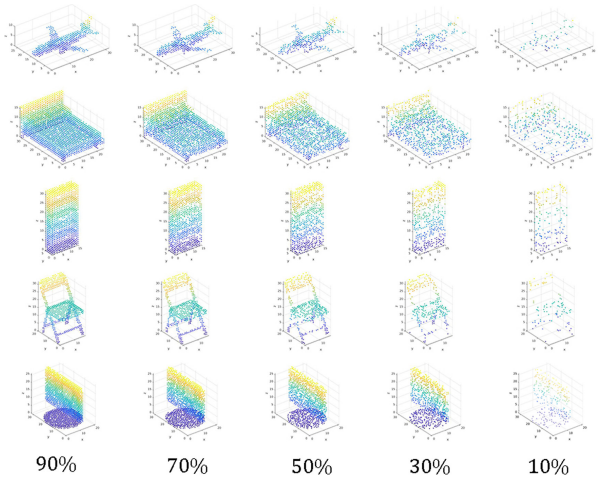


Fig. 4. Point cloud downsampling with different sampling rates

We used the linear classifier that trained on the complete point cloud to process the incomplete data. The classifier is not trained with augmented incomplete data. The proposed method is compared with the Point Pair Feature (PPF)-based point cloud alignment method in Fig. 5. The result shows that the proposed method is remarkably robust to missing points. Even with only 10% of the points, the method can still recognize objects from different poses with acceptable accuracy (more than 80%).

4.2 Estimating Object Poses

The object pose is calculated based on the cascades described in Sect. 3E. For each testing instance, the mapped result M is saved with the input point cloud P as a pair. The input point cloud P for each instance is shuffled to simulate the unordered data collected by 3D sensors.

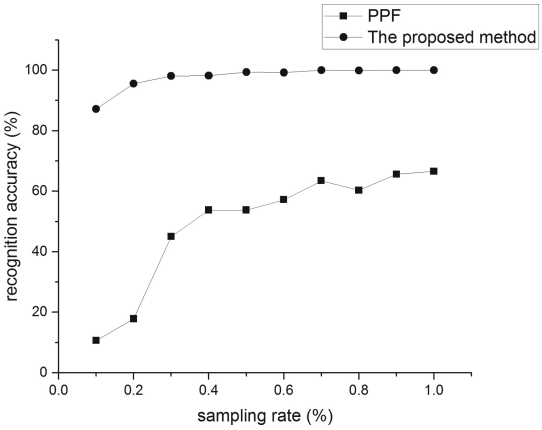


Fig. 5. The recognition accuracy under different downsampling rates

For each test object, the pose is calculated based on the 2D views of the standard pose and the test object. The standard pose refers to the reference (zero) angles, and then the test object can rotate about any arbitrary axes and at any angles. The rotation matrix R which can transform the test object to the standard pose is calculated based on (12).

The accuracy of 3D object poses is quantified by the error of the point cloud alignment. The test object rotates to the standard pose with the calculated rotation matrix R. Then, the nearest neighborhood searching tree is built to match the nearest points between the test object and the standard pose point cloud as the ICP algorithm does. The mean distance is taken as the quantified error of pose estimation. Figure 6 shows the errors of pose estimation in different poses. For each test object, 125 poses are tested as described in Sect. 3. The average error is around 0.1 unit of the object scale which is only 10% compared to the object scale. Alignment error is 1.0 unit means that the object is not successfully recognized. The proposed method is compared with the Point Pair Feature (PPF)-based point cloud alignment method in Fig. 6. The result shows that the proposed method has a better pose estimation precision and is more robust compared to the PPF-based method. Fine tuning algorithms such as ICP can reduce the error with further processing.

A real-world object alignment experiment is conducted to validate that the proposed method can be used in the sensor scanning data. A drill is scanned using a Kinect sensor. The drill model is 3D reconstructed and represented by the point cloud. Then 4 different rigid body poses of the drill is used to validate the method and the drill is scanned by the kinect sensor. The proposed method can automatically align the reconstructed 3D model to the real-world scanning data, shown in Fig. 7.

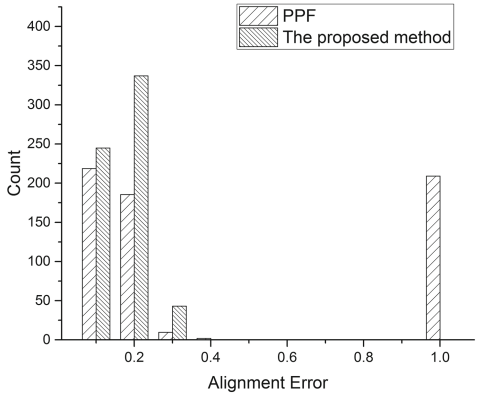


Fig. 6. The error under different poses and categories. The proposed method is compared with the Point Pair Feature (PPF)-based point cloud alignment method in Fig. 6. The result shows that the proposed method has a better pose estimation precision and recognition accuracy. Fine tuning algorithms such as ICP can reduce the error with further processing.

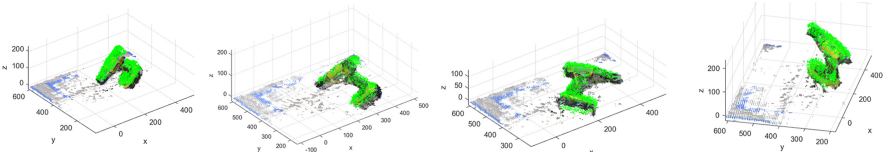


Fig. 7. The real-world experiment on point cloud alignment.

5 Conclusion

In this paper, we proposed a novel idea and an effective method for aligning point clouds and help robot to simultaneously recognize objects and estimate the pose of rigid-body objects. An object in different poses can be mapped into a unique 2D view from its 3D point cloud representation. Based on this idea, the 2D view can dramatically facilitate object recognition and pose estimation performance in terms of efficiency and accuracy. Deep Point Cloud Alignment Network (DPCAN) method is proposed to implement the unique 2D view mapping function. The network can be trained unsupervised by both CAD models and real point clouds of target without the need to labeling the training datasets. The proposed network is verified to be robust against missing points of the test data. Experiments showed that the model has acceptable accuracy and robustly recognize more than 80% of the objects even when only 10% of the points of the point clouds were used. Based on the proposed method, the pose can be calculated continuously instead of estimating the pose at discrete pose labels. The accuracy of the pose estimation is about 10% of the object scale which means the proposed DPCAN is sufficient for many industrial robotic and automation applications

References

1. Rusu, R.B., et al.: Fast 3D recognition and pose using the viewpoint feature histogram. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2010)
2. Aldoma, A., et al.: CAD-model recognition and 6DOF pose estimation using 3D cues. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE 2011
3. Li, Y., et al.: Database-assisted object retrieval for real-time 3D reconstruction. In: Computer Graphics Forum. Wiley Online Library (2015)
4. Malleus, L., et al.: KPPF: keypoint-based point-pair-feature for scalable automatic global registration of large RGB-D scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
5. Drost, B., et al.: Model globally, match locally: efficient and robust 3D object recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2010)
6. Kiforenko, L., et al.: A performance evaluation of point pair features. Comput. Vis. Image Underst. **166**, 66–80 (2018)
7. Choi, C., et al.: Voting-based pose estimation for robotic assembly using a 3D sensor. In: 2012 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2012)
8. Choi, C., Christensen, H.I.: RGB-D object pose estimation in unstructured environments. Rob. Auton. Syst. **75**, 595–613 (2016)
9. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K.: Going further with point pair features. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 834–848. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_51
10. Chen, X., Chen, Y., Najjaran, H.: 3D object classification with point convolution network. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2017)
11. Qi, C.R., et al.: Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), vol. 1, no. 2, p. 4. IEEE (2017)
12. Wang, P., et al.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. (TOG) **36**(4), 72 (2017)
13. Klokov, R., Lempitsky, V.: Escape from cells: deep kd-networks for the recognition of 3D point cloud models. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE (2017)