

31st CIRP Conference on Life Cycle Engineering (LCE 2024)

6D Pose Estimation on Point Cloud Data through Prior Knowledge Integration: A Case Study in Autonomous Disassembly

Chengzhi Wu ^{*†a}, Hao Fu ^{†a}, Jan-Philipp Kaiser^b, Erik Tabuchi Barczak^c,
Julius Pfrommer^d, Gisela Lanza^b, Michael Heizmann^c, Jürgen Beyerer^{a,d}

^aInstitute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

^bwbk Institute of Production Science, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany

^cInstitute of Industrial Information Technology, Karlsruhe Institute of Technology, Hertzstraße 16, 76187 Karlsruhe, Germany

^dFraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Fraunhoferstraße 1, 76131 Karlsruhe, Germany

* Corresponding author. Tel.: +49-(0)1523 8476995. E-mail address: chengzhi.wu@kit.edu

† Equal Contribution

Abstract

The accurate estimation of 6D pose remains a challenging task within the computer vision domain, even when utilizing 3D point cloud data. Conversely, in the manufacturing domain, instances arise where leveraging prior knowledge can yield advancements in this endeavor. This study focuses on the disassembly of starter motors to augment the engineering of product life cycles. A pivotal objective in this context involves the identification and 6D pose estimation of bolts affixed to the motors, facilitating automated disassembly within the manufacturing workflow. Complicating matters, the presence of occlusions and the limitations of single-view data acquisition, notably when motors are placed in a clamping system, obscure certain portions and render some bolts imperceptible. Consequently, the development of a comprehensive pipeline capable of acquiring complete bolt information is imperative to avoid oversight in bolt detection. In this paper, employing the task of bolt detection within the scope of our project as a pertinent use case, we introduce a meticulously devised pipeline. This multi-stage pipeline effectively captures the 6D information with regard to all bolts on the motor, thereby showcasing the effective utilization of prior knowledge in handling this challenging task. The proposed methodology not only contributes to the field of 6D pose estimation but also underscores the viability of integrating domain-specific insights to tackle complex problems in manufacturing and automation.

© 2024 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 31st CIRP Conference on Life Cycle Engineering (LCE 2024)

Keywords: Remanufacturing; 6D pose estimation; Prior knowledge utilization; 3D point cloud; Machine learning

1. Introduction

In today's context, within the framework of life cycle engineering (LCE), the importance of End-of-Life (EOL) strategies has been heightened due to growing environmental concerns. As one of the EOL strategies, remanufacturing salvages the efforts that were initially used to shape the product and presents opportunities for creating further business profits [1]. In remanufacturing, the used products are reintegrated into the production line as a practical application of LCE. After disassembly and additional processing, the individual parts are reassembled into remanufactured products, which should possess functionality and quality equivalent to that of a new product [2]. In contrast to traditional manufacturing, remanufacturing is chal-

lenging due to the complexity, uncertainty, and inconsistency of the returned products [3]. Consequently, the primary processes in the current remanufacturing are still carried out manually, which limits its widespread adoption [4]. In order to automate this process, a flexible and adaptive system is required.

The AgiProbot project [5, 6, 7] focuses on such automated disassembly system of end-of-life electric motors. As shown in the leftmost part of Fig. 1, the processed starter motor is clamped and disassembled by a multifunctional robotic arm. Concerning the bolts on the motor, besides part segmentation, accurate 6D pose, which encompasses their 3D position and 3D orientation, is essential to facilitate the robotic arm's spatial movement to the correct position and ensure its disassembly movement in the correct direction. However, due to cost and space constraints, the 3D camera is usually installed in a fixed position, which can provide only single-view point clouds. Meanwhile, under the rigidity and stability requirements of the

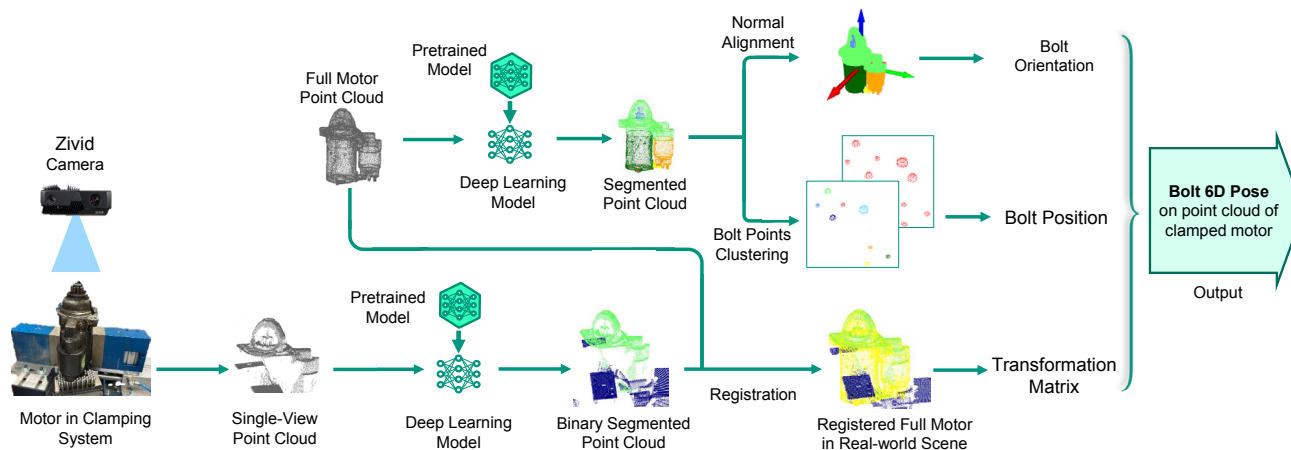


Fig. 1. Full pipeline of our proposed method. A starter motor is clamped in the clamping system and scanned by the Zivid camera. Extra input data of the full motor point cloud provides the prior knowledge of the motor model. After registration, the 6D pose information of bolts under real-world scenes can be directly given.

clamping system, the motor remains in a random but immovable position during the disassembly process. This results in low-quality scanning or even view obstruction of numerous critical features, such as bolts on the complex motor. In this paper, we propose an ingenious method that uses extra prior knowledge to compute precise bolt 6D pose, even under these unfavorable conditions.

Driven by the demands and product iterations, the starter motors exhibit a significant degree of diversity [8]. However, greater variations introduce higher risks and costs. Therefore, there are always specific reference systems that remain consistent [9], which can be utilized as prior knowledge. Our prior knowledge indicates that the bolts are located only in specific regions, oriented along the motor's axis. We can simplify the initially challenging task of 6D pose estimation for small objects that lack distinct features by focusing on the prominent features of the motor. Additionally, estimating high-precision 6D pose solely from a single model is still challenging. Therefore, we attempted to decompose it into multiple, more precise steps using more comprehensive information sources.

The remainder of this paper is structured as follows: Section 2 summarizes related works. Section 3 describes the technical details of our bolt 6D pose estimation pipeline. Section 4 provides experimental results, while section 5 gives a conclusion.

2. Related Work

6D Pose Estimation with RGB(D) Data. Initially, methods based on single RGB images were applied. These methods extract distinctive invariant local features from the images and subsequently match them with their corresponding features in 3D space [10, 11, 12]. Based on the 2D-3D correspondences, algorithms, such as PnP [13] and ICP [14], are applied to estimate the 6D pose. However, the application of RGB-based methods is often an issue when dealing with occlusions or objects without significant texture and discriminative color.

Therefore, additional depth information has been introduced. MV3D [15], Frustum PointNets [16] and AVOD [17]

apply convolutional neural networks (CNN) to extract features from RGB images, LIDAR Bird view and LIDAR Front view. Subsequent to information fusion, another CNN outputs the positions of bounding boxes, with the assumption that they are all situated on a common plane. PointFusion [18] and DenseFusion [19] employ PointNet [20] as the feature extractor for point cloud data and CNN as the feature extractor for RGB images. The information from both sources is combined together by a Dense Fusion module. This method evolves into MV6D [21] by employing multi-view point clouds and introducing the detection of 3D center points in addition to key points.

6D Pose Estimation with Point Cloud Data. With the rapid technological advancements in LiDAR sensors, which output data in the form of point clouds, a multitude of point cloud based pose estimation methods have been developed [22, 23]. Early research aimed to expand classical convolutional neural networks into voxelized 3D space to form an end-to-end model [24]. However, 3D convolution is prohibitively computationally expensive for practical applications. Therefore, more recent approaches have employed various other algorithms, such as PointNet [20], PointNet++ [25], and DGCNN [26], to encode geometric features. A deep Hough voting network was developed for end-to-end 3D object detection [27]. The success of Transformer [28] in 2D images has inspired transformer-based algorithms. Building upon Point Cloud Transformer (PCT) [29], the Multiscale Point Cloud Transformer (MSPCT) [30] was introduced to address point cloud data.

3. Methodology

3.1. Full Pipeline

We employ three distinct coordinate systems in our study. The first is the robot coordinate system, referred as Coordinate System \mathcal{A} , which has its origin at the center of the robot arm's base and serves as the final reference for all the results in the actual remanufacturing process. The second is the Zivid camera coordinate system, denoted as Coordinate System \mathcal{B} , which is

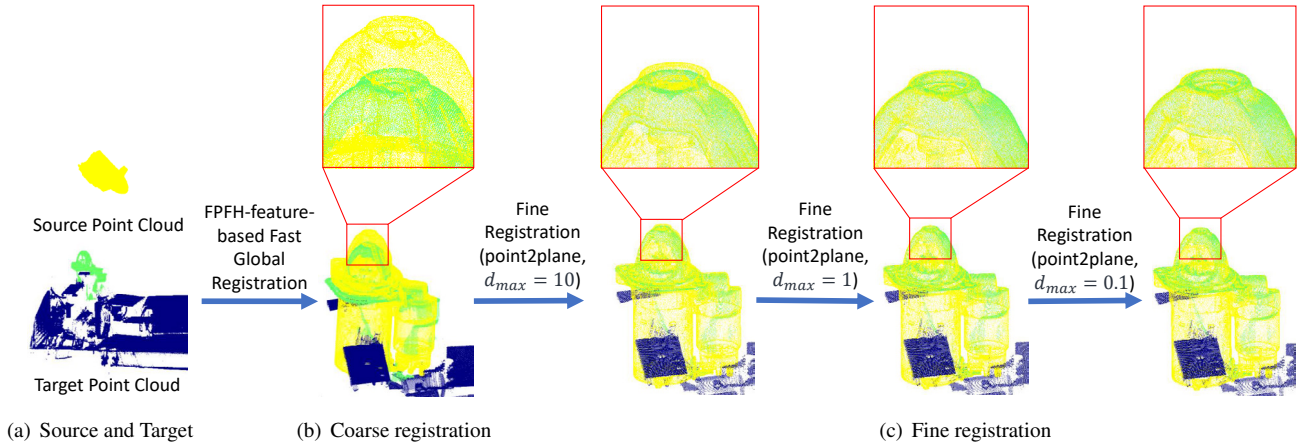


Fig. 2. Registration between the source point cloud (full motor) and the target point cloud (real scene). A coarse registration is first performed, followed by three successive fine registration processes.

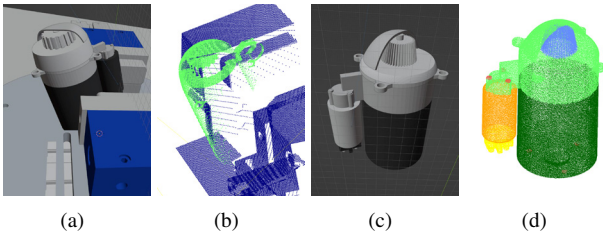


Fig. 3. Synthetic datasets created with Blender. (a, b) Synthetic scene and automatically labeled point cloud for binary segmentation. (c, d) Synthetic motor and automatically labeled point cloud for part segmentation.

used for the scanned single-view point clouds. The third one is the coordinate system of the extra input of full motor point clouds, denoted as Coordinate System C . The objective of this multi-stage pipeline is to estimate the 6D pose of motor bolts in the robot's coordinate system, orientation ${}^{\mathcal{A}}R_{bolt} \in SO(3)$ and position ${}^{\mathcal{A}}\mathbf{t}_{bolt} \in \mathbb{R}^3$, with the additional help of certain extra input data and prior knowledge of the starter motor. As illustrated in Fig.1, our full pipeline consists of five stages: (i) binary segmentation on real-world scene point cloud; (ii) part segmentation on the extra input of full motor point cloud; (iii) registration between two point clouds; (iv) bolt point clustering for bolt position; and (v) normal alignment for bolt orientation.

The full model point cloud of the starter motor, which can be generated from its CAD model or obtained through a scanner, is acquired as prior knowledge in this study. The extra input full model point cloud is first fed into a deep learning model that segments it into 6 semantic parts. The orientation of bolt ${}^C R_{bolt}$ is determined based on the orientation of the full motor ${}^C R_{motor}$. Prior knowledge suggests that the orientation of the bolt ${}^C R_{bolt}$ can either be the same as or opposite to ${}^C R_{motor}$.

$${}^C R_{bolt} = \begin{cases} {}^C R_{motor} & , \text{bolt on the top} \\ -{}^C R_{motor} & , \text{bolt on the bottom} \end{cases} \quad (1)$$

However, the full motor point clouds are typically from different sources or scanned in different positions, which means ${}^C R_{motor}$ is unknown. Therefore, a segmentation result-based normal alignment method is proposed, allowing us to align the

motors to obtain their respective ${}^C R_{motor}$ and determine ${}^C R_{bolt}$ accordingly. The 3D positions of the bolts ${}^C \mathbf{t}_{bolt}$ within coordinate system C is represented by the clustered centers of individual bolt point groups in the segmented full motor point cloud.

During remanufacturing, a single-view point cloud is captured by a Zivid camera. The single-view point cloud is passed through a deep learning model, which distinguishes between the clamping system and the motor. Using the point cloud of this motor as the target and the full motor point cloud as the source, a registration process is executed, resulting in a transformation matrix from the extra input coordinate system to the Zivid coordinate system $T_{C \rightarrow \mathcal{B}}$. As a result, the 6D information of the bolts in the Zivid coordinate system and the robot's coordinate system can be obtained from following equations:

$$\begin{bmatrix} {}^{\mathcal{B}}R_{bolt} & {}^{\mathcal{B}}\mathbf{t}_{bolt} \\ 0 & 1 \end{bmatrix} = T_{C \rightarrow \mathcal{B}} \begin{bmatrix} {}^C R_{bolt} & {}^C \mathbf{t}_{bolt} \\ 0 & 1 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} {}^{\mathcal{A}}R_{bolt} & {}^{\mathcal{A}}\mathbf{t}_{bolt} \\ 0 & 1 \end{bmatrix} = T_{\mathcal{B} \rightarrow \mathcal{A}} \begin{bmatrix} {}^{\mathcal{B}}R_{bolt} & {}^{\mathcal{B}}\mathbf{t}_{bolt} \\ 0 & 1 \end{bmatrix} \quad (3)$$

3.2. Scene Binary Segmentation and Motor Part Segmentation

For this study, we have access to only 18 starter motors from a series. To obtain a full model data set, we utilize a T-scanner to scan the motors. Subsequently, the motors are secured in a clamping system in accordance with the remanufacturing process requirement. They are scanned by a fixed Zivid camera. And a real-world single-view dataset is obtained. However, these data sets proved insufficient in terms of size and representativeness for our two segmentation tasks. The allocation and annotation of significantly larger data sets are both costly and time-consuming, rendering it infeasible. As an alternative, we turn to synthetic data sets generated by a Blender add-on called MotorFactory [31]. This add-on initially creates a substantial number of motors with diverse specification, as shown in Fig. 3(a) and Fig. 3(b), outputs them as synthetic single-view point clouds with clamping system from various viewpoints and

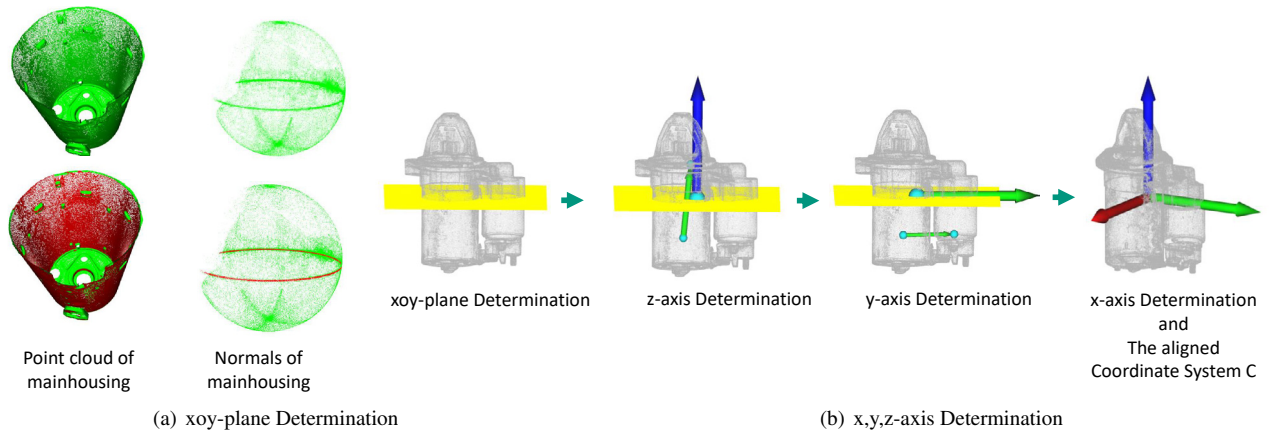


Fig. 4. Normal alignment for motor rotation computation. Three axes directions are computed one by one with the help of part segmentation results.

as synthetic full model point clouds without clamping system as shown in Fig. 3(c) and Fig. 3(d).

Binary segmentation and motor part segmentation are carried out using two structurally identical deep learning neural networks, both of which are based on PCT [29]. The PCT network for binary segmentation is initially pretrained on the synthetic single-view point cloud data set and then fine-tuned on the real-world single-view point cloud data set. Similarly, the PCT network for part segmentation is pretrained and fine-tuned on their corresponding synthetic and real-world data set.

3.3. Registration

To determine the 6D pose of bolts in the motor in clamping system, ${}^{\mathcal{B}}R_{bolt}$ and ${}^{\mathcal{B}}t_{bolt}$, a Transform matrix, $T_{C \rightarrow \mathcal{B}}$ is required. As the full motor, considered as source point cloud, coincide after registration with the motor in clamping system, regarded as target point cloud, the bolts in the source will also coincide with bolts in the target. The transformation matrix obtained during this registration process is the desired $T_{C \rightarrow \mathcal{B}}$. Since errors can result in a detrimental impact on instruments in subsequent process, such as the screwdriver bit on the robot arm, a high level of precision is demanded in our results. Simultaneously, a significant 6D pose difference exists between our source and target, making it challenging to establish correct matches for fine registration methods. Therefore, we divided our approach into two steps: coarse registration and fine registration.

In the coarse registration step, we downsample the point cloud, estimate normals, and compute a 33-dimensional Fast Point Feature Histogram (FPFH) [32] for each point. The FPFH feature comprehensively describes the local geometric properties of a point. These features are then employed in Fast Global Registration [33], the results of which are depicted in Fig. 2(b).

For the fine registration step, we employ Iterative Closest Point (ICP) [14]. As we reduce the maximum matching distance d_{max} , the likelihood of false matches decreases, yielding a smaller registration error. However, it is vital to note that the initial 6D poses of the source and target point clouds must be close, or an insufficient number of matches may result in registration failure. To overcome this limitation, we divide the

Table 1. Different registration methods.

Method	d_{max} (mm)	RMSE (mm)	Process time (s)
point2point	10	4.11	9.64
	10 → 1	4.27e-1	13.82
	10 → 1 → 0.1	7.66e-2	21.93
point2plane	10	7.96e-1	3.72
	10 → 1	3.44e-1	4.65
	10 → 1 → 0.1	7.62e-2	6.39

fine registration process into multiple steps, with the results of each step serving as the source point cloud for the subsequent one, and with a decreasing maximum matching distance d_{max} . As demonstrated in the Fig. 2(c), we implement a three-steps fine registration approach. The rationale behind our choice of this approach, along with its details, is explained in the section 4. This iterative approach results in a precise registration, where the registered full motor point cloud perfectly overlaps with the point cloud of the motor in segmented point cloud.

3.4. Bolt Point Clustering

After part segmentation described in section 3.2, the point cloud is segmented into bolt and five other classes. Given our prior knowledge of the motor structure, points belonging to the same bolt should be grouped together, while those from different bolts should be separated. However, the precise number of bolts on the scanned motor is unknown. Additionally, there exists a certain level of noise, and the shape and size of the bolt point clouds can exhibit variations. These characteristics make the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [34] exceptionally well-suited for this task. We utilize the DBSCAN algorithm to cluster the bolt points into distinct bolt point clouds, with the centers of these bolt point clouds representing the desired 3D positions ${}^C t_{bolt}$.

3.5. Normal Alignment

The alignment of the full motors is as follows: The center point of the motor coincides with the origin of the coordinate

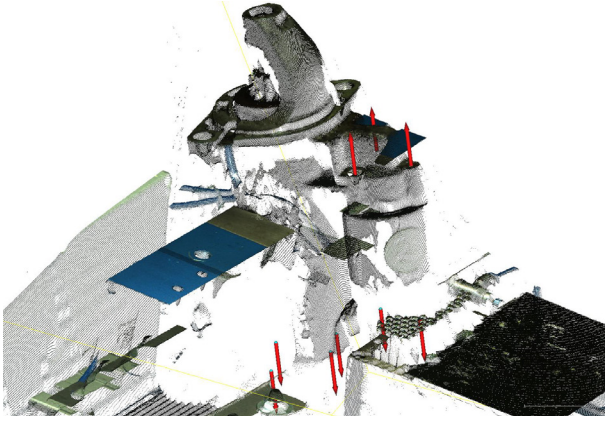


Fig. 5. 6D pose of bolts in a real-world case. Note that the red arrows are the real numerical results computed with our proposed pipeline.

Table 2. Variations on backbones of deep learning architecture.

Model	Overall Accuracy (%)	mIoU (%)	Bolt IoU (%)
PointNet	66.82	59.63	26.10
PointNet++	80.64	72.13	48.16
DGCNN	84.57	75.07	48.04
PCT	90.47	84.58	64.92

system. The z-axis is parallel to the cylindrical axis of a motor part known as the Main Housing. The positive direction of the z-axis points from the Main Housing part towards the Connector part. The y-axis is positioned perpendicular to the z-axis, and it is positioned such that the center of the Solenoid part lies in the yoz plane. Based on prior knowledge, the orientation of the bolts ${}^C R_{bolt}$ can be determined by the orientation of the aligned motor ${}^C R_{motor}$, as represented in Equation 1.

According to the coordinate system definition, the normal vectors of points on the main housing cylinder are parallel to the xoy plane. Based on the segmented full motor point cloud, we obtain the normal vectors for each point on the Main Housing part [35]. These normal vectors are represented as points in space, as shown in the Fig. 4(a). It can be observed that these points form a spherical surface, which is most dense near the equator of the sphere. This equatorial plane corresponds to the motor's xoy plane and is determined by RANSAC [13].

With this plane as a reference, other orientations can be determined as Fig. 4(b). The z-axis should be perpendicular to the xoy plane, directing from the Main Housing to the Connector. The positive direction of the y-axis is determined by projecting the vector from the center of the Main Housing to the center of the Solenoid onto the xoy plane. The x-axis is determined using the z-axis and y-axis with the right-hand rule. The motor's orientation ${}^C R_{motor}$ aligns with the positive z-axis direction.

4. Experimental Results

Registration: We conduct experiments on two ICP variants: one based on point-to-point matches between the source and target, and another relying on matches between points in the

Table 3. Variations on Settings of deep learning architecture.

Point Number	Pretrain	Warm_up	Overall Accuracy (%)	mIoU (%)	Bolt IoU (%)
2048	-	-	79.35	72.80	44.35
2048	-	✓	83.02	76.07	44.80
2048	✓	-	87.58	81.12	59.45
2048	✓	✓	90.47	84.58	64.92
1024	✓	✓	86.21	79.58	53.85
4096	✓	✓	71.89	61.33	33.20
4096 (x2 epoch)	✓	✓	83.49	76.72	55.43

source and planes in the target, which are determined by the points and their normals. Additionally, we test several combinations of maximum matching distances. The results are presented in Table 1, which demonstrates that as the number of fine registration steps increases, processing time exhibits linear growth, and registration accuracy improves significantly. Furthermore, compared with the point-to-point algorithm, the point-to-plane algorithm exhibits notably reduced processing time. In this study, although real-time performance holds importance, we prioritize precision over real-time performance, leading us to choose a point-to-plane registration approach with three fine-registration steps.

Full Model Segmentation: Our primary goal is to identify the most suitable combination of structures and algorithms. Given that our test set comprises only 18 motors, which is extremely small, the results derived from a single train-test split setting lack representativeness. To mitigate this issue, we use a method of cross-validation. In each training, 2 random motors are selected as the test set, and the other 16 motors are used as the training set. We perform such trainings 5 times and report their mean results as the final cross-validated performance.

Four widely-used backbones are evaluated in our experiments, as depicted in Table 2. We find that the PCT [29] yields the most favorable results. Following this, we conducted tests on various hyperparameters based on the PCT, and the most significant experimental results are presented in Table 3. Note that all the experiments in this table use the aforementioned cross-validation. Moreover, for a fairer comparison, we make the 5 train-test splits identical for all the experiments. From it, we can observe that the effectiveness of performing pre-training on the synthetic dataset is evident. To mitigate model instability resulting from an excessively high initial learning rate during transfer learning, using a warm-up strategy can further optimize the performance of our network model. Regarding the input sub-point cloud size for the training, as indicated in the table 3, a sampling size of 2048 points performs the best, even better than a sampling size of 4096 points. Since increased sampling size leads to a reduction of the actual interaction steps in one epoch, an additional experiment of 4096 points with doubled training epochs is performed. However, its performance still does not surpass that of the one with 2048 points.

Binary Segmentation: Binary segmentation is performed on our single-view data set, employing the identical neural network architecture and hyperparameter settings, with the only

modification being made to the output fully connected layer to accommodate a reduced number of classes. And the overall accuracy and mIoU yield the values of 99.80%, and 99.07%. Given the algorithm's already high accuracy and the inherent limitations related to labeling errors that cannot be entirely eliminated, further tests are deemed unnecessary.

Full Pipeline Demo: The Fig. 5 illustrates an example input point cloud in a real-world case. The red arrows depict the computed 6D pose of all 9 bolts. As shown in Fig. 5, our method can accurately determine the 6D pose of bolts, even in cases where they have not been precisely scanned or are occluded.

5. Conclusion

This study presents a novel pipeline designed for the precise determination of the 6D pose of bolts leveraging prior knowledge within a remanufacturing line. By utilizing a pre-acquired model, we perform part segmentation, normal alignment, and bolt point clustering to extract valuable information, which includes bolt 6D pose on the full motor and the full motor point cloud. This extracted knowledge is subsequently transferred to the motors being processed through binary segmentation and registration steps, so that the 6D pose of bolts on the processed motors is determined. The experimental results illustrate the effectiveness of our pipeline even in challenging case characterized by low-quality scans and occlusion.

Acknowledgements

The AgiProbot project is funded by the Carl-Zeiss Foundation.

References

- [1] E. Sundin. *Product and process design for successful remanufacturing*. PhD thesis, Linköping University Electronic Press, 2004.
- [2] A. P. Barquet et al. An integrated approach to remanufacturing: model of a remanufacturing system. *Journal of Remanufacturing*, 3:1–11, 2013.
- [3] M. I. Rizova et al. A systematic review of decision-making in remanufacturing. *Computers & Industrial Engineering*, 147:106681, 2020.
- [4] J. Kurilova-Palisaitiene et al. Remanufacturing challenges and possible lean improvements. *Journal of Cleaner Production*, 172:3225–3236, 2018.
- [5] G. Lanza et al. Agiles produktionsystem mittels lernender roboter bei ungewissen produktzuständen am beispiel der anlasser-demontage. *at-Automatisierungstechnik*, 70(6):504–516, 2022.
- [6] C. Wu et al. Sim2real transfer learning for point cloud segmentation: An industrial application case on autonomous disassembly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4531–4540, 2023.
- [7] C. Wu et al. Synmotor: A benchmark suite for object attribute regression and multi-task learning. In *VISGRAPP*, 2023.
- [8] B. Baud-Lavigne et al. Mutual impacts of product standardization and supply chain design. *International Journal of Production Economics*, 135(1):50–60, 2012.
- [9] A. Albers et al. Reference products in pge-product generation engineering: analyzing challenges based on the system hierarchy. *Procedia CIRP*, 70:469–474, 2018.
- [10] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pp. 1150–1157. Ieee, 1999.
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [12] S. Peng et al. Pvnnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570, 2019.
- [13] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [14] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pp. 586–606. Spie, 1992.
- [15] X. Chen et al. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.
- [16] C. R. Qi et al. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 918–927, 2018.
- [17] J. Ku et al. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8. IEEE, 2018.
- [18] D. Xu et al. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 244–253, 2018.
- [19] C. Wang et al. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3343–3352, 2019.
- [20] C. R. Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.
- [21] F. Duffhauss et al. Mv6d: Multi-view 6d pose estimation on rgb-d frames using a deep point-wise voting network. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3568–3575. IEEE, 2022.
- [22] J. Chen et al. Survey on 6d pose estimation of rigid object. In *2020 39th Chinese Control Conference (CCC)*, pp. 7440–7445. IEEE, 2020.
- [23] D. Fernandes et al. Point-cloud based 3d object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, 68:161–191, 2021.
- [24] B. Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1513–1518. IEEE, 2017.
- [25] C. R. Qi et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [26] Y. Wang et al. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.
- [27] C. R. Qi et al. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286, 2019.
- [28] A. Vaswani et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [29] M.-H. Guo et al. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021.
- [30] Y. Wu et al. Mpct: Multiscale point cloud transformer with a residual network. *IEEE Transactions on Multimedia*, 2023.
- [31] C. Wu et al. Motorfactory: A blender add-on for large dataset generation of small electric motors. *Procedia CIRP*, 106:138–143, 2022.
- [32] R. B. Rusu et al. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pp. 3212–3217. IEEE, 2009.
- [33] Q.-Y. Zhou et al. Fast global registration. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pp. 766–782. Springer, 2016.
- [34] M. Ester et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pp. 226–231, 1996.
- [35] H. Hoppe et al. Surface reconstruction from unorganized points. In *Proceedings of 19th conference on computer graphics and interactive techniques*, pp. 71–78, 1992.