Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 106 (2022) 138–143

www.elsevier.com/locate/procedia

9th CIRP Conference on Assembly Technology and Systems

MotorFactory: A Blender Add-on for Large Dataset Generation of Small Electric Motors

Chengzhi Wu *[a], Kanran Zhou[a], Jan-Philipp Kaiser[b], Norbert Mitschke[c], Jan-Felix Klein[d], Julius Pfrommer[e], Jürgen Beyerer[a,e], Gisela Lanza[b], Michael Heizmann[c], Kai Furmans[d]

[a] *Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*
[b] *wbk Institute of Production Science, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*
[c] *Institute of Industrial Information Technology, Karlsruhe Institute of Technology, Hertzstraße 16, 76187 Karlsruhe, Germany*
[d] *Institute for Material Handling and Logistics, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany*
[e] *Fraunhofer 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

Abstract

To enable automatic disassembly of different product types with uncertain condition and degree of wear in remanufacturing, agile production systems that can adapt dynamically to changing requirements are needed. Machine learning algorithms can be employed due to their generalization capabilities of learning from various types and variants of products. However, in reality, datasets with a diversity of samples that can be used to train models are difficult to obtain in the initial period. This may cause bad performances when the system tries to adapt to new unseen input data in the future. In order to generate large datasets for different learning purposes, in our project, we present a Blender add-on named MotorFactory to generate customized mesh models of various motor instances. MotorFactory allows to create mesh models which, complemented with additional add-ons, can be further used to create synthetic RGB images, depth images, normal images, segmentation ground truth masks and 3D point cloud datasets with point-wise semantic labels. The created synthetic datasets may be used for various tasks including motor type classification, object detection for decentralized material transfer tasks, part segmentation for disassembly and handling tasks, or even reinforcement learning-based robotics control or view-planning.

© 2022 The Authors. Published by Elsevier Ltd.
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 9 th CIRP Conference on Assembly Technology and Systems

Keywords: Remanufacturing; Blender Add-on; Machine learning; Large synthetic dataset generation

1. Introduction

Today's industrial landscape is characterised by linear economies. End-of-life (EOL) strategies for products such as remanufacturing, in which used products are reprocessed, offer the potential to decouple resource consumption from sustainable economic growth [25]. During the remanufacturing process, the products are inspected and disassembled, then individual parts are reworked or exchanged and again reassembled into final products [24]. In contrast to related EOL-strategies, remanufactured products ensure functionality and quality that is equivalent or better compared to a new product [25]. Remanufacturing systems face various challenges originating from uncertain product states, inconsistent quality and fluctuating availability of products. Consequently, even today the vast majority of processes in a remanufacturing system are carried out manually [14]. In order to automate these processes, agile production systems consisting of autonomously operating subsystem are required which provide the highest possible flexibility and adaptability.

In this paper, we consider the automated disassembly of different variants of end-of-life actuators which are commonly used in vehicle manufacturing, e.g., as seat adjuster motors, window lift motors or rear door motors. As shown in figure 1, the considered subsystems are the intralogistics, the inspection, as well as the disassembly which all heavily rely on product information. However, in remanufacturing, each core is unique since products have a high variance concerning their product state. Therefore, handling, inspection and disassembly strategies can not be defined in advance. Accordingly, the system

2212-8271 © 2022 The Authors. Published by Elsevier Ltd.
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 9 th CIRP Conference on Assembly Technology and Systems
10.1016/j.procir.2022.02.168

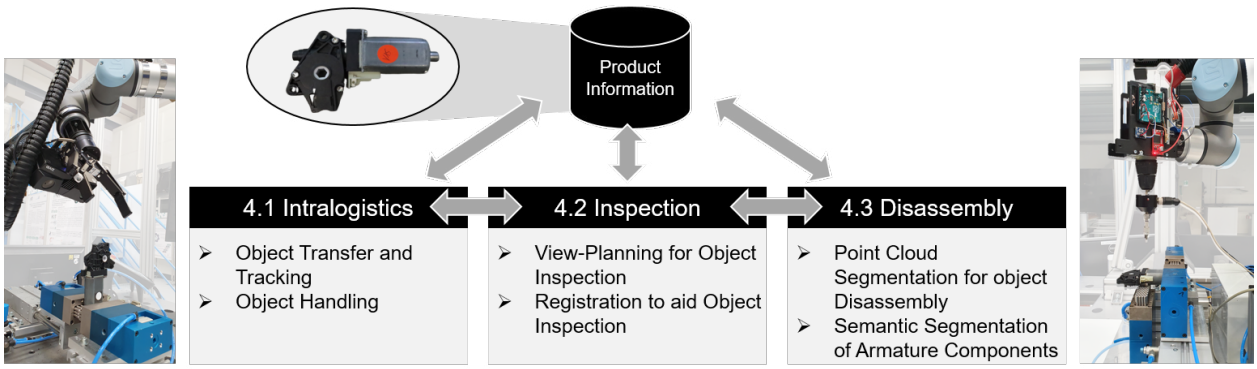


Fig. 1: Product-based relationships and considered applications between intralogistics, inspection and disassembly in an automated remanufacturing system.

must derive and execute these strategies during runtime based on the actuator at hand.

Machine learning methods, and especially deep learning methods, may be the key to achieve the necessary robustness to deal with the high degree of variability. By learning the internal structure on part level, (e.g. gear container, pole pot, electrical connection), processes on unseen variants which have similarities to the known population of actuators become feasible. However, a major disadvantage of machine learning methods is the required amount of data. Allocating and annotating a large amount of data is time-consuming or even impossible in reality. In the recent past, the synthetic generation of training data for machine learning methods has become a popular alternative. By leveraging transfer learning, it allows to train models that rely less on elaborately labeled real-world data. We therefore present an approach for generating synthetic training dataset as well as its relevant applications for the handling-, inspection-, and disassembly processes.

The remainder of this paper is structured as follows: Section 2 summarizes the state of the art of 3D synthetic dataset creation and application. Section 3 describes technical details on the developed Blender add-on. Section 4 provides brief overviews on applications build on top of the add-on while section 5 summarizes presented outcome and discusses future work.

2. State of the art

Since the appearance of the iconic *Stanford Bunny*, generating synthetic datasets as training data for machine learning purposes has already been widely discussed and used as a possible learning approach for various computer vision applications. Regarding synthetic dataset of 3D models, the Princeton Shape Benchmark [23] provides a collection of 1,814 polygonal models of objects from different categories as an early dataset. ModelNet [30] contains 127,915 models for 3D object classification and retrieval. More than three million annotated 3D models are collected in ShapeNet [2]. Its subsequent work PartNet [17] additionally offers fine-grained semantic segmentation information for a subset of the models. By utilizing Thingiverse, Thingi10K [32] provides a large dataset of 3D-printing models. More recently, the ABC dataset [12] collects over 1 million CAD models, including lots of mechanical parts with sharp edges and well defined surfaces, which are seldom included in the previous synthetic datasets.

Regarding 3D scenes, [27] generates a synthetic dataset for the segmentation and detection of objects in virtual street scenes. Also [21] and [10] consider urban scenes and each provides a dataset for semantic segmentation in these environments. Other approaches in the image domain deal with the generation of images from garden scenes [15], or specifically for object detection and pose estimation [26]. There are also approaches for the generation of point clouds, such as that of [6], in which, in contrast to the previously mentioned work, point clouds of urban scenes are generated using Blender. Another work using Blender deals with automatic generation of point clouds of historical objects [18].

However, in the production environment for industrial applications, approaches of generating synthetic dataset are seldom employed. They may contribute in various applications including product classification, segmentation of product components, product tracking, and even determination of grasp points.

3. MotorFactory Add-on

The AgiProbot project aims for auto-detection, tracking and disassembly of end-of-life products. To be specific, in our current experimental setting, we work on small electric motors used in vehicle manufacturing. Universal method needs to be developed to deal with various types of motors, including the ones with unseen specifications. However, we are only provided with a handful of motors with only few different product specifications. The variance of data is actually not sufficient for training with machine learning methods, especially those deep learning-based ones. To deal with this problem, we created a Blender add-on named MotorFactory, which can generate motor mesh models with a variety of specifications based on the motor types we have currently.

As an open source software, Blender is a proven tool that performs well in modeling shapes and creating highly customizable add-ons. Our MotorFactory add-on is able to generate mesh models with various specifications and save them in desired file formats. Each component of a generated motor can

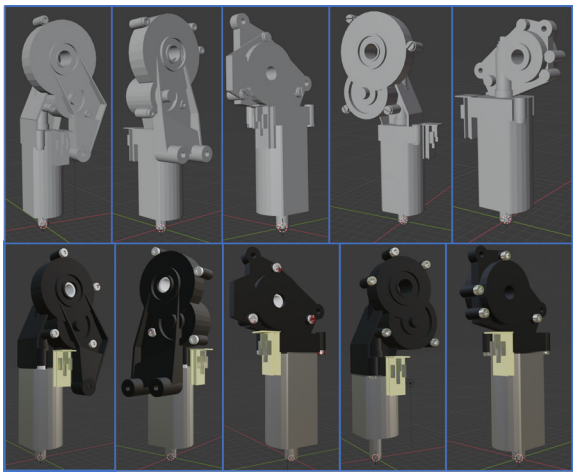


Fig. 2: Generated demo motors. Upper row: no textures added; bottom row: textures added and rendered. Column 1/2/4: Type-A motors with two gears; column 3/5: Type-B motors with one gear.

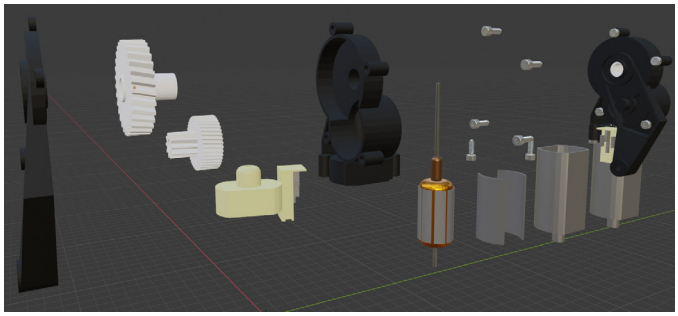
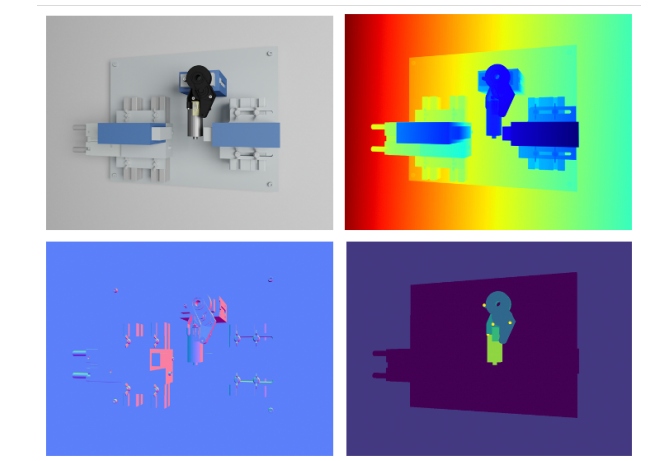


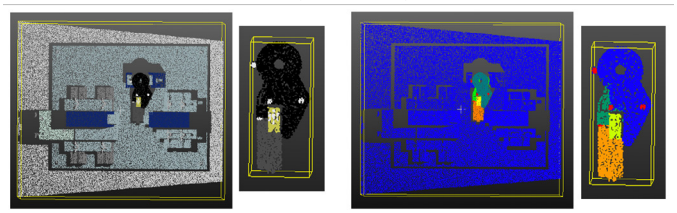
Fig. 3: An explosion figure of a motor generated with Version 2.0. The original assembled motor model is also shown at the right most.

also be saved separately. Considering different requirements, the add-on is implemented in two versions. Version 1.0 only considers the components of motors that can be directly observed from the appearance, while Version 2.0 further considers the inner components. The generated models of both versions contain the following components: (i) Pole Pot; (ii) Electric Connection; (iii) Gear Container; (iv) Cover and (v) Bolts. Regarding inner components in version 2.0, the additionally generated parts are: (vi) Magnets; (vii) Armature; (viii) Lower Gear and (ix) Upper Gear. To generate motors with various specifications, we provide lots of parameter options that control the type, size, position and rotation of different parts of motor, e.g. bolt position, gear size, or pole pot length. Additionally, both versions provide multiple bolt generation options to meet different requirements.

To better design the building graph of different types of motors, we define two basic types based on the number of gears in a motor. Type-A motor indicates the motors with two gears inside, while Type-B motor indicates the motors with only one gear. Each motor type has different kind of gear containers. Different gear containers further have different covers, which come with different mounting points. Three options of extension shapes for covers are provided for Type-A motors, while two options of are provided for Type-B motors. Figure 2 shows



(a) Demo of generated image dataset



(b) Demo of generated point cloud dataset

Fig. 4: Demos of generated dataset: (a) a rendered scene image with its corresponding depth image, normal image, and segmentation ground truth image; (b) a simulated scene point cloud with its point-wise segmentation ground truth.

ten generated demo motors with different parameters. Figure 3 shows an exploded view of a demo motor generated with Version 2.0. All the individual components mentioned above are modeled separately as illustrated.

The MotorFactory can be used to create a large amount of different motor variants which are further used to generate synthetic image and point cloud datasets. For example, to create an image datasets, apart from the scene images rendered by Blender directly, we can use BlenderProc [4] to generate the corresponding depth images, normal images, and segmentation ground truth images as illustrated in figure 4a. To create point cloud datasets, we can use Blensor [7] to simulate the sensors. Figure 4b gives a demo of generated scene point cloud with its segmentation ground truth. The dataset generation setting and pipeline depend on the actual needs of different tasks, hence they may vary from task to task.

4. Application Use-Cases for the Motor Factory Add-On

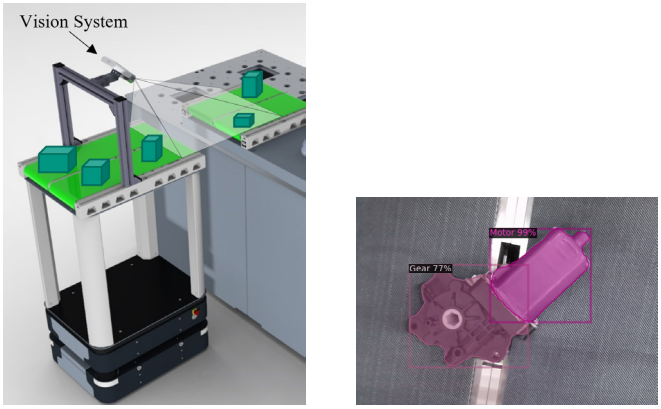
The ability to generate synthetic motor data for computer vision tasks enables the development or improvement of a variety of different use-cases inside the interdisciplinary AgiProbot project. The presented use-cases follow an exemplary product flow inside the AgiProbot demo-factory as illustrated in figure 1. The intralogistics system is responsible to transfer the product to a station and handle it with a vision based manipulator (4.1). Depending on the station type, inspection related tasks (4.2) or disassembly procedures (4.3) are carried out.

4.1. Intralogistics

Object Transfer and Tracking: The AgiProbot production system follows the Fluid Automation framework to achieve the required changeability needed for automated remanufacturing [31]. Realizing the material flow between individual stations in a matrix layout is the main task of the embedded intralogistics system [11]. This includes the transfer of objects between an autonomous mobile robot (AMR) and a transfer unit mounted on a station module, as illustrated in figure 5a. From a material flow perspective, objects are components such as products, parts or assemblies or carrier devices such as boxes or workpiece holders. The object transfer control workflow includes a vision system mounted on the AMR which is used to detect and track objects on the transfer units. Detecting objects and estimating their pose is essential to realize the decentralized control between the AMR and the transfer unit. For known objects, 2D object detection algorithms can be used to estimate the object's bounding boxes and track them over time. However, well performing deep learning based approaches, e.g. YOLO [1] and its tracker DeepSort [29], require a vast amount of labeled input data for the training process. In our case, by feeding generated motor models to Blender and using a BlenderProc-based pipeline to generate synthetic image data, one can create high quality rendered images with accurate ground truth annotations [4]. Enriched with labeled real-world images, one can create a dataset of reasonable size to train well performing real-time detection and tracking network models.

Object Handling: Object handling describes vision-based robotic pick-and place tasks, which are commonly required to meet necessary preconditions to perform value-adding tasks like manufacturing, assembly or disassembly. A robotic grasping system commonly consists of a robotic manipulator, an RGB-D camera, a gripper and a target object. The robotic grasping problem can be generally divided into three subproblems, namely the grasp detection, the trajectory planning and the execution [13].

Hereby, the grasp detection system is the key entry point and includes the target object localization, the object pose estimation, and the grasp estimation [5]. To solve the localization and pose estimation problem for known objects, simple boundingboxes, as described in the previous use-case, are not sophisticated enough. However, since the MotorFactory generates mesh models on part level, annotated instance segmentation masks can be generated. Training a segmentation network like Mask R-CNN [9] allows to detect pixel-level segmentation instances of classes. Being able to detect parts individually allows to handle sub-assemblies and deriving pre-defined grasping points on part level. Figure 5b shows the segmentation results on a real-world image with a network model solely trained on synthetic data. Trained an instance segmentation network on many artificially generated product variants increases its robustness in detecting individual parts when a new variant is exposed to the system. In an adaptive vision system, compared to objects that are categorized to be unknown which only rely on salient detection, a never seen motor variant can therefore be detected with the help of more sophisticated methods .



(a) Object transfer scenario (b) Instance segmentation

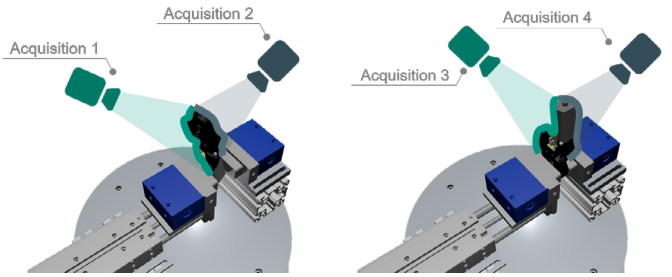
Fig. 5: Object transfer and handling, using a motor of Type-A as an example.

4.2. Inspection

View-Planning for Object Inspection: Since wear and tear or defects can occur anywhere on a returning product, it is usually necessary to inspect the entire surface of the product. The formalized problem of completing product surface coverage using a robot-based optical system can be expressed as a view-planning problem. In the literature so far, there are model-based and non-model-based solution approaches [22]. Model-based approaches determine robot poses for sensor sensing by pre-planning based on a prior model. Non-model-based approaches determine the next best robot pose for acquisition at runtime through maximizing certain criteria. However, in many cases, independent remanufacturers do not have product information such as mesh models of the product at hand due to confidentiality issues [8]. This hinders efficient pre-planning of visual strategies for inspection.

With the help of this add-on, an important research gap can be approached. In remanufacturing, the product variants considered are mostly different in detail but similar in principle. By using the generated dataset and machine learning methods such as reinforcement learning, inspection strategies can be learned in the simulation and transferred to previously unknown but similar product variants under sufficient generalization capability. In this case, the reinforcement learning agent generates an action (pose of the robot's end effector in space) based on a state (currently recorded point cloud of the product) and receives a reward (e.g. based on the relative information gain in the form of new detected surfaces). The reinforcement learning agent is trained in the simulation until its performance is satisfactory. In theory, the amount of data required for validation in reality then becomes smaller and the inspection agent can be used to fulfil inspection goals defined by the reward signal for a large number of different but similar variants of electric motors.

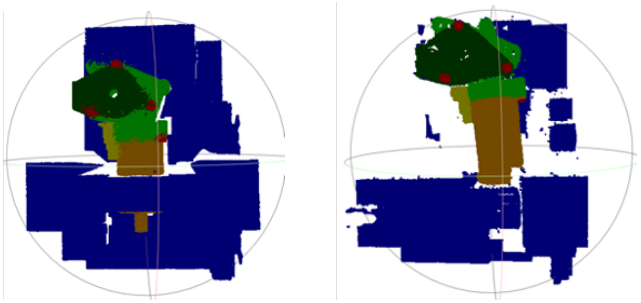
Registration to aid Object Inspection: The requirements of completing surface coverage for objects can usually not be satisfied with one-time application of view-planning approaches. This is due to the fact that during inspection some surface areas of a product are permanently occluded by grip-



(a) First clamping setup with the gear container facing upwards

(b) Second clamping setup with the pole pot facing upwards

Fig. 6: Visualization of an exemplary inspection routine with two clamping setups to enable full target object inspection.



(a) Synthetic point cloud

(b) Real-world point cloud

Fig. 7: Comparison between (a) a synthetic point cloud with labels generated automatically, and (b) a real-world point cloud from the Zivid camera with points annotated manually. Using a motor of Type-B as an example.

ping or clamping units, or simply because the measured object is placed on a surface. After a successful product acquisition, for example with the approach described in view planning, a repositioning as well as the acquisition of the previously hidden surfaces is necessary. Since the repositioning changes the position of the object, a registration is necessary to merge the point clouds of all capturing steps in order to determine the entire object surface captured so far as well as to plan the next view based on.

Random sample consensus (RANSAC)-based approaches are widely used for solving the registration problem, but they are model-free, computationally expensive and do not take the specific product properties into account [33]. Here, the presented add-on offers the possibility to apply data-driven registration approaches with deep learning-based methods, which have proven successful compared to classical approaches [3]. Synthetic point clouds of a product model can be generated from different views of a sensor, whose pose is determined in the form of the position and the orientation. This also allows the transformation matrix between the acquired point clouds to be determined and can be used as a target variable for registration procedures to learn. In this way, product-specific properties can be taken into account during the training of the virtual registration process in order to achieve a faster and more precise registration result for real-world data.

4.3. Disassembly

Point Cloud Segmentation for Object Disassembly: Compared to images, point clouds contain more 3D information and is another widely used data representation in industrial applications. Point cloud segmentation provides an alternative way to segment the different components of a product to provide important information to robots. In recent years, methods of applying deep learning techniques on 3D point clouds have been intensively investigated. PointNet [19] pioneered on this topic by designing a symmetric function for unordered point input, followed by works like PointNet++ [20] or DGCNN [28] which focused on gathering local information. Apart from classification, those methods also focus on point cloud segmentation including part segmentation on 3D shapes and scene segmentation for indoor scenes.

In our case, the target object is set fixed in a fixture when it is transferred to the disassembly station. Then a real-world point cloud is obtained with Zivid 3D Camera. However, the variance of different taken point clouds, especially regarding the motors, is not sufficient for learning purposes. In this case, we create a synthetic dataset using the proposed add-on. A demo is given in Figure 4b. Apart from the station table, the scene contains a fixure with a random motor placed in a relatively fixed position. To solve the problem of batches are not representative enough due to the large size of raw point clouds, we cut a cuboid of sub-point cloud that only considers the target object and its neighbor points after generating them. Further data augmentation is needed to enhance the generalization ability of the network, e.g. adding random small mask panels, adding random jitter noise, shifting fixure components, or rotating virtual cameras. Figure 7 gives a comparison between a synthetic point cloud and a real-world point cloud. Their scene settings are slightly different, but the synthetic one is already good enough to use as training data. After the network is pre-trained with the synthetic dataset, transfer learning may be applied with a few real-world data to finetune the model. Further important specification information may also be obtained with additional post-processing and subsequently provided to robots for other tasks.

Semantic Segmentation of Armature Components: The armature represents a functionally relevant part of the motor that requires a separate inspection in the given remanufacturing task. Semantic segmentation is required for the recognition of important surface parts of a armature. In a first proof-of-work approach, a small dataset of armatures has already been created to train a U-Net [16] for the segmentation task. However, the dataset contains only a small variety of armatures with 75 training images, most of them are taken from a similar perspective and are even partially inaccurately or inconsistently annotated. Hence, the proposed Blender add-on can be used to generate additional training data to supplement or replace the given dataset. Different armature types and wear conditions can be obtained through suitable parameterization. Arbitrary perspectives on the armature can also be generated. Furthermore, the accurate annotation of the synthetic dataset improves the segmentation performance and also increases the meaningfulness of validation metrics.

5. Conclusion and Outlook

In this paper, we present MotorFactory, a blender add-on which allows the end-user to automatically create a variety of different small electric motor models and components. Based on the generated motors, synthetic datasets can be created for different computer vision tasks. Being able to generate and annotate product models easily may further increase the level of automation in challenging remanufacturing environments. Inside the interdisciplinary AgiProbot project, we identified several use-cases from the intralogistics, inspection, and disassembly domains. All of the use-cases benefit greatly from having access to synthetic product models.

In the future, an open-source benchmark dataset for object attribute estimation based on the add-on will be created and published. New versions of the presented add-on will include a greater amount of other electric motor types and variants, e.g. starters, or even other common remanufacturing products. Textures, which increase the level of realism, e.g. worn-out screw heads, rust or discolouration, are also intended to be added. The elaborated use-cases will be fully implemented and evaluated. Feedback based on the evaluation can then be used to further improve the add-on and its features.

Acknowledgements

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

References

[1] Bochkovskiy, A., Wang, C.Y., Liao, H., 2020. Yolov4: Optimal speed and accuracy of object detection. ArXiv abs/2004.10934.

[2] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. Shapenet: An information-rich 3d model repository. ArXiv abs/1512.03012.

[3] Choy, C., Dong, W., Koltun, V., 2020. Deep global registration. CVPR , 2511–2520.

[4] Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H., 2019. Blenderproc. arXiv preprint arXiv:1911.01911 .

[5] Du, G., Wang, K., Lian, S., Zhao, K., 2021. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review 54, 1677–1734.

[6] Griffiths, D., Boehm, J., 2019. Synthcity: A large scale synthetic point cloud. ArXiv abs/1907.04758.

[7] Gschwandtner, M., Kwitt, R., Uhl, A., Pree, W., 2011. Blensor: Blender sensor simulation toolbox, in: ISVC.

[8] Hascoet, J., Touzé, S., Rauch, M., 2017. Automated identification of defect geometry for metallic part repair by an additive manufacturing process. Welding in the World 62.

[9] He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask r-cnn, in: ICCV, IEEE. pp. 2980–2988.

[10] Khan, S., Phan, B., Salay, R., Czarnecki, K., 2019. Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks, in: CVPRW, pp. 88–96.

[11] Klein, J.F., Wurster, M., Stricker, N., Lanza, G., Furmans, K., 2021. Towards ontology-based autonomous intralogistics for agile remanufacturing production systems, in: ETFA, IEEE. pp. 01–07.

[12] Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D., 2019. Abc: A big cad model dataset for geometric deep learning. CVPR , 9593–9603.

[13] Kumra, S., Kanan, C., 2017. Robotic grasp detection using deep convolutional neural networks, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 769–776.

[14] Kurilova-Palisaitiene, J., Sundin, E., Poksinska, B., 2018. Remanufacturing challenges and possible lean improvements. Journal of Cleaner Production 172, 3225–3236.

[15] Le Hoang-An, Mensink, T., Das, P., Karaoglu, S., Gevers, T., 2021. Eden: Multimodal synthetic dataset of enclosed garden scenes. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) , 1578–1588.

[16] Mitschke, N., Heizmann, M., 2020. Semantische Segmentierung von Ankerkomponenten von Elektromotoren, in: Forum Bildverarbeitung 2020, pp. 329–340.

[17] Mo, K., Zhu, S., Chang, A., Yi, L., Tripathi, S., Guibas, L., Su, H., 2019. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding .

[18] Pierdicca, R., Mameli, M., Malinverni, E., Paolanti, M., Frontoni, E., 2019. Automatic generation of point cloud synthetic dataset for historical building representation, in: AVR, pp. 203–219.

[19] Qi, C., Su, H., Mo, K., Guibas, L., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR , 77–85.

[20] Qi, C., Yi, L., Su, H., Guibas, L., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: NIPS.

[21] Ros, G., Sellart, L., Materzynska, J., Vázquez, D., López, A.M., 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. CVPR , 3234–3243.

[22] Scott, W., Roth, G., Rivest, J.F., 2003. View planning for automated three-dimensional object reconstruction and inspection. ACM Comput. Surv. 35, 64–96.

[23] Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., 2004. The princeton shape benchmark. Proceedings Shape Modeling Applications, 2004. , 167–178.

[24] Sundin, E., 07.12.2004. Product and Process Design for Successful Remanufacturing. Ph.D. thesis.

[25] Tolio, T., Bernard, A., Colledani, M., Kara, S., Seliger, G., Duflou, J., Battaïa, O., Takata, S., 2017. Design, management and control of demanufacturing and remanufacturing systems. Cirp Annals-manufacturing Technology 66, 585–609.

[26] Tremblay, J., To, T., Birchfield, S., 2018. Falling things: A synthetic dataset for 3d object detection and pose estimation. CVPRW , 2119–21193.

[27] Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C., 2017. Learning from synthetic humans. CVPR , 4627–4635.

[28] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M., Solomon, J., 2019. Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) 38, 1 – 12.

[29] Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 3645–3649.

[30] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. CVPR , 1912–1920.

[31] Wurster, M., Häfner, B., Gauder, D., Stricker, N., Lanza, G., 2021. Fluid automation — a definition and an application in remanufacturing production systems. Procedia CIRP 97, 508–513.

[32] Zhou, Q., Jacobson, A., 2016. Thingi10k: A dataset of 10, 000 3d-printing models. ArXiv abs/1605.04797.

[33] Zhou, Q.Y., Park, J., Koltun, V., 2016. Fast global registration, in: ECCV, pp. 766–782.