Journal of Remanufacturing (2026) 16:1
https://doi.org/10.1007/s13243-025-00157-8

RESEARCH

Semantic 3D product modelling for automated inspection in remanufacturing processes

Jan-Philipp Kaiser[1] · **Dominik Koch**[1] · **Florian Stamer**[1] · **Jef Peeters**[2,3] · **Gisela Lanza**[1]

Received: 17 February 2025 / Accepted: 3 October 2025

Abstract

Remanufacturing is a key process for enabling a circular economy by restoring used products, often referred to as cores, to a like-new condition. It still heavily relies on manual work. One exemplary manual task is the visual inspection of cores before further processing. This manual effort arises from uncertainties, such as varying product conditions, a broad variety of product variants, and the lack of product information to support automation. To address these challenges and to enable the automation of visual inspection tasks, flexible and adaptive inspection systems are required. These systems must be capable of automatically detecting defects and performing product-specific inspections across a wide range of product variants. Therefore, this work introduces a semantic 3D product modelling method that integrates two-dimensional (images) and three-dimensional (point cloud) data. With the resulting semantic 3D model, the semantic information, such as detected defects and components requiring closer inspection, can be encoded on a geometric product model. This model provides the information basis for an adaptive inspection approach. Using semantic (U-Net) and instance segmentation approaches (Mask-RCNN), the proposed method assigns each surface point of the 3D model to a specific component, thereby creating the semantic 3D product model during the inspection process. The results show that the method presented can achieve semantic 3D modelling both in accuracy and model completeness, encoding component information on the geometric product model. Furthermore, we show that the U-Net architecture used to detect components is also able to detect corrosion as one exemplary defect type, enabling the encoding of various semantic information into the semantic 3D product model. This semantic 3D product model then enables targeted individual inspection of these semantically mapped components and defects on the product model in a later stage of an automated inspection procedure.

Introduction

Remanufacturing involves dismantling used products (cores), reprocessing or replacing components, and reassembling them to produce products with the same or higher functionality as new ones [1]. While industrially scalable [2], remanufacturing heavily relies on

---

Extended author information available on the last page of the article

manual labour [3], leading to high operational costs [4] that hinder adoption, especially in high-wage countries. This reliance stems from uncertainties, such as unpredictable reverse value chains, small batch sizes, high variant diversity, and variable core conditions [4].

Before remanufacturing, a trained operator often manually inspects cores visually for defects and quality flaws. This inspection concerns defects on the whole core (e.g., corrosion) and certain core components (e.g., inspection of wear). Core acceptance criteria, such as [5] and [6], give insights into often appearing defects on used products. The operator determines whether a core is suitable for remanufacturing based on the defects and the components on which they occur based on these criteria. Automating this process is challenging due to the following factors:

1. Defects may occur on any part of the product and vary in scale and location, making targeted inspection planning difficult. Diverse types of defects (e.g., corrosion, wear, cracks) appear across various components on used products [7, 8].
2. Product variants are often unknown at the time of inspection, and relevant product information, such as geometric or CAD, may be unknown, particularly for independent remanufacturers. This limits the applicability of pre-defined inspection plans and makes adaptable systems necessary [2, 9, 10].

These factors make the automation very complex, since pre-planning of inspection routines is not possible. For instance, inspection parameters, such as sensor poses or scanning paths, must be dynamically adjusted for each product, as variants differ in size and shape. It may not be known beforehand which variant must be inspected since product identification has not yet been conducted. Robot-guided visual acquisition systems offer the necessary flexibility and can adapt to such uncertainties at runtime by using suitable algorithms. However, this is particularly challenging when the exact location of components requiring detailed inspection is unknown due to missing product information.

Therefore, the objective of this work is to develop and evaluate a semantic 3D product modelling approach to enable adaptive, automated visual inspection in remanufacturing. Using RGB and depth data from a robot-guided acquisition system, the method reconstructs the product geometry as a 3D point cloud and enriches each spatial point with semantic labels derived from deep learning-based segmentation. These labels encode component affiliation. This method is called semantic mapping [11]. The resulting semantic 3D model serves as a structured, inspection-oriented data representation that allows targeted inspection of specific regions of interest without requiring prior CAD models. As a use case, the method is applied to starter motors - a representative product already being remanufactured - whose components (housing, solenoid, gear, carrier, and electrical connection) are individually segmented and semantically mapped onto the surface geometry by the semantic 3D model.

Related work

Approaches to semantic mapping are already well-established in the field of mobile robotics [12]. Optical acquisition systems, in particular RGB-D sensors, are often used for semantic mapping due to their ability to acquire data in a non-contact, fast, and cost-effective manner [9]. These systems capture 3D geometry and image data simultaneously. They generate colour-coded point clouds in which spatial and visual information are precisely aligned [13]. By processing this image and/or 3D geometry data, contextual information can be extracted and mapped directly onto a 3D model, resulting in semantic 3D model construction. Context information in semantic mapping refers to spatial, temporal, and functional data that enhance the interpretation of objects and their relationships within an environment. Techniques for extracting such contextual information are methods like image segmentation approaches based on deep neural networks. This work follows the definition of semantic segmentation from [14], where it is defined as the assigning of a label to a set of random variables $\mathrm{X} = \{x_1, x_2, \ldots, x_N\}$ which usually is a 2D image with $W\, x\, H = N$ pixels. Each label $\mathcal{L}$ in the label space $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ represents a different class or object, where k+1 is the size of the label space, with l_0 representing the background. Segmentation approaches can consequently be used to detect and localize regions of interest, for example, components or defects on a product [8] that need to be detected/segmented.

Despite these benefits, existing work in remanufacturing only addresses 3D product modelling without integrating contextual information. Such approaches focus primarily on additive processes during remanufacturing, where the ideal geometry of the core to be remanufactured is known. The core geometry is typically captured using techniques such as structured light or laser scanning [15] to reconstruct the core geometry. Reconstruction refers to the process of generating such 3D geometric models from available data sources, such as point clouds or even 2D images [16]. These reconstructed 3D models allow deviation analysis between the core and the ideal geometry, facilitating path planning for additive repairs [17]. Existing work using acquired 3D models also includes defect localization and classification (e.g. wear, cracks) [7], geometry matching [18], and voxel-based reconstruction for aircraft engine parts [19]. Reconstruction therefore aims to approximate the original shape as closely as possible within the constraints of the chosen representation. Further approaches in Remanufacturing, using reconstruction approaches, integrate defect detection and tool path generation [20], while others specifically address the development of 3D acquisition systems [21], nominal geometry estimation [22], or image-based defect identification [23]. Beyond reprocessing, 3D models are increasingly being used for inspection, measurement, disassembly, and robot path planning when CAD models are not available. Techniques such as Visual Structure from Motion (VFM) enable cost-effective model generation [9]. These models support downstream processes, such as ultrasonic non-destructive testing, via robot path planning [10].

The vision of an adaptive inspection system using semantic 3D modelling

The concept of adaptive inspection for defect detection and evaluation considered in this work follows a two-stage strategy, combining an overall inspection (a) of the entire product with a targeted individual inspection (b) of specific components. Figure 1 illustrates this approach. First, a general inspection is conducted, in which the entire surface of the inspection object is scanned. This is necessary, on the one hand, since defects may appear on the whole surface of a used product (e.g., corrosion). On the other hand, the data collected in this process is used to generate a semantic 3D model of the object, which is the main subject in the remainder of this work.

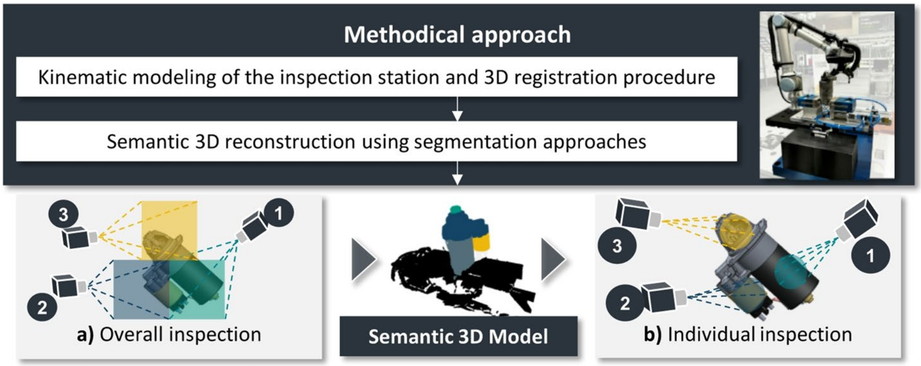


Fig. 1 Overview of the vision of an adaptive inspection system using semantic 3D modelling, including the methodical approach of this work

The novelty of the resulting semantic 3D model is that it not only represents the geometry of the object but also encodes specific object surfaces that require detailed inspection. For example, surface areas belonging to components that are susceptible to defects (e.g., wear or corrosion) can be encoded accordingly. Furthermore, defects detected in the overall inspection can be encoded, whereby this semantic 3D model can then be used as a basis of information for the individual inspection of the defects. In contrast to prior approaches in remanufacturing that use 3D reconstruction mainly for deviation analysis based on known CAD data [18, 19], this work presents a semantic 3D modelling method that combines 2D image and 3D point cloud data through deep learning-based segmentation. This enables real-time component and defect annotation during inspection, even in the absence of prior product information. This differentiates the approach from commercially available and often closed systems on the software side, such as the Zeiss ScanCobot [24] or the Keyence VL 3D-Scanner [25], which purely generate 3D models for geometric deviation analysis.

The semantic 3D model then enables individual inspection by a targeted inspection of specific regions of interest to decide on the remanufacturability of the present core. As seen in Fig. 1, the visualized semantic model encodes different starter motor components by colour. The inspection system proposed in this work consists of a robot-guided acquisition system and a inspection object mounted on a turntable. The acquisition system is capable of capturing colour (I_C) and depth images (I_D) as well as color-coded point clouds. In the case of the acquisition system used in this work (Zivid One+S), a spatial coordinate (point of a point cloud) can be assigned to each pixel in the RGB image, resulting in a one-to-one relationship between pixels in the RGB image and a corresponding point in space. An exemplary inspection process at the inspection station follows the following steps:

1. A starter motor, representing the exemplary product group considered in this work, is clamped onto the turntable.
2. The overall inspection begins, where the entire surface of the starter motor is scanned to detect potential defects. The underlying problem to be solved is the view planning problem (VPP). The solution of a VPP returns the minimal number of viewpoints of the acquisition system to fully inspect the overall surface of an inspection object. This VPP

has been previously addressed by the authors of this work using reinforcement learning, assuming no prior knowledge of the object's geometry [26, 27].

3. The data (RGB and depth images and point clouds) collected during the overall inspection are used to construct the semantic 3D model of the inspection object.
4. This model is then used for the targeted individual inspection, which explicitly inspects specific regions of interest, detected during the overall inspection.

Methodology

This section details the methodology for constructing a semantic 3D model of a core to achieve the vision of an adaptive inspection system proposed in this work. First, the 3D registration procedure using the results from the kinematic modelling of the inspection station to construct a 3D model is presented. Second, the semantic 3D modelling approach that uses segmentation approaches to enhance the 3D reconstruction procedure with semantic information is detailed.

Kinematic modelling of the inspection station and 3D reconstruction procedure

For a 3D reconstruction of the inspection object, point clouds acquired from various poses of the acquisition system and turntable must be registered, requiring a kinematic description of the overall inspection system. The goal is to register all acquired point clouds from various poses of the acquisition system in one global coordinate system.

Figure 2 illustrates the inspection station's kinematics, showing coordinate systems and transformations. Relevant coordinate systems include:

- The static robot base coordinate system C_R.
- The end effector coordinate system C_E.
- The acquisition system coordinate system C_A.
- The static rotary table coordinate system C_{RT}.
- The rotated rotary table coordinate system $C_{RT'}$.

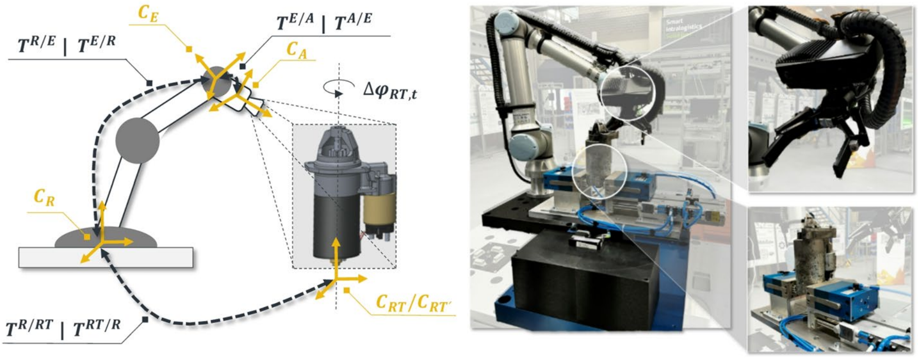


Fig. 2 Simplified depiction of the inspection station and its kinematic relationships (left). Hardware setup of the inspection station (right)

During acquisition, the robot's end-effector is in a specific pose. The acquisition system (coordinate system C_A) is mounted on the end-effector (coordinate system C_E), while the static robot base is denoted as C_R. The turntable has two coordinate systems: C_{RT} (static) and $C_{RT\prime}$ (rotated with an absolute angle ϕ_{RT} in relation to C_{RT}). To enable optimal 3D reconstruction, point clouds are expressed in the same reference system $C_{RT\prime}$ in this work.

The static global coordinate system C_R relates to the end-effector C_E via transformation matrix $T^{R/E}$. The acquisition system's camera (C_A) is linked to C_E via $T^{E/A}$. Additionally, C_R transforms to the rotary table $C_{R/RT}$ using $T^{R/RT}$, while the transformation due to table rotation is given by $T^{RT/RT\prime}$, which includes only rotational components. $T^{R/E}$ is known due to robot kinematics and $T^{E/A}$ as well as $T^{R/RT}$ have been determined using calibration methods (hand-eye calibration).

Point clouds (PC) are transformed between these coordinate systems using the defined matrices. Each acquired point $P_{t,i}^{A}$ in PC_t^A, acquired at time t and expressed in the coordinate system C_A, is first transformed into $C_{RT\prime}$ using $T^{A/RT\prime}$. This transformation yields points $P_{t,i}^{RT\prime}$ of the transformed point cloud $PC_t^{RT\prime}$. The points $P_{t,i}^{RT\prime}$ of $PC_t^{RT\prime}$ retain their colour values while being represented in the rotating coordinate system of the turntable. The transformation is performed based on the following equation using the concatenation of the described transformation matrices:

$$P_{t,i}^{RT\prime} = T^{A/RT\prime} P_{t,i}^{A} = T^{RT/RT\prime} T^{R/RT} T^{A/R} P_{t,i}^{A} \qquad \forall \; P_{t,i}^{A} \in \; PC_t^A \tag{1}$$

Based on these kinematic relationships, a workflow can be defined that allows three-dimensional modelling of the inspection object. This workflow is based on the reconstruction cycle of [28] and is shown in Fig. 3. It is detailed in the following.

- View Generation (1) & Scanning (2): The robot moves, positioning the acquisition system at pose $p_t^{A/R}$, while the rotary table rotates by $\Delta \phi_{RT,t}$. The system captures the clamped object, generating a point cloud PC_t^A in coordinate system C_A.
- Registration (3): The point cloud PC_t^A is transformed into $C_{RT\prime}$ using Eq. (1), yielding $PC_t^{RT\prime}$. The point cloud model $PC_{res,\,t-1}^{RT',t-1}$ from previous acquisitions is also transformed to $C_{RT\prime}$ to account for the rotation $\Delta \phi_{RT,t}$ of the rotary table, resulting in $PC_{res,\,t-1}^{RT'}$. An iterative closest point (ICP) algorithm registers $PC_t^{RT\prime}$ with $PC_{res,\,t-1}^{RT'}$, correcting geometric inaccuracies of robot positioning and 3D acquisition. The transformed clouds are merged to form $PC_{res,\,t}^{RT'}$.
- Integration (4): Voxel downsampling ensures uniform point density and reduces the size of $PC_{res,\,t}^{RT'}$. Filtering functions remove noise for improved accuracy.

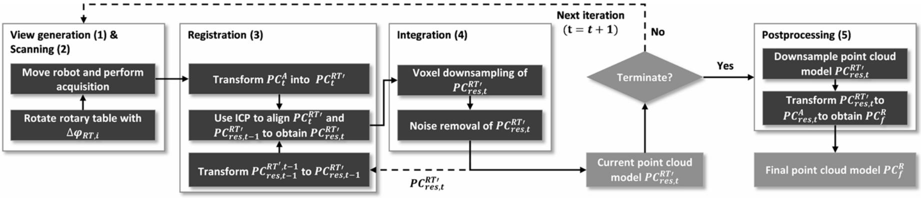


Fig. 3 Flowchart for 3D reconstruction at the inspection station, following the formalization of [28]

- Postprocessing (5): After the acquisition cycle, the resulting point cloud $PC^{RT'}_{res,\,t}$ is transformed into the static robot coordinate system C_R, producing the final 3D model PC^R_f.

Semantic 3D reconstruction using segmentation approaches

The semantic 3D reconstruction approach is obtained by semantically mapping component information onto the surface geometry. Thus, the 3D reconstruction approach is enriched using neural network-based segmentation, which is detailed in the following.

Dataset generation and data preprocessing

To train the deep-learning segmentation approaches evaluated in this work, a suitable dataset is required. The dataset used in this work consists of colour images I_C and depth images I_D, acquired at the inspection station. For training and testing, 42 used starter motors are clamped and scanned. Each motor is captured from 15 different acquisition system poses C_A with varying turntable rotations. The adaptive view planning introduces stochasticity, randomly sampling poses within a permissible range. The acquisition system's positions are sampled on a spherical surface with azimuth angle ϕ_A (45°–90°), polar angle θ_A (45°–90°), and distance r_A (400–600 mm) from the rotary table center C_{RT}. The orientation of C_A is defined by Euler angles α_A, β_A, γ_A, ensuring the acquisition system faces the sphere's centre. To enhance variability, offsets $\Delta\alpha_A$, $\Delta\beta_A$, $\Delta\gamma_A$ ($\pm 10°$) are applied. The turntable is randomly rotated to ϕ_{RT} (0°–360°) before each acquisition, thus capturing the motor from different perspectives and lighting conditions.

According to [29] a starter motor consists of four main components: housing, solenoid, carrier, and gear. For the segmentation in this work, the electrical connection is added as an additional segmentation class. This is because often the electrical connection of a starter is defective and may need individual inspection. Labelling using the annotation tool *labelme* [30] is performed for all acquired images, resulting in the label segmentation mask C_{GT}. C_{GT} has the dimensions H x W x C, where H is the height, W is the width (of the image the label segmentation mask is generated from), and C is the number of classes, including background. Each pixel in the label segmentation mask is one-hot encoded in the third dimension (C) of the label segmentation matrix, indicating its class assignment. The label segmentation mask serves as the ground truth for the segmentation approaches in this work.

Due to the labelling effort the dataset is smaller than standard benchmark datasets. After sorting out unusable data, the dataset used in this work consists of 584 RGB and depth images of the 42 starter motors with the associated label segmentation masks. The starter motors greatly vary in shape and outer characteristics (corrosion, dirt, …). Data augmentation is used to increase dataset size, extract more information, and reduce overfitting [31]. Successful augmentation in small datasets has been demonstrated, e.g., by [32] with the U-Net. In this work, the images are augmented during the training of the model. As a result, new or slightly modified images are continuously passed to the model. This online augmentation allows a significant saving of the required storage space compared to offline augmentation [31]. In this work, all images are augmented with a probability of 50%, so that half of the images remain unchanged. Figure 4 displays raw and augmented colour images, depth images and the augmented colour-coded label segmentation mask of a starter motor.

Augmentation techniques used in this work can be found in [31], including geometric transformations (flipping, rotation, scaling, translation, cropping), colour adjustments (brightness, contrast, saturation), and filtering (noise addition, contrast enhancement with CLAHE, blurring). All transformations are applied to the colour images I_C. Geometric transformations are additionally applied to depth images I_D and masks C_{GT} to prevent a discrepancy between colour images I_C and depth images I_D, as well as segmentation masks C_{GT}.

Segmentation network architectures and learning procedure

Deep neural network-based segmentation can be roughly categorized into two distinct approaches: semantic segmentation and instance segmentation. Semantic segmentation assigns a class label to each pixel, grouping all objects of the same category, while instance segmentation goes a step further by distinguishing individual objects of the same class. Semantic segmentation relies on pixel-wise classification, whereas instance segmentation combines pixel-wise classification with bounding box-based object detection to separate instances. In this work, two widely used models representing these approaches are evaluated and compared.

For semantic segmentation, the U-Net architecture [32] serves as the base model. It enables pixel-wise classification by learning to distinguish different regions within an image. For object detection and instance segmentation, the Mask-RCNN (M-Net) model [33] is employed. This architecture extends object detection by incorporating segmentation capabilities, allowing precise per-instance mask generation for detected objects.

The U-Net model by [32] follows an encoder-decoder architecture, utilizing convolutional and upsampling blocks. The encoder extracts relevant image features while progressively reducing spatial resolution. In contrast, the decoder restores the original resolution using deconvolution layers. A key component of this architecture is the skip connections between the encoder and decoder, which directly transfer fine-grained spatial information. These connections ensure the retention and fusion of both global and local image features, enhancing segmentation accuracy. The M-Net model by [33] extends object detection to instance segmentation by integrating stacked network components. Initially, a convolutional backbone extracts hierarchical features from the input image. A Region Proposal Network (RPN) then identifies potential object locations. The extracted candidate bounding boxes (BBs) are further processed through three dedicated branches. The first branch, BB classification, determines the object category. The second branch, BB regression, refines the

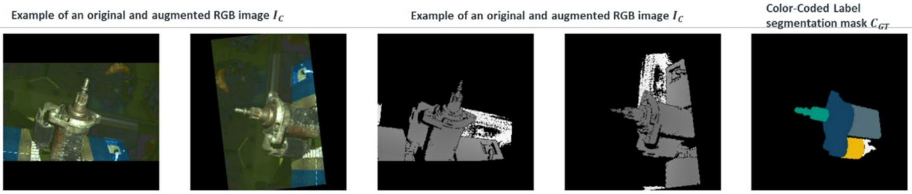


Fig. 4 Visualization of raw and augmented RGB and depth images as well as the color-coded segmentation mask of the augmented images

bounding box coordinates for precise localization. Finally, the segmentation head generates binary per-pixel masks, enabling precise segmentation of each detected object.

Pre-trained backbones VGG16 and ResNet34 trained on ImageNet are utilized for both base model architectures. This transfer learning approach with finetuning on the dataset of this work leverages the pre-learned weights on ImageNet for near-optimal feature extraction at the start of training, reducing training time. The following three training strategies are implemented and compared:

1. Random Initialization: No pre-trained backbone, weights of the whole model are initialized randomly. No finetuning takes place.
2. Frozen Backbone: The backbone's pre-trained weights are initially frozen, and only the downstream layers of the respective segmentation model are trained. After a defined period, the backbone is unfrozen, allowing full model training and finetuning of the backbone.
3. Direct Fine-Tuning: The entire model, including the backbone, is fine-tuned from the start.

Both model alternatives, U-Net and M-Net, normally do not take into account depth data. However, depth data integration may enhance segmentation accuracy. The representatives integrating depth data are called UD-Net (U-Net with depth data integration) and MD-Net (M-Net with depth data integration) in this work and are investigated using different fusion strategies in the implemented neural networks:

- Early Fusion: RGB and depth data are combined before entering the network by stacking the R, G, and B channels with the depth channel.
- Intermediate Fusion: Intermediate features from the backbone (VGG16 or ResNet34) for RGB and depth data are merged before being passed to subsequent layers. Two fusion techniques are considered: element-wise feature addition or feature fusion using an intermediate convolutional layer.
- Late Fusion: In MD-Net, separate feature extraction occurs for RGB and depth data before fusion. In UD-Net, skip connections incorporate late fusion by adding intermediate features or processing them through an additional network before feeding into the decoder.

A visualization of the evaluated fusion approaches for the segmentation network architectures can be found in Fig. 5.

For training, the dataset was split into 80% training data, 10% validation data while training and 10% test data. Validation was conducted between each of the 200 epochs with the validation data being also augmented as defined in the training regimen. Testing was conducted with the test data without augmentation to account for evaluation of the segmentation networks in real use cases.

Two common loss functions for semantic segmentation, Dice Loss and Focal Loss, are evaluated. Dice Loss optimizes class mask overlap by increasing the similarity between predicted and ground truth masks [34]. Focal Loss adjusts pixel-wise loss contributions, reducing the impact of well-classified pixels while emphasizing misclassified ones, making it effective for class imbalance [34]. For the M-Net and MD-Net, the default loss function

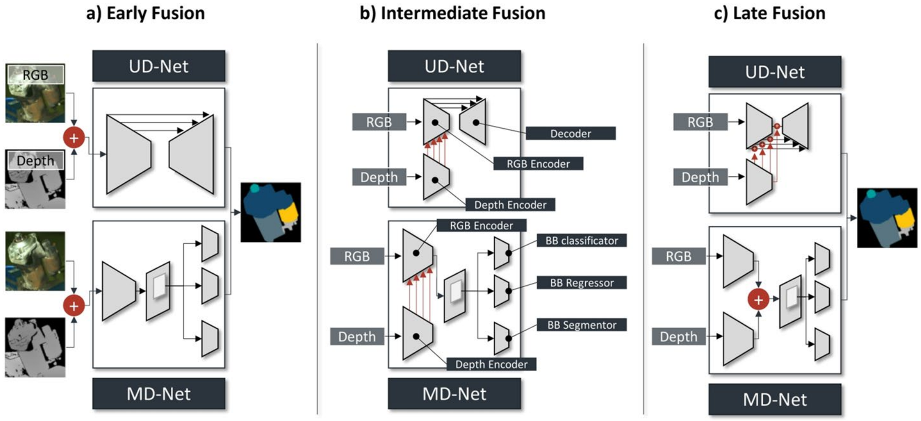


Fig. 5 Overview of the implemented fusion strategies for the UD-Net and MD-Net variants. Early fusion with RGB and depth data results in a segmented image (left). Intermediate fusion with additional description of the subcomponents of the segmentation networks considered (middle). Late fusion (right)

proposed in [33] is used. It consists of a loss combining classification, bounding box regression and mask segmentation accuracy. Segmentation quality evaluation is assessed using Intersection over Union (IoU). The IoU evaluates segmentation mask similarity by computing the overlap ratio between predicted and ground truth masks to their union.

Uncertainty estimation for segmentation results

In the scope of this work, the semantic 3D reconstruction relies on image and depth data acquired from multiple viewpoints to fully reconstruct the semantic 3D model. As a result, identical surface regions of an object are frequently segmented multiple times from different poses. These redundant segmentations can lead to inconsistencies in the assigned class labels due to variations in illumination, occlusion, or perspective depending on viewpoints. To achieve a non-redundant semantic 3D model, it is therefore essential to quantify the uncertainty of each individual segmentation result obtained from a single pose. This enables the system to identify predictions with low confidence and exclude them from the semantic 3D reconstruction process, thereby ensuring the consistency of the final 3D reconstruction. In this work, we address this challenge by applying a GUM-compliant uncertainty estimation strategy to each segmentation output prior to its projection into the semantic 3D space. The GUM defines expanded uncertainty U, computed as.

$U = k\, u_C\left(\widehat{y}\right)$, where $u_C\left(\widehat{y}\right)$ is derived from standard uncertainties of input variables of the measurement function $\widehat{y} = f\left(x_1, \ldots, x_N\right)$ [35].

Segmentation, as a machine learning-based function mapping, aligns with GUM principles, treating neural networks as nonlinear measurement functions. For such nonlinear systems, GUM Supplement 1 [36] introduces a simulation-based method using Monte Carlo sampling. These approaches simulate the probability distributions of input quantities by generating many random samples according to their specified distributions and then computing the corresponding outputs through the measurement model. The resulting output distribution reflects the combined effect of all input uncertainties, including their nonlinearity. In the case of segmentation, pixels of the images (RGB and depth) are treated as input quantities, while the class labels for each pixel are treated as the outputs. The GUM-compliant Monte Carlo approach is then employed to estimate the predictive uncertainty of each pixel-wise segmentation outcome.

A comprehensive review of methods for uncertainty estimation in deep learning is provided by [37], which also gives insights into how the simulation of the probability distributions of the pixels as input quantities can be achieved. Deterministic methods, bayesian methods, ensemble methods, and test-time augmentation methods are distinguished. All these approaches estimate predictive uncertainty $\widehat{\sigma}$, combining model (systemic) and data (aleatory) uncertainty. In this work, the GUM-compliant combination of an ensemble-based Monte-Carlo approach together with a test-time augmentation approach is chosen to estimate the predictive uncertainty $\widehat{\sigma}$ of a segmentation.

Figure 6 illustrates the proposed method. Given the training time constraints, $n = 10$ (M_1, …, M_n) are trained using varied training dataset compositions resulting in a segmentation model ensemble. The ensemble prediction is obtained by averaging the prediction of these models for a dataset tuple $I_{C,T}$ and $I_{D,T}$ to derive the pixel-wise segmentation result $\widehat{y}_{uv,T}$. The predictive uncertainty $\widehat{\sigma}_{uv,T}$ is obtained by applying random augmentations to the dataset tuple $I_{C,T}$ and $I_{D,T}$, where the random augmentations aim to simulate the probability distributions of the pixels as input quantities. Each augmented sample is passed through all trained models and the pixel-wise standard deviation $\widehat{\sigma}_{uv,T}$ across predictions determines the final uncertainty measure.

Semantic 3D reconstruction procedure using segmentation approaches

To achieve semantic 3D modelling, the 3D reconstruction approach is extended by incorporating segmentation before registration (step 3). After view generation and acquisition (steps 1 & 2), segmentation of the acquired point cloud PC_t^A is performed using the color image

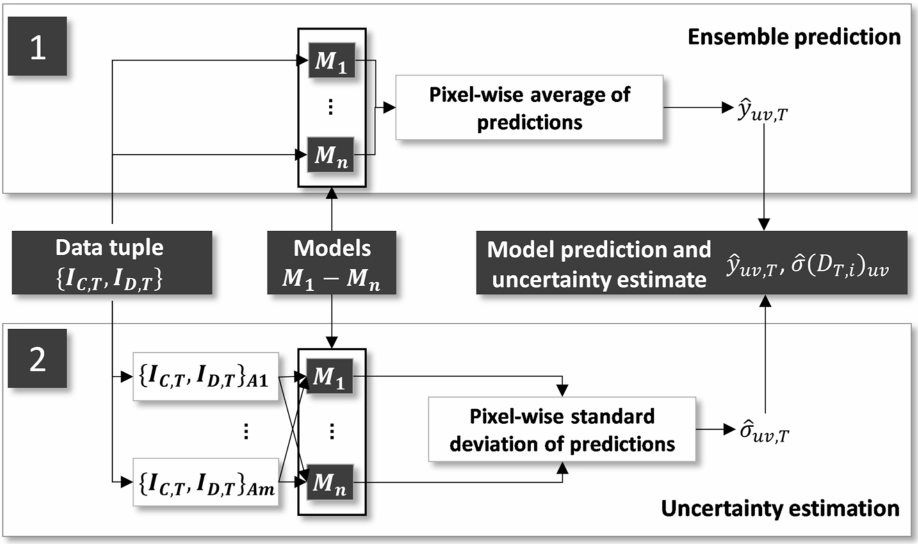


Fig. 6 Approach for model prediction and predictive uncertainty estimation using ensembles and test-time augmentation

I_C and depth image I_D. The acquisition system ensures a unique mapping between image pixels and spatial coordinates, generating a semantic point cloud (SPC_t^A). This extends PC_t^A by assigning each point a class label K_t based on segmentation results:

$$SPC_t^A = [X_t^A,\ Y_t^A, Z_t^A, R_t,\ G_t,\ B_t,\ K_t] \tag{2}$$

Multiple segmentation models can be integrated for different tasks, such as component and defect segmentation. If, for example, defect segmentation would be included, an additional column D_t would encode surface defects:

$$SPC_t^A = [X_t^A,\ Y_t^A, Z_t^A, R_t,\ G_t,\ B_t,\ K_t,\ D_t] \tag{3}$$

The predictive uncertainty approach must be applied separately to each segmentation model. Pixels with high uncertainty (e.g. $\widehat{\sigma}_{uv,T} > 0.5$) can be excluded from reconstruction to improve accuracy. The semantic 3D modelling procedure can then be formalized as seen in Fig. 7.

After generating the semantic 3D point cloud model, a semantic mesh can optionally be created in a post-processing step by meshing the point cloud and assigning the semantic labels to the respective nearest mesh vertices. This enables further processing or visualization, while preserving the advantages of the initial point cloud representation during the modelling pipeline.

Results

The results are structured along the methodology. First, the quality of the 3D reconstruction process is assessed. Second, the semantic and instance segmentation results with consideration of the depth data are examined. Last, the semantic 3D reconstruction process is evaluated, followed by a short outlook on defect segmentation with the use case of corrosion segmentation. All experiments are conducted on a Linux system with an AMD Ryzen Threadripper PRO 3955WX with 16 cores running at 3.9 GHz and NVIDIA RTX A6000 graphics.

Quality of the 3D reconstruction process

For evaluation of the quality of the 3D reconstruction process, four 3D product models (C) of starter motors generated at the inspection station are compared to highly accurate refer-

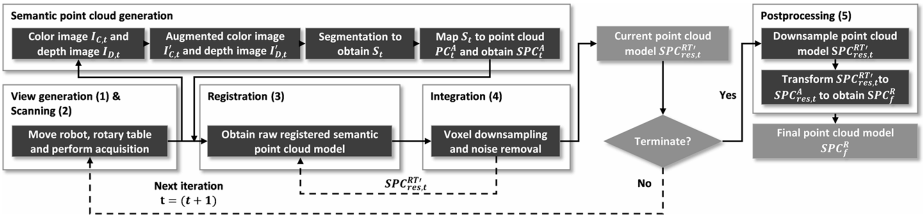
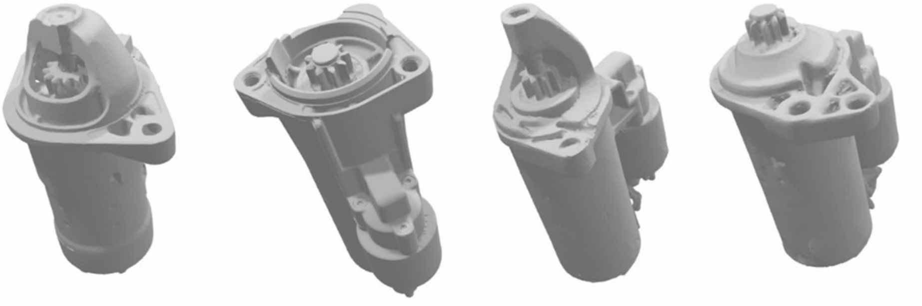


Fig. 7 Flowchart for the semantic 3D modelling procedure at the inspection station



Starter Motor Model #1 Starter Motor Model #2 Starter Motor Model #3 Starter Motor Model #4

Fig. 8 Visualization of the reference product models (L) of the starter motors being used for evaluating the reconstruction approach

Table 1 Overview of the metrics used to compare the product models C and L

	$\mathrm{HD}_{95}^{\mathrm{C}\rightarrow\mathrm{L}}$[µm]	$\mathrm{HD}_{95}^{\mathrm{L}\rightarrow\mathrm{C}}$[µm]	$\mathrm{ME}^{\mathrm{C}\rightarrow\mathrm{L}}$[µm]	$\mathrm{MCE}^{\mathrm{C}\rightarrow\mathrm{L}}$	CD
	min/max mean	min/max mean	min/max mean	min/max mean	min/max mean
3D model # 1	623.2/629.9 627.9	2004/2008 2007	252.4/256.1 254.1	0.3263/0.3271 0.32666	0.7208/0.7257 0.7231
3D model # 2	701.6/747.2 723.1	8977/9003 8990	303.1/319.8 311.1	0.2612/0.2714 0.2657	1.669/1.691 1.679
3D model # 3	645.7/648.0 647.0	5322/5335 5329	282.9/287.6 285.3	0.1842/0.1859 0.1851	1.163/1.167 1.165
3D model # 4	414.1/424.6 420.2	4673/4677 4675	157.4/158.0 157.5	0.087/0.088 0.088	0.8235/0.8242 0.8238

ence product models (L) of the same products generated by a hand-held laser scanner (Zeiss T-Scan) as seen in Fig. 8. Only four models are chosen, since the reconstruction approach is purely deterministic.

Table 1 summarizes the results with metrics (Hausdorff distance, mean error, mean squared error and chamfer distance) used to compare sets of points. Before comparison, C as well as L are registered 10 times using a global (RANSAC) followed by a local (ICP) approach. All these registration results are then used to calculate the minimum (min), maximum (max), and mean metric values. Variations in metric performance stem from product geometry, colour and reflectance properties of the product surface as well as point cloud registration. For example, model two exhibits a 5.5% variation in $ME^{C\rightarrow L}$ across the 10 registrations, highlighting the importance of accurate registration.

The Hausdorff distance $HD_{95}^{C\rightarrow L}$ from the product model (C) generated by the robot-mounted acquisition system to the reference model (L) remains below 750 µm for all models. This implies that 95% of points in C deviate by at most this value. In contrast, $HD_{95}^{L\rightarrow C}$ reaches nearly 9 mm for one model, suggesting poor accuracy. However, this discrepancy stems from missing data rather than acquisition system inaccuracy. The laser-scanned model is acquired continuously by moving the laser scanner by hand, while the acquisition system model relies on only a few viewpoints, which leaves some areas uncaptured. Fortunately, core features like drillings are captured with sufficient accuracy in the product model C (see Fig. 9).

The mean error $ME^{C\rightarrow L}$ is below 320 μm, indicating high accuracy given that each model consists of over two million points. Model four is particularly precise, with an average error of 157.5 μm. The mean squared error $MSE^{C\rightarrow L}$ ranges from 0.088 to 0.3266, with model four showing minimal deviation (0.088). The Chamfer Distance (CD) exceeds $MSE^{C\rightarrow L}$ for all models, as missing areas in the camera model cause large point-to-point distances. Increasing the number of viewpoints would reduce this error but at the cost of longer inspections. Overall, the reconstructed 3D model using the approach proposed in this work reconstructs the geometry of the starter motors considered in this work with sufficient detail.

Results of the semantic and instance segmentation approaches

The evaluation of the segmentation approaches developed in this work is based on the IoU metric, as previously introduced. The segmentation models were trained and tested using a validation dataset. Three different neural networks (NNs) were trained for each model configuration. The following sections present the model configurations and the results achieved.

Segmentation models using only RGB image data

The first set of experiments focused on training segmentation networks using only RGB data. Table 2 gives an overview of the network and learning parameters considered in this work. The training process is performed using different hyperparameters in a full factorial combinatorial way for the U-Nets and M-Nets, including backbone architecture, optimizers, loss functions, and learning rates.

Table 3 presents the performance of the best-trained U-Net and M-Net configurations. Due to the large number of different trained networks with different hyperparameters, only an extract of the results is shown in Table 3, but the complete results of the investigations are presented in the text.

In general, the experiments show that the M-Net variants outperform the U-Net variants when being evaluated on the validation data (with data augmentation). The best U-Net con-

Fig. 9 3D model reconstructed using the Zivid One+ S (C) on the left and the 3D model reconstructed using the Zeiss T-Scan (L) in the middle. The visualization of the deviation analysis between both models can be seen on the right

Table 2 Overview of the network and learning parameters considered in this work for U-Net and M-Net

Parameter	**Value**		**Value**	
Model	U-Net		M-Net	
Backbone	VGG16	ResNet34	VGG16	ResNet34
Optimizer	SGD	Adam	SGD	Adam
Learning rate[a]	0.01/0.005/0.001		0.01/0.005/0.001	
Loss	Dice	Focal	Default	
Pretrain	True	False	True	False
Freeze[b]	True	False	True	False
Batch size	8		8	

[a] Initial learning rate, reduced by 20% every 10 epochs. Learning rate for Adam is 0.01 times the learning rate of SGD

[b] Freezing in the initial epochs of the training takes place (yes or no). Training is performed for 200 epochs and if freezing is enabled, the network weights of the encoder are frozen for 50 epochs

Table 3 Performance of U-Net and M-Net configurations trained in this work

Model	Backbone	Optimizer	Loss	Learning rate	Pretrain	Freeze	Mean IoU
$\mathrm{U}_{\mathrm{pot}}$	VGG16	Adam	Dice	0.0001	True	False	0.9187
U_{C1}	ResNet34	Adam	Dice	0.0001	True	False	0.9005
U_{C2}	VGG16	SGD	Dice	0.01	True	False	0.6449
$\mathrm{M}_{\mathrm{opt}}$	ResNet34	Adam	Default	0.0001	True	True	0.9537
M_{C2}	VGG16	Adam	Default	0.0001	True	True	0.9462
M_{C2}	ResNet34	SGD	Default	0.01	True	True	0.9488

figuration U_{opt} used a VGG16 backbone, Adam optimizer, Dice loss function, and a learning rate of 0.0001. In all cases, U-Net variants trained with ResNet34 Encoder performed worse than their counterparts using the VGG16 encoder, as shown by the comparison of network U_{opt} with U_{C1}. This is not the case when using M-Net variants. Here, the ResNet34 encoder also proved to be a solid variant in addition to VGG16 (see M_{opt} and M_{C1}). In both cases, the Adam optimizer shows better results in the segmentation task (compare U_{opt} with U_{C2} and M_{opt} with M_{C2}). Furthermore, using pre-trained encoder weights has proven successful for both network variants, while no freezing was beneficial for the U-Net variants and freezing is beneficial for the M-Net variants.

Segmentation models integrating depth information and uncertainty filtering

To enhance segmentation performance, additional models are trained with depth data integration, resulting in UD-Net (U-Net with depth) and MD-Net (M-Net with depth). Training configurations are adapted based on previous findings. For instance, SGD and ResNet34 are excluded from U-Net training, while VGG16 is omitted from M-Net configurations. The configurations of the conducted experiments can be found in Table 4.

The results of these experiments are shown in Table 5. The best-performing UD-Net model ($UD_{opt,v}$) on the validation data used an intermediate feature fusion with addition, achieving a mean IoU of 0.9291. In contrast, the best MD-Net model ($MD_{opt,\,v}$) used early fusion with ResNet34, reaching a mean IoU of 0.9527. As in the case of only training with RGB image data, the MD-Net variant outperforms the UD-Net variant. A direct comparison of the IoU values of M-Net and MD-Net as well as U-Net and UD-Net variants reveals that

Table 4 Overview of the network and learning parameters considered in this work for UD-Net and MD-Net

Parameter	**Value**				**Value**			
Model	UD-Net				MD-Net			
Fusion type	Early	Intermediate		Late	Early	Intermediate		Late
Fusion variant[al]	Add	Add	Net	Add	Add	Net	Net	Add
Backbone	VGG16				ResNet34			
Optimizer	Adam				SGD		Adam	
Learning rate[b]	0.0001 & 0.00005				0.01	0.005	0.0001	0.00005
Loss	Dice		Focal		Default			
Pretrain	True		False		True		False	
Freeze[c]	True		False		True		False	
Batch size	8				8			

a: Add denotes feature addition, while Net denotes feature fusion by a convolutional layerb: Initial learning rate, reduced by 20% every 10 epochsc: Freezing in the initial epochs of the training takes place (yes or no). Training is performed for 200 epochs and if freezing is enabled, the network weights of the encoder are frozen for 50 epochs

Table 5 Performance of UD-Net and MD-Net configurations trained in this work

Model	Fusion type	Fusion variant	Backbone	Optimizer	Loss	Learning rate	Pretrain	Freeze	Mean IoU
$\mathrm{UD}_{\mathrm{opt,v}}$	Intermediate	Add	VGG16	Adam	Focal	0.0001	True	True	0.9291
$\mathrm{MD}_{\mathrm{opt,v}}$	Early	Add	ResNet34	Adam	Default	0.00005	True	True	0.9527
$\mathrm{UD}_{\mathrm{opt,t}}$	Intermediate	Add	VGG16	Adam	Focal	0.0001	True	True	0.9662
$\mathrm{MD}_{\mathrm{opt,t}}$	Early	Add	ResNet34	Adam	Default	0.00005	True	True	0.9464
$\mathrm{UD}_{\mathrm{opt,E}}$	Intermediate	Add	VGG16	Adam	Focal	0.0001	True	True	0.9707
$\mathrm{MD}_{\mathrm{opt,E}}$	Early	Add	ResNet34	Adam	Default	0.00005	True	True	0.9574
$\mathrm{UD}_{\mathrm{opt,EF}}$	Intermediate	Add	VGG16	Adam	Focal	0.0001	True	True	0.9529
$\mathrm{MD}_{\mathrm{opt,EF}}$	Early	Add	ResNet34	Adam	Default	0.00005	True	True	0.9359

integrating depth information improved segmentation accuracy. The UD-Net and MD-Net models exhibited more stable training behaviour and higher accuracy, particularly in distinguishing foreground objects from the background.

Further experiments revealed, that when evaluating $UD_{opt,v}$ and $MD_{opt,\ v}$ on test data containing non augmented images from starter motors not seen during training (resulting in $UD_{opt,t}$ and $MD_{opt,\ t}$ it could be noted that the UD-Net even outperformed the MD-Net). A possible explanation is that data augmentation introduces significant noise into the NN, which affects the robustness of semantic segmentation more than that of instance segmentation. While semantic segmentation directly classifies each pixel, instance segmentation follows a two-step process: it first detects objects and then applies binary segmentation to the identified regions. This additional object detection step likely improves robustness by providing a more structured approach to segmentation. Further experiments showed that an ensemble learning approach (resulting in $UD_{opt,E}$ and $MD_{opt,E}$) improved segmentation quality. Filtering out pixels with high predictive uncertainty resulted in lower IoU values ($UD_{opt,EF}$ and $MD_{opt,EF}$). However, by filtering out pixels with high predictive uncertainty, the false detection rate (FDR) of falsely segmented pixels could also be lowered (FDR from 0.0770 for $UD_{opt,E}$ to 0.0315 for $UD_{opt,EF}$ and 0.0759 for $MD_{opt,E}$ to 0.0248 for $MD_{opt,EF}$) indicating a more robust segmentation.

Evaluation of the semantic 3D reconstruction process

It has been shown that the segmentation approaches are capable of segmenting the individual starter motor components with sufficient accuracy. For the evaluation of the semantic 3D reconstruction approach, the ten additional starter motors whose acquired images were used for testing the segmentation models (results from $UD_{opt,t}$ and $MD_{opt,t}$) were semantically reconstructed.

For evaluation of the semantic 3D modelling approach, the reconstructed 3D models were labelled using the labelling *tool semantic-segmentation-editor* [38] (resulting in a ground truth semantic 3D model) and compared with the semantic 3D models by using the segmentation approaches presented in this work. As metrics, the segmentation accuracy (how many surface points were segmented correctly) and the completeness (how many surface points are retained after segmentation and filtering of pixels/surface points with high predictive uncertainty) were used. The results in Table 6 show that $UD_{opt,EF}$ achieved the highest model accuracy, with an average accuracy of 93.9% across all components. Regarding model completeness, $UD_{opt,EF}$ also outperformed $MD_{opt,EF}$.

Outlook on the integration of defect segmentation approaches

The results so far have demonstrated the applicability of the method presented in this work for the semantic 3D reconstruction of an inspection object. Using neural networks for component segmentation, a semantic 3D model can be constructed. In addition to component segmentation, the methodical approach presented in this work also discusses an extension of the semantic 3D model with information about defects on the inspection object. Given training with a suitable dataset, the models proposed in this work can also be used to segment defects in an image.

Initial results with the model $U_{opt,v}$ show promising results in the segmentation of corrosion on the individual components of the starter motor. In this case, the use of a segmentation model with integrated depth data is avoided because, in the opinion of the authors, it does not provide any additional information regarding the segmentation of corrosion. This is because corrosion exhibits a rather small geometric change from the corrosion-free geometric state of the starter motors, and the geometric change of a surface due to corrosion is smaller than the accuracy of the Zivid One+S. The results of the segmentation in this pioneering study can be seen in Fig. 10. The neural segmentation model $U_{opt,v}$ trained achieved an average IoU of 0.9634 resulting in a comparable segmentation quality as in the component segmentation use case. In future work, the information provided by the semantic 3D model, such as the affiliation of surfaces to specific components and defects occurring on these surfaces, can be used to draw conclusions about remanufacturability of the product.

Table 6 Results of the evaluation of accuracy and completeness of the semantic 3D models

Model	Carrier	Housing	Solenoid	Electric	Gear	Mean
Accuracy						
$\mathrm{UD}_{\mathrm{opt,EF}}$	96.1%	94.4%	93.1%	90.7%	95.3%	93.9%
$\mathrm{MD}_{\mathrm{opt,EF}}$	94.0%	91.6%	93.5%	84.4%	94.9%	91.7%
Completeness						
$\mathrm{UD}_{\mathrm{opt,EF}}$	93.3%	97.9%	99.0%	94.2%	96.9%	96.2%
$\mathrm{MD}_{\mathrm{opt,EF}}$	93.1%	97.2%	98.0%	91.6%	92.4%	94.4%

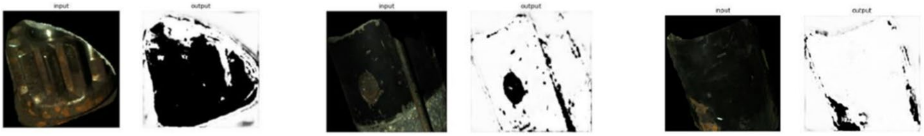


Fig. 10 Overview of binary segmentation results for the detection of corrosive pixels in an image

Conclusions

Remanufacturing is a key process to achieve a circular economy, restoring used products and returning them to the market. However, the process still relies heavily on manual labour, changing its cost-efficiency and scalability. One example is the visual inspection of products before remanufacturing. Automating this step is challenging due to unknown product variants, a lack of CAD models, and defects appearing in different forms and locations.

To address these challenges, this paper presents a semantic 3D product modelling approach that integrates image and 3D point cloud data. Using neural networks (U-Net for semantic segmentation and Mask-RCNN for instance segmentation), both product components and defects, such as corrosion, can be detected and encoded into a semantic 3D model. This model serves as a basis for adaptive inspection strategies.

Experimental results show that the proposed method achieves high accuracy in component segmentation and in embedding semantic information into the geometric product structure. Furthermore, it is shown that the U-Net architecture is not only suitable for component segmentation but also capable of detecting defects, such as corrosion. Finally, it has been shown that the combination of RGB and depth data significantly improves segmentation accuracy, while uncertainty estimation improves robustness.

In summary, the approach shows great potential for the inspection of starter motors. In principle, the present approach of semantic 3D modelling is suitable for a wide range of object geometries of different sizes. However, it must be noted that the acquisition of undercuts must be accompanied by a meaningful pose planning (see visual planning problem VPP) so that a modelling of these object areas is possible. Likewise, for larger objects, it is worth noting that acquisition systems have a specific working range. For the inspection system used here, more acquisitions would be necessary for semantic 3D modelling to capture the entire product. This would result in longer inspection times and inspection costs. General limitations for semantic 3D modelling also exist in that some surface properties of inspection objects (e.g., highly reflective surfaces) make it difficult for optical systems to capture them. Such difficulties can be mitigated by meaningful pose planning (see visual planning problem VPP) or parameter selection of the acquisition system, although this was not part of this work.

Furthermore, it must be noted that the applicability of the presented segmentation algorithms to other object types depends on the availability of adequate training data. The neural networks require training on images of the respective object types to perform accurate segmentation. In this work, the segmentation algorithms are only trained on image and depth data of starter motors. Consequently, generalization of the semantic 3D reconstruction approach is only feasible for varying types of starter motors since they share sufficient visual and geometric similarity with those used for training.

Future work is recommended to focus on the limitations mentioned above and extend the method to acquire additional defect types, integrate improved uncertainty quantification, and optimize the system for real-time industrial applications. In the current system implementation, each starter motor is acquired from five different viewpoints. Each acquisition takes approximately one second using the robot-guided Zivid One + S system. Leveraging the ROS framework [39], subsequent processing steps (semantic segmentation (1 s per image) and point cloud registration) are executed in parallel with robot and rotary table movements or additional acquisitions. This enables an efficient processing pipeline, resulting in a total acquisition and processing time of approximately 15 to 20 s per starter motor. While these runtimes are promising for a proof-of-concept implementation, future work will focus on further runtime. In addition, the application of the approach to different product classes will be investigated to evaluate its generalisability.

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1574–471687386. The authors thank the Karlsruhe House of Young Scientists (KHYS) for funding the research travel to Belgium, where the research was conducted.

Author contributions Jan-Philipp Kaiser: Conceptualization, Methodology, Validation, Writing – Original DraftDominik Koch: Methodology, Software, Writing - Review & EditingFlorian Stamer: Conceptualization, Supervision, Writing - Review & EditingJef Peeters: Conceptualization, Supervision, Writing - Review & EditingGisela Lanza: Conceptualization, Supervision, Writing - Review & Editing.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Daniel V, Guide R (1997) Scheduling with priority dispatching rules and drum-buffer-rope in a recoverable manufacturing system. Int J Prod Econ 53(1):101–116
2. Tolio T, Bernard A, Colledani M, Kara S, Seliger G, Duflou J, Battaia O, Takata S (2017) Design, management and control of demanufacturing and remanufacturing systems. CIRP Ann 66:585–609
3. Errington M, Childe SJ (2013) A business process model of inspection in remanufacturing. J Remanufact 3(1):1–22
4. Kurilova-Palisaitiene J, Sundin E, Poksinska B (2018) Remanufacturing challenges and possible lean improvements. J Clean Prod 172:3225–3236
5. WABCO Reman Solutions (2025), May 30 Core acceptance criteria. https://www.rb-automotive.de/fileadmin/data/RundB/R_B/Herstellerunterlagen/WABCO_E_APU_Acceptance_criteria.pdf

6. CoremanNet (2025), May 30 Return Criteria. https://www.coremannet.com/assets/docs/return-criteria/new-2019/Starter.pdf
7. He Y, Hao C, Li Y, Lim MK, Wang Y (2020) A failure feature identification method for adaptive remanufacturing. Procedia CIRP 90:552–556
8. Wu Y, Zhou C, Sterkens W, Piessens M, De Marelle D, Dewulf W, Peeters J (2024) A color and depth image defect segmentation framework for EEE reuse and recycling. IEEE 2024 Electronics Goes Green 2024+(EGG) pp. 1–7
9. Siddiqi MU, Ijomah WL, Dobie GI, Hafeez M, Pierce G, Ion S, Mineo W, MacLeod C (2019) Low cost three-dimensional virtual model construction for remanufacturing industry. J Remanuf 9:129–139
10. Khan A, Mineo C, Dobie G, Macleod C, Pierce G (2021) Vision guided robotic inspection for parts in manufacturing and remanufacturing industry. J Remanufact 11(1):49–70
11. Nüchter A, Hertzberg J (2008) Towards semantic maps for mobile robots. Robot Auton Syst 56(11):915–926
12. Kostavelis I, Gasteratos A Semantic mapping for mobile robotics tasks: A survey. Robot Auton Syst (2015), 66, 86–103
13. Martín F, González F, Guerrero JM, Fernández M, Gines J (2021) Semantic 3d mapping from deep image segmentation. Appl Sci 11(4):1953
14. Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J (2017) A review on deep learning techniques applied to semantic segmentation. ArXiv Preprint arXiv :170406857
15. Bernardini F, Rushmeier H (2002) The 3D model acquisition pipeline. Comput Graph Forum 21(2):149–172
16. Phang JSP, Lim KH, Chiong RCW (2021) A review of three dimensional reconstruction techniques. Multimedia Tools Appl 80:17879–17891
17. Rickli JL, Dasgupta AK, Dinda GP (2014) A descriptive framework for additive remanufacturing systems. Int J Rapid Manuf 4(2–4):199–218
18. Wang W, Ma Z, Wu X, Wu D (2010) Damaged region positioning for flexible remanufacturing using structured light. Proceedings of the 2010 International IEEE Conference on Modelling, Identification and Control pp. 304–309
19. Zhang X, Cui W, Liou F (2021) Voxel-based geometry reconstruction for repairing and remanufacturing of metallic components via additive manufacturing. Int J Precision Eng Manufacturing-Green Technol pp. 1–24
20. Li L, Li C, Tang Y, Du Y (2017) An integrated approach of reverse engineering aided remanufacturing process for worn components. Robotics and Computer-Integrated Manufacturing 48:39–50
21. Yin Z, Yuan X, Zhang G, Zhao H (2012) Research on structured light 3D vision in the remanufacturing system based on robotic arc welding. Proceedings of the 10th IEEE World Congress on Intelligent Control and Automation pp. 4527–4531
22. Zheng Y, Liu J, Liu Z, Wang T, Ahmad R (2019) A primitive-based 3D reconstruction method for remanufacturing. Int J Adv Manuf Technol 103:3667–3681
23. Jing O, Lai Z, Qinghong W, Xin L, Yingjie L (2021) Weld-seam identification and model reconstruction of remanufacturing blade based on three-dimensional vision. Adv Eng Inform 49:101300
24. Zeiss (2025), May 30 Zeiss ScanBot. https://www.zeiss.de/messtechnik/systeme/optische-3d-messtechnik/automatisierte-loesungen/scancobot.html
25. Keyence Keyence (2025), May 30 VL-700. https://www.keyence.de/products/3d-measure/3d-scanner/vl-700/
26. Kaiser JP, Gäbele J, Koch D, Schmid J, Stamer F, Lanza G (2024) Adaptive acquisition planning for visual inspection in remanufacturing using reinforcement learning. J Intell Manuf. https://doi.org/10.1007/s10845-024-02478-0
27. Kaiser JP, Koch D, Gäbele J, May MC, Lanza G (2024) View planning in the visual inspection for remanufacturing using supervised- and reinforcement learning approaches. CIRP J Manuf Sci Technol 53:128–138
28. Scott WR, Roth G, Rivest JF (2003) View planning for automated three-dimensional object reconstruction and inspection. ACM Comput Surv 35(1):64–96
29. Füvesi V, Kovács E, Blága C (2010) Measurement and identification of a starter motor system. Proceedings of the 2nd international conference on recent chievements in mechatronics, automation, computer science and robotics, Macro pp. 129–134
30. Labelme (2025) May 30). https://github.com/wkentaro/labelme
31. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6(1):1–48
32. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention proceedings part III 18, pp. 234–241

33. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. Proceedings of the IEEE international conference on computer vision pp. 2961–2969
34. Jadon S (2020) A survey of loss functions for semantic segmentation. IEEE Conf Comput Intell Bioinf Comput Biology (CIBCB) 2020:1–7
35. JCGM 100:2008 Evaluation of measurement data – Guide to the expression of uncertainty in measurement
36. JCGM 101:2008 Evaluation of measurement data – Supplement 1 to the guide to the expression of uncertainty in measurement – Propagation of distributions using a Monte Carlo method
37. Gawlikowski J, Tassi CRN, Ali M, Lee J, Humt M, Feng J, Kruspe A, Triebel R, Jung P, Roscher R, Shahzad M, Yang W, Bamler R, Zhu XX (2023) A survey of uncertainty in deep neural networks. Artif Intell Rev 56(Suppl 1):1513–1589
38. Hitachi Automotive and Industry Lab (2025), May 30 https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor
39. Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A (2009) ROS: an open-source robot operating system. ICRA Workshop Open Source Softw 3(32):5

Authors and Affiliations

Jan-Philipp Kaiser[1] · Dominik Koch[1] · Florian Stamer[1] · Jef Peeters[2,3] · Gisela Lanza[1]

✉ Jan-Philipp Kaiser
jan-philipp.kaiser@kit.edu

Dominik Koch
dominik.koch@kit.edu

Florian Stamer
florian.stamer@kit.edu

Jef Peeters
jef.peeters@kuleuven.be

Gisela Lanza
gisela.lanza@kit.edu

[1] Institute of Production Science (wbk), Karlsruhe Institute of Technology (KIT), KarlsruheKaiserstr. 12, 76131, Germany

[2] Department of Mechanical Engineering, KU Leuven, Heverleecelestijnenlaan 300, 3001, Belgium

[3] Flanders make @ KULeuven, Heverleecelestijnenlaan 300, 3001, Belgium