Journal of Intelligent Manufacturing (2023) 34:303–314
https://doi.org/10.1007/s10845-022-02036-6

Automated 3D burr detection in cast manufacturing using sparse convolutional neural networks

Ahmed Mohammed[1] · Johannes Kvam[1] · Ingrid Fjordheim Onstein[2] · Marianne Bakken[1] · Helene Schulerud[1]

Received: 22 December 2021 / Accepted: 23 September 2022 / Published online: 11 October 2022
© The Author(s) 2022

Abstract
For automating deburring of cast parts, this paper proposes a general method for estimating burr height using 3D vision sensor that is robust to missing data in the scans and sensor noise. Specifically, we present a novel data-driven method that learns features that can be used to align clean CAD models from a workpiece database to the noisy and incomplete geometry of a RGBD scan. Using the learned features with Random sample consensus (RANSAC) for CAD to scan registration, learned features improve registration result as compared to traditional approaches by (translation error (Δ18.47 mm) and rotation error(Δ43°)) and accuracy(35%) respectively. Furthermore, a 3D-vision based automatic burr detection and height estimation technique is presented. The estimated burr heights were verified and compared with measurements from a high resolution industrial CT scanning machine. Together with registration, our burr height estimation approach is able to estimate burr height similar to high resolution CT scans with Z-statistic value ($z = 0.279$).

Keywords Burr detection · Burr height · Deep learning · Convolutional neural network · Registration

Introduction

When sand casted parts come out of the mold they have to be cleaned in a fettling process to remove sprues, runners, risers, and flashing. Flashing, also called burrs, is the material left on the cast part in the separation plane between the two molds. Figure 1 shows a cast part with risers and flashing. Fettling of cast parts is important to ensure that the part meets its design requirements. Removal of sprues, risers, and runners can be done in a cutting process that leaves a burr to be removed. The burrs from the cutting and flashing are removed in a deburring process. Deburring can be very challenging and are mostly done manually (Aertbeliën, 2009) where the workers are exposed to high noise and vibration levels. It is therefore desirable to automate the deburring process.

Detecting and measuring the size of burrs to adjust the feed rate and registration of CAD model with the workpiece-scan for creating a deburring tool path are some of the essential steps in automated deburring process. A deburring process is traditionally planned on either a reference CAD model of reference workpiece (Onstein et al., 2020), this includes both the tool path and machining parameters. The challenge with automating deburring of cast parts is that both the workpiece and the burrs vary in shape and size due to the solidifying in the casting process. During solidifying, uneven shrinkage and deformation can occur. The uneven shrinkage and deformation can be controlled by optimizing the casting process, but it cannot be completely avoided (Huang et al., 2021). To formalize this observation, we define the domain gap as a measure of the discrepancy between the CAD (source) and target (scan) domains. This means that the deburring process has to be robust to domain gap, i.e. corresponding CAD and scan geometric variations.

The conventional approach to burr localization (Huang et al., 2021) follows a two-step coarse and fine point cloud registration approach where the relative positions of workpieces and CAD models must be accurate. However, such approach are sensitive to variation in point density, missing data (partial scans) and noise in the scan. Furthermore, registration of the workpieces and CAD models fails due to domain gap issues.

✉ Ahmed Mohammed
ahmed.mohammed@sintef.no

1 Smart Sensor Systems, SINTEF Digital, Oslo, Norway

2 Department of Manufacturing and Civil Engineering, Norwegian University of Science and Technology, Gjøvik, Norway

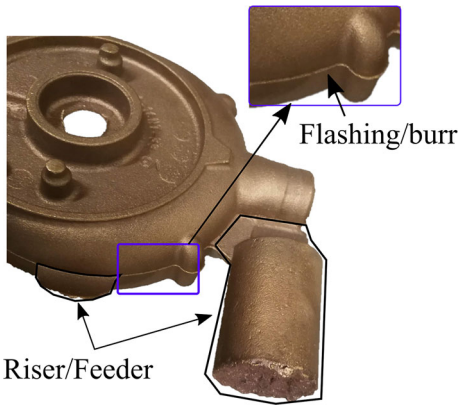


Fig. 1 Cast part with riser/feeder and flashing

In this paper, we propose a machine learning approach that is robust to missing data in the scans, view point, noise, rotation and scale. Our approach is able to learn features of the CAD model/scan without any labeled data through a series of augmentations to the CAD models. Note that the our approach uses only CAD models for training and is able to generalize to scans during testing. Furthermore, using the learned features we show a burr localization approach that is able to estimate burr height similar to high resolution CT scans with Z-statistic value ($z = 0.279$) supporting the hypothesis that the CT measured and our estimate burr height distributions are similar. This paper is structured as follows. "Background" section briefly reviews previous works on deburring and burr measurement. In "Methods" section we present the proposed burr detection and quantization approach, and the data collection setup. In "Results" section we present evaluation of the CAD-scan registration and burr size estimation. Finally, in "Discussion and conclusion" section we conclude the paper with discussion and future work.

Background

Scan-to-CAD registration for tool path planning

The deburring process consists of, among other steps, detecting and quantifying the burrs, generating a deburring tool path and optimising machining parameters. Because of the geometric variations caused by deformations and uneven shrinkage during solidifying, the deburring process has to be adapted to each individual workpiece. The deburring tool path is traditionally generated off-line based on the CAD model using a CAM software. This reference path has to be registered onto the workpiece, where the goal is to find the corresponding relationship between two point sets and compute an appropriate transformation (Du et al., 2015). The most widely used method for registration of 3D shapes is the iterative closest point (ICP) algorithm (Besl & McKay, 1992).

Kosler et al. (2016) propose a method using ICP to register a 3D scan with the CAD model to adapt a reference tool path, generated using off-line teaching, to the workpiece. Song and Song (2013) propose a method that generates a tool path using CAM software and then correct the path based on teaching points using an ICP-based contour matching algorithm. These two ICP-based methods assume that the deformations caused by the solidification can be ignored because it is not significant for the overall geometry.

A method that consider the deformations are presented in Villagrossi et al. (2017). A reference tool path is generated trough teaching on reference workpiece. The workpiece to be deburred and the reference are both scanned and registered using ICP to get a rough alignment. To compensate for the deformations, a set of control points are taught on the reference part in an area where burrs are highly unlikely. The same control points are found on the workpiece, and the local deformations are computed. The reference tool path is compensated based on the computed deformations. Kuss et al. (2016) propose a method where a series of CAD models are generated based on the dimensional tolerances of the workpiece CAD model. Registration using ICP is then performed on the scan of the workpiece and the set of CAD models. The CAD model with the best fit is used for generating a tool path using CAM software. Both proposed methods work well for simple geometries, but both will be challenging and time consuming for more complex geometries.

Béarée et al. (2011) and Huang et al. (2021) propose methods that segment the part and do registration on each sub-segment. The tool path is then adapted to each segment. Both methods are based on the assumption that the deformations within one segment is negligible.

Adjusting feed rate

A burr can be defined as undesirable or unwanted projections of the material formed as the result of a manufacturing process (Aurich et al., 2009). The shape and size of a burr can be described in different ways, it can be defined by its longitudinal and cross-sectional profile including thickness, height, and radius. The (ISO 13715:2019, 2020) standard use only one parameter to describe the burr. This is the height from the intended geometry and top of the burr as shown in Fig. 2.

During deburring, a set of machining parameters has to be set. One of these are feed rate. An approach to tuning this parameter is to choose a feed rate that is slow enough to be able to remove the largest burr. This will most likely increase the cycle time. If, on the other hand, the feed rate is too high, the tool can be unable to remove the burr. Methods have been proposed to adjust the feed rate based on burr height

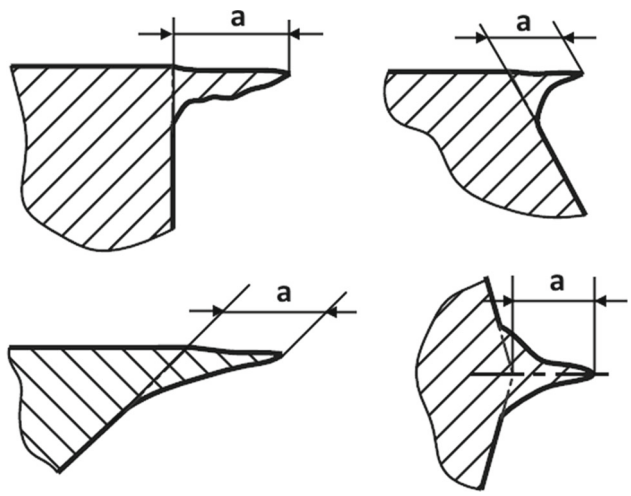


Fig. 2 External edge geometry as described in ISO 13715:2019 (2020)

(Lai et al., 2018; Xiong et al. 2018), where the feed rate is adjusted based on given burr height thresholds.

Burr measurement

There is a large number of burr measuring and detection methods available. The most appropriate system depend on the application, requested accuracy, and the burr values to be measured. Burr measurement systems can be divided into two main categories: in-process and out of process (Franke et al., 2010). Out of process can be divided into two categories: with contact and contactless. Contact methods include stylus methods and contactless methods include optical and electro-mechanical solutions. In-process methods include process monitoring, force and sound emission analysis.

Contact methods are only suitable for measuring burr height and can be limited due to workpiece material stiffness. Various optical systems for measuring burrs are available. These include camera systems, microscopes and lasers. The measured values from the optical systems are analyzed using specific measurement software. To characterize non-uniform burrs, several measurement are necessary.

A thorough comparison of burr measurement method formed by drilling and milling is presented in Franke et al. (2010). The results of the test show variations in the measurements, especially when the burrs are formed in an angle. It is concluded that very high caution is necessary for the comparison of burr measurements.

A burr measurement method based on burr surface area is proposed by Bahçe and Özdemir (2021), where they aim at evaluating burr height, arc length, area, and the geometrical characteristics of the burr. The proposed method capture a 3D scan of the burr and fit a cubic Bézier curve around the outline of the burr. All burr parameters are gathered from this curve.

Tellaeche and Arana (2016) use ICP to register a scan from a 3D vision systems with the CAD model. The matching is used to characterize different type of burrs.

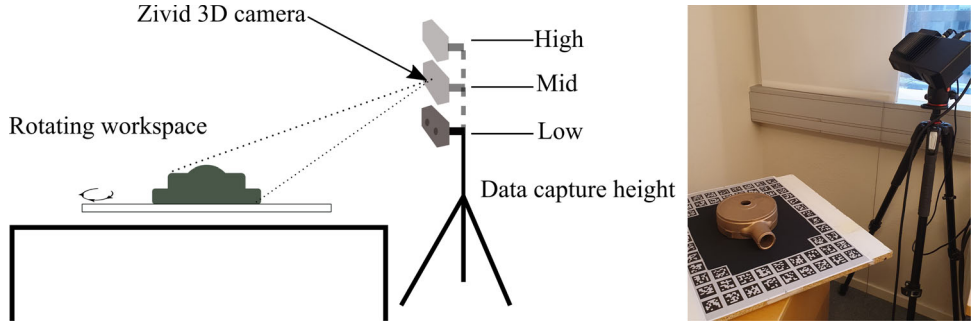
Summary

In summary, local CAD-scan registration approaches such as ICP, require a good initial guess of the transformation between CAD and scan for convergence. Furthermore, earlier works assume nearly identical CAD and scan ignoring the deformation caused by solidification process (domain gap). Proposed methods that consider the deformations are time consuming and not applicable for complex geometries.

Compared to earlier works, our CAD-Scan registration approach is robust to deformation caused by the solidification process, differences between CAD and scan, and relies on global registration unlike ICP. Moreover, burr measurement is based on local smoothness of scan surface by reconstructing burr free scan which do not assume exact match between CAD and scan. Common geometric features between CAD and scan that simulate domain differences are learned on the CAD model using sparse convolutional neural networks (CNN). In the next section, we present the proposed approach to learning features that are robust to domain gap between CAD and scan followed by burr measurement.

Methods

In the process of developing an automatic burr detection and deburring path planning system, having to align the reference CAD model to scan is the first step. Compared to CAD models, scans exhibit camera noise, missing data, manufacturing artifacts and geometric variations. Owing to the irregularity and variations between the reference CAD model and scan, it is difficult to align synthetic data to scan - which complicates burr measurement. To circumvent this problem, this paper proposes a deep learning model that extracts robust features from reference CAD model and scans that is robust CAD-to-scan domain gap, as well as variations in viewpoint, noise, rotation and scale. Figure 4 shows our pipeline for automated burr detection. Our acquisition setup captures high-resolution scans of the parts, with efficient alignment of registration ground truth (see "3D data acquisition" section). CNN features are extracted from a pre-trained model for the scan and the reference CAD model. Traditional CNN techniques are mainly applied to data with a structured grid, point cloud, on the other hand consists of sparse and unordered set of 3D points. These properties of point clouds make it difficult to use traditional CNN architectures for point cloud processing. Therefore, pre-trained model is based on a sparse CNN



(a) Schematic diagram **(b)** Capture setup

Fig. 3 3D image acquisition setup with rotating workspace and high-resolution 3D camera at different heights

deep learning model (Choy et al., 2019) trained only on CAD models (see "Training on CAD models" section). Finally, given the approximate alignment between the reference CAD model and scan, a burr detection approach (see "Burr detection and measurement" section) is proposed to quantify the height of the burr as shown in Fig. 4.

3D data acquisition

Our data acquisition setup consists of a turntable and a high-resolution 3D camera, to capture accurate point clouds of the parts and acquire ground truth registration data in an efficient manner. The sensor used in our setup is a Zivid One[1], which is an RGB-D camera based on structured light, with 1920×1200 pixels and a Field-of-View (FOV) of $40° \times 26°$. The acquisitions were performed with High Dynamic Range (HDR) imaging with 5 different exposure times from 40 to 100 ms, at a distance of about 0.5 m, which gives a nominal depth resolution of 0.18 mm.

Each workpiece is placed on a turntable with a dark background and ArUco markers along the edges, and recorded with 8 different rotation angles and 3 different camera heights (see Fig. 3). For each height, the camera tilt is adjusted to keep the part in view. This gives 24 pointclouds (scans) per workpiece, resulting in 480 total number of scans that are automatically co-registered during post-processing, using ArUco marker detection from OpenCV. However, in some of the scans, the ArUco markers were not detected with OpenCV, therefore, we are not able to establish accurate ground truth pose for these parts. After removing parts without ground truth pose, we are left with 285 scans in total. In this way, we only need one manual registration per workpiece for ground truth generation.

The acquired dataset consists of 4 parts with CAD models and 20 different physical workpieces in total (10 part ID 1, (7 with burr, 3 without burr), 2 from part ID 2, (1 with burr, 1 without burr), 7 from part ID 3, (5 with burr, 2 without burr), 1 from Part ID 4, (without burr)).

[1] https://www.zivid.com

Training on CAD models

Supervised training of deep neural networks requires large amounts of labeled data. This complicates their application to domains where training data is scarce and/or the process of collecting new datasets is laborious and expensive. Therefore, we aim to train an sparse CNN model using only CAD models which can be generalized to work on Zivid scans (more details on the network architecture in "Data pre-processing and training" section). We used CAD models as it is easy to generate in large numbers and geometric variations. As we wish to train solely on CAD models and have generalized performance on Zivid scans, the domain gap between CAD and scan need to be small as possible. Consequently, we seek to emulate the real domain by applying a series of augmentations to the CAD models. Moreover, as we have limited training samples the augmentations themselves increases our training samples to an arbitrary amount.

In addition to traditional augmentations such as point jittering, random rotation and translation, we simulate a casting burr randomly, generate a random viewpoint of the cloud and apply scaling which affects point density. The input to the random view-point generation (rendering) is the CAD model of the workpiece and then it is rendered from desired virtual view points. The generated point-view point can be controlled by randomly changing the camera position and orientation.

Given the CAD models, it is known a priori the possible location where a burr could form. Therefore, using the separation plane of the parts, we can simulate burrs by extruding additional points in this plane. By calculating the outward facing normals of the CAD point cloud, we know that point will extrude from the separation plane along the direction of the normal. Consequently, we produce extra points along the separation plane as

$$p_B = \vec{p}_i + \Delta L \vec{n}_i, \quad \forall i \in S \tag{1}$$

where p_B is a set of added burr points, p_i are points in the separation plane S and $\vec{n}_i$ is the associated outward facing

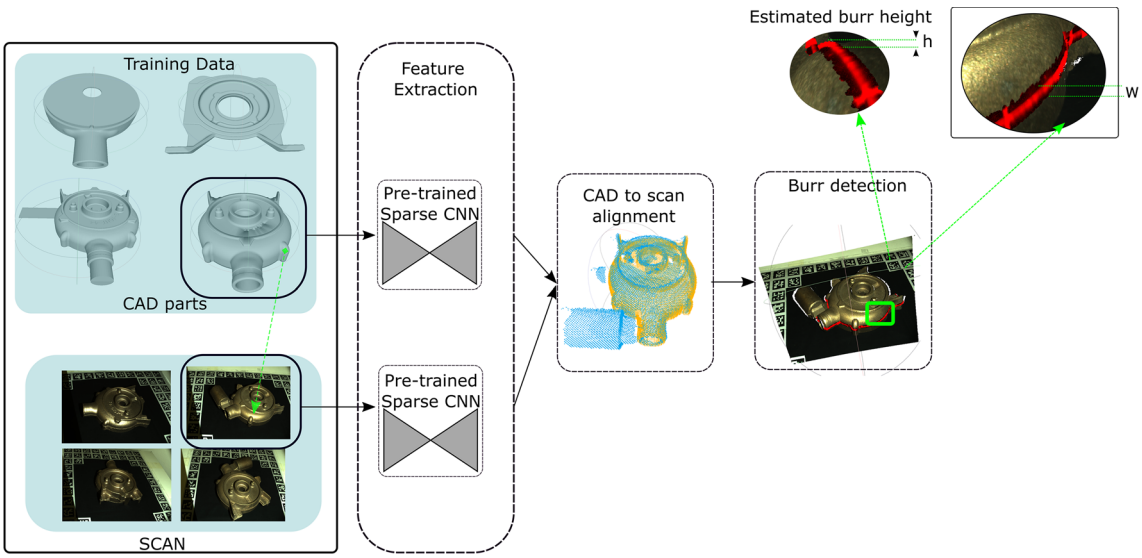


Fig. 4 Overview of the proposed approach: The dataset contains multiple CAD models and their corresponding scans taken from different view angles. To align CAD with the scan, DPFH features are extracted using pre-trained sparse CNN network. The extracted DPFH feature for both scan and CAD is used for global feature matching(alignment) with RANSAC. Finally, burr height is estimated by taking the distance between surface reconstructed scan and the input scan ("Burr detection and measurement" section)

normal of the point. A burr thickness of L can be achieved by iteratively adding burr points, p_B, with increasing ΔL_j until $\sum_j \Delta L_j = L$.

Our approach to scan-to-CAD alignment relies on the fact that geometric feature of CAD model ideally be locally invariant to different data augmentation that simulate scan(view point, point density, noise, jittering, etc) and can easily be computed (Khoury et al., 2017). However, traditional Fast Point Feature Histogram (FPFH) (Rusu et al., 2009) features are sensitive to such data augmentation and results in an incorrect alignment between CAD and scan. Therefore, we aim to make the FPFH descriptor insensitive to such data augmentation within the point neighbourhoods. To make traditional FPFH features robust against noise, occlusion and etc, we train a sparse CNN (Choy et al., 2019) that predicts a robust FPFH feature hereinafter referred to as DPFH (a Deep learning based FPFH feature descriptor). The learned features are robust to the transformation that simulate scan and the model is trained using solely the CAD models. The pseudocode for training DPFH feature network is summarized in Algorithm 1.

Algorithm 1 DPFH feature training procedure

Require: CAD models (C), FPFH features (Rusu et al., 2009) (Y), Data augmentations (T) ▷ Initialize the CNN model
$model \Leftarrow sparseUnet$
while Not converged **do**
  for $c_i, y_i \in C, Y$ **do**
    $c_i \Leftarrow T(c_i)$ ▷ apply random data augmentation
    $y_i \Leftarrow y_i(c_i)$ ▷ get the corresponding FPFH feature
    $\hat{y}_i \Leftarrow model(c_i)$ ▷ pass through model
    $loss(y_i, \hat{y}_i) \Leftarrow KLdiv(y_i, \hat{y}_i)$ ▷ KL divergence loss
    minimize: $loss(y_i, \hat{y}_i)$
  end for
end while
return $model$

Burr detection and measurement

Based on the DPFH features we get an initial proposed registration, in terms of rotation R_0 and translation T_0, using RANSAC with parameters as in Choi et al. (2015). In order to achieve accurate burr measurement, the alignment of scan and CAD has to be close to perfect. This is implausible as burrs and other deficiencies in the scan will affect the resulting registration. Consequently, in order achieve accurate burr measurement further processing is needed. Approximate location of burr relative to CAD model is pre-computed (ROI on CAD) since casting process is known and a burr is formed in the separation plane between cope and drag. As casting burrs are mostly present in the separation plane, we have *a priori* information regarding where the deficiencies are most likely present in the CAD model, shown in Fig. 5a Therefore, we can extract points in a region of interest (ROI) as;

$$p_{C,\mathrm{ROI}} = p_{C,i} \quad \forall i \in \mathrm{ROI} \tag{2}$$

$$p_{S,\mathrm{ROI}} = p_{S,i} \quad \forall i \quad \text{where} \quad \left\| p_{S,i} - p_{C,\mathrm{ROI}} \right\| < \epsilon. \tag{3}$$

here, p_C and p_S describe the points in the CAD and scan point clouds respectively. The ROI is extracted as a thick slab in the separation plane in the CAD model. The points in the scan that are within the ROI are chosen as all points

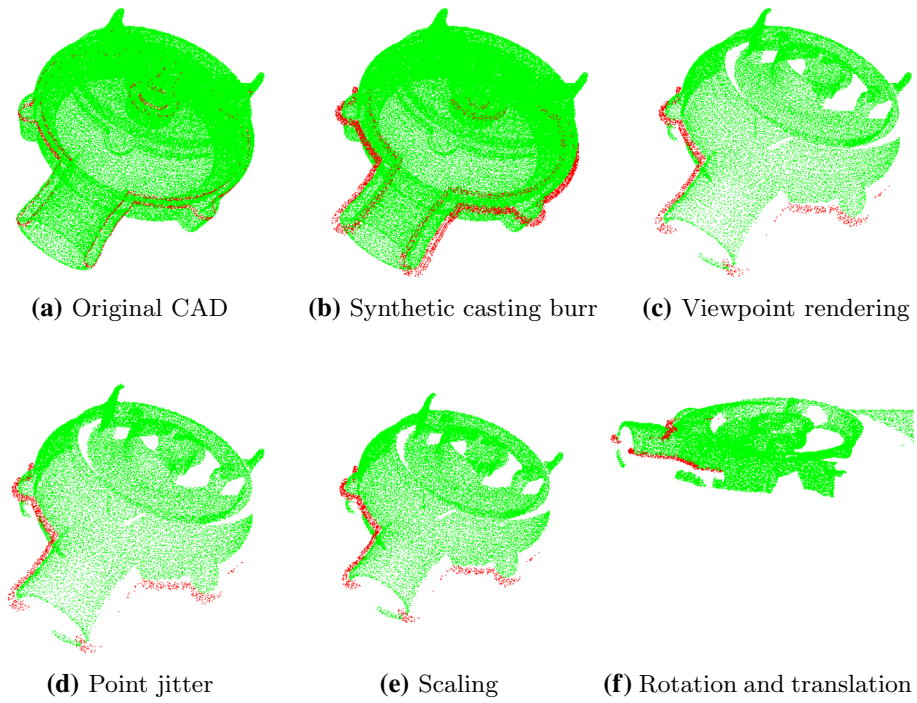


Fig. 5 Sample augmentations. In **a** the original point cloud of the CAD model is shown. The burr locations are colored red. In **b** a synthetic burr is added by adding points along the casting plane. In **c** the points that are not visible for a given camera angle and height are removed. Further, random jittering is applied **d** different scaling is applied (affecting point density) as in **e** and random rotation and translation is applied (**f**) (Color figure online)

that have a distance norm to its closest neighbour in the CAD ROI below some threshold ϵ.

Algorithm 2 Burr measurement algorithm

Require: Roughly aligned CAD (C) and scan (S) point cloud

▷ Extract ROI. See Fig. 6b

$$C_{\text{ROI}} \Leftarrow C_i, \quad \forall i \in \text{ROI}$$
$$S_{\text{ROI}} \Leftarrow S_i, \quad \forall i \in d(S_i, C_{\text{ROI}} < \epsilon)$$

▷ Calculate normals with different radii, $r_l > r_q$, see Fig. 6c

$$\vec{n}_S^{r_l} = normals(S_{\text{ROI}}, r_l)$$
$$\vec{n}_S^{r_q} = normals(S_{\text{ROI}}, r_q)$$
$$\vec{n}_C = normals(C_{\text{ROI}}, r_l)$$

▷ Calculate difference of normals

$$\Delta D_s = \frac{\vec{n}_S^{r_l} - \vec{n}_S^{r_q}}{2};$$

▷ Initial selection of non-burr points

$\hat{S}_i \Leftarrow S_{i,j} \quad \forall j \in \text{Nearest neighbors}(S_i(\Delta D_s)) \quad \forall \Delta D_s < \sigma$

$\hat{S}_i \Leftarrow Poisson(\hat{S}_i)$ ▷ Poisson surface reconstruction, see Fig. 6d

▷ Calculate the distance between reconstructed surface and scan, see Fig. 6e

$$L_i \Leftarrow \left| \left(\hat{S}_i - S_{i,j} \right) \vec{n}_{C,j} \right| \quad \forall j \in \text{Nearestneighbors}(S_{i,ROI})$$

$M_b \Leftarrow 1 \quad \forall L_i > \sigma$

$L_b \Leftarrow L_i \quad \forall L_i > \sigma$

return Burr point mask M_b and thickness L_b

Given the ROI, difference of normals (Ioannou et al., 2012) are computed. The intuition behind using difference of normals here is that if the direction of the two surface normals computed at different radius is nearly identical, then the structure of the surface does not change significantly. By contrast, if there are burrs on the surface, the direction of the two estimated normals are likely to vary by a larger margin. Figure 6c shows difference of normals along the ROI region. Therefore, a burr is detected by thresholding the difference of normals above some ϵ.

To simulate a burr-free scan, Poisson surface reconstruction (Kazhdan et al., 2006) is applied after removing the detected burr points. The Poisson surface reconstruction method solves a regularized optimization problem to obtain a smooth scan surface without burrs. Finally, the burr height is approximated by taking the distance between original and burr free surface reconstructed scan. The complete overview of burr detection and measurement approach is given in Algorithm 2.

Data pre-processing and training

The network is trained only on CAD models. We converted the workpiece CAD model from STEP 3D file to point cloud by sampling 1M points to generate point clouds with uniform density. Two out of four CAD models are used for training. Note that none of 285 scans are used in the training. We use a Sparse Residual U-Net (Sparse CNN, Choy et al., 2019) architecture in this work. It is a 34-layer U-Net architecture that has an encoder network of 21 convolution layers and a

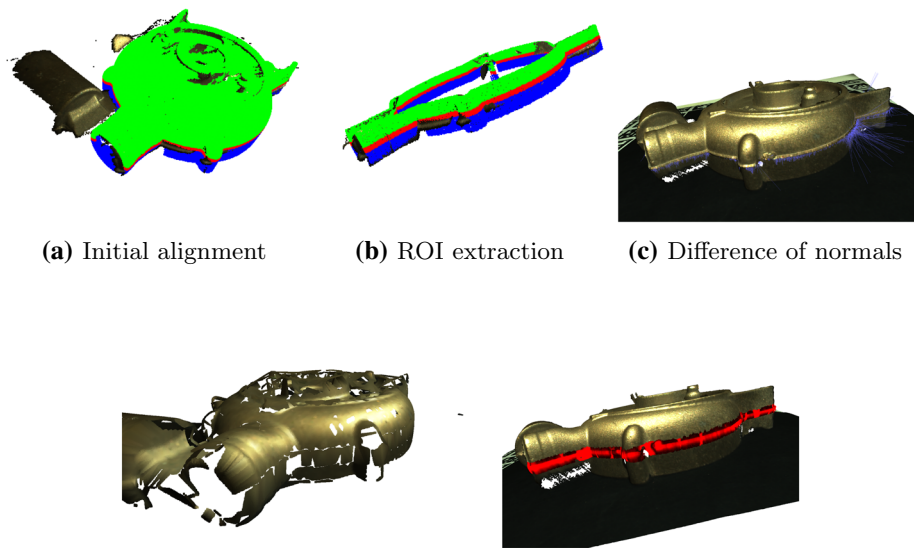


Fig. 6 Burr measurement. In **a** rough alignment of CAD and scan is shown. The prior information on burr locations are colored red. In **b** the region of interest interest (ROI) is extracted along the burr location. In **c** a difference of normals is calculated for ROI region and based on difference of normals, burr points are removed and the surface is reconstructed in **d**. **e** shows the final detected burr. The burr height is represented as bright red (large burr) and dark (no burr). (Best viewed in color) (Color figure online)

decoder network of 13 convolution and deconvolution layers. It follows the 2D ResNet basic block design and each conv/deconv layer in the network is followed by Batch Normalization and ReLU activation. The overall Sparse CNN architecture has 37.85M parameters. The Sparse CNN architecture was originally designed in Choy et al. (2019) that achieved significant improvement over prior methods on the challenging ScanNet (Dai et al., 2017) semantic segmentation benchmark. We train the sparse CNN network on a single RTX A6000 with batch size 32 and voxel-size 1cm. Adam is used with a learning rate $1e^{-3}$. We trained the network for 500 epochs, and observed that the loss converged after approximately 100 epochs. The network outputs the estimated DPFH feature, $\hat{y}$, and the final loss function has the following form.

$$\mathscr{L}_{KL}(y, \hat{y}) = \frac{1}{M}\sum_{i=1}^{M} y_i \log \frac{y_i}{\hat{y}_i} \tag{4}$$

where M is the total number of points. For registration, we used global RANSAC registration based on feature matching and using $r_n = 3$ correspondences.

Results

We analyze the proposed approach pipeline in two scenarios: CAD to scan registration where we estimate an SE(3) transformation between CAD models and the Zivid scans, and burr size estimation which generates a per point estimation of burr height for all viewing angles of the workpiece.

Scan-to-CAD registration evaluation

For global registration, we use RANSAC (feature matching). For each iteration of RANSAC, we select N random points from the CAD point cloud. The nearest neighbor in the DPFH feature space is queried to find their corresponding points in the scan point cloud. We examine if aligned point clouds are near together to quickly discard erroneous matches (4cm). A transformation matrix is computed using the remaining points. RANSAC is conducted for 500 iterations (Open3d.org, 2015).

The performance of CAD to scan registration algorithm is evaluated under different views of the workpieces using the Translation Error (TE) and Rotation Error (RE) metrics that measures the deviations between the predicted and and ground truth pose as defined in Elbaz et al. (2017). Given the ground truth rotation $\mathbf{R}$ and translation $\mathbf{T}$ of each object, the TE and RE are defined as follows:

$$\begin{aligned} TE &= \|\tilde{\mathbf{T}} - \mathbf{T}\| \\ RE &= arccos\left(\frac{trace(\tilde{\mathbf{R}}^T\mathbf{R}) - \mathbf{1}}{2}\right) \end{aligned} \tag{5}$$

where $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{R}}$ denote the estimated translation vector and translation matrix, respectively. Note in this experiment we train the encoder network only on CAD models and do not fine-tune on the other datasets.

We report the registration results on the dataset, which contains 4 different CAD models as discussed in "3D data acquisition" section. We measure translation error (TE) and rotation error (RE) as defined in Eq. 5, and accuracy. Accuracy is the ratio of successful scan to CAD registrations and we define a registration to be successful if its rotation error

Table 1 Rotation and translation error average over scans with burrs (B) and feeder(P)

Part		CAD 1 Workpieces								CAD 2 Workpieces	CAD 3 Workpieces						
Metric	Method	1B-1	1B-2	1B-3	1B-4	1B-6	1B-7	1B-8	Avg.	2P-1	3B-1	3B-2	3B-3	3B-4	3B-5	Avg.	Average
Acc	DPFH	**1.0**	**1.0**	**0.88**	**1.0**	**1.0**	**1.0**	**0.94**	**0.97**	**0.95**	**1.0**	**0.84**	**0.67**	**0.6**	**0.74**	**0.77**	**0.89**
	FPFH	0.56	0.9	0.71	0.75	0.82	0.82	0.71	0.75	0.58	0.25	0.32	0.33	0.4	0.47	0.35	0.59
Rerr	DPFH	**5.05**	**4.24**	**9.67**	**4.23**	**4.14**	**3.85**	**7.27**	**5.49**	**12.11**	**2.47**	**20.3**	**34.35**	**46.68**	**29.33**	**26.63**	**14.13**
	FPFH	57.68	19.89	46.03	31.36	33.05	27.15	51.25	38.06	63.65	99.2	56.55	70.05	72.38	45.8	68.8	51.85
Terr	DPFH	**5.61**	**5.27**	**4.17**	**4.27**	**4.2**	**4.67**	**3.4**	**4.51**	**7.34**	**6.79**	**3.66**	**5.17**	**6.12**	**7.28**	**5.8**	**5.23**
	FPFH	24.44	11.46	15.56	20.6	18.1	7.94	21.74	17.12	12.28	84.5	17.86	38.64	14.73	13.62	33.87	23.19

The accuracy is estimated by taking for rotation and translation error less than θ_r and T_t respectively. $RE < 10°$, $TE < 10$mm are in degrees and mm
Bold signifies the best-performing method

Table 2 Rotation and translation error average of scans after manual de-burring

Part		CAD 1 Workpieces				CAD 2 Workpieces	CAD 3 Workpieces			CAD 4 Workpieces	
Metric	Method	1N-1	1N-2	1N-3	Avg.	2N-1	3N-1	3N-2	Avg.	4N-1	Average
Acc	DPFH	**1.0**	**1.0**	**1.0**	**1**	**1.0**	**0.78**	**0.76**	**0.77**	**1.0**	**0.93**
	FPFH	0.0	0.67	0.62	0.43	0.83	0.44	0.35	0.39	0.7	0.51
Rerr	DPFH	**4.88**	**3.85**	**2.73**	**3.82**	**4.69**	**27.34**	**27.06**	**27.2**	**2.15**	**10.38**
	FPFH	146.25	51.99	59.19	85.81	20.3	49.96	74.6	62.28	42.91	63.6
Terr	DPFH	**6.54**	**4.75**	**3.52**	**4.94**	**3.84**	**5.26**	**3.91**	**4.58**	**5.19**	**4.71**
	FPFH	69.67	13.94	29.12	37.58	11.91	13.56	24.9	19.23	5.91	24.14

The accuracy is estimated by taking for rotation and translation error less than θ_r and T_t respectively. $RE < 10°$, $TE < 10$ mm are in degrees and mm
Bold signifies the best-performing method

and translation error are smaller than predefined thresholds. Average TE and RE are computed for different views workpiece of each part.

We compare robustness of learned features, using the proposed DPFH for registration with the classical FPFH as baseline. The result is separately shown for physical parts with burr Table 1 and without burr Table 2 for clarity. In Tables 1 and 2, we measure accuracy with the TE threshold 10mm which is practical with the voxelization process, and RE threshold 10 degrees which is typically a good initial value for registration refinement. For this experiment we used 4 CAD models (parts) and 285 scans taken from different view points. Our learned features outperforms the baseline on accuracy by a large margin (91% vs. 56%) and achieves the lowest translation (5.05mm vs. 23.52) and rotation error (12.82° vs 55.96°) consistently on both scans with burr and no-burr, and CAD models. In Table 3, we show the RANSAC registration fitness for both DPFH and FPFH by taking the ratio of the inlier correspondences to the number of points in the Zivid scans. Higher RANSAC registration fitness value indicates DPFH feature gives a higher matching points for CAD to scan registration task. We additionally show qualitative results in Fig. 7

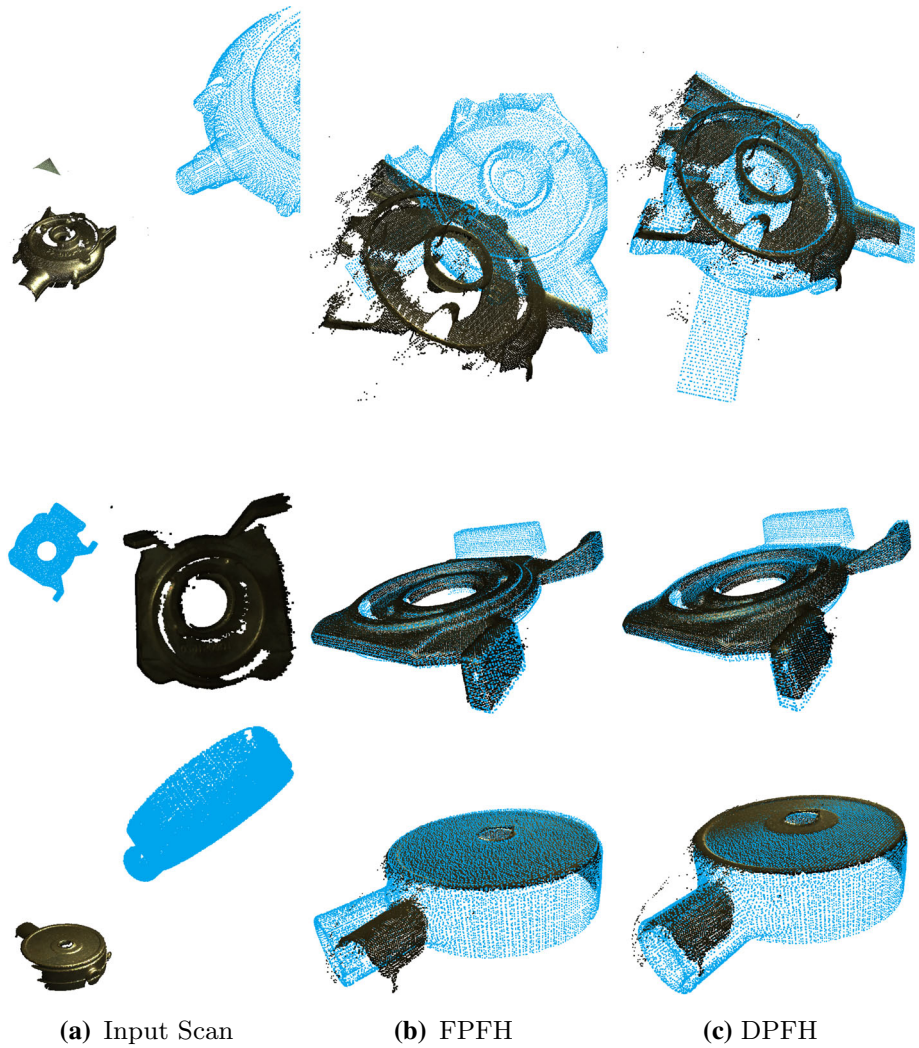
Table 3 Registration Fitness (higher is better)

Part	Fitness DPFH	FPFH
CAD1	0.313	0.260
CAD2	0.312	0.268
CAD3	0.341	0.140
CAD4	0.351	0.350
Average	**0.329**	0.254

For all real scans, DPFH feature gives a higher matching points as compared to FPFH feature
Bold signifies the best-performing method

Burr size evaluation data

Although quantitative evaluation on the Zivid scans serves the purpose of quantifying the burr quantization algorithm, it is important to compare the estimated burr size with a high resolution industrial burr size measuring device. An industrial computed tomography (CT) scanning machine (Zeiss Metrotom 1500) is used to scan the workpiece and produce a volumetric representation of the scanned object. Volume



(a) Input Scan **(b)** FPFH **(c)** DPFH

Fig. 7 Qualitative comparison of alignments on three different CAD models. Our approach to learning feature by mimicking real scans produce reliable correspondences, which coupled with RANSAC algorithm, produces significantly more accurate alignments

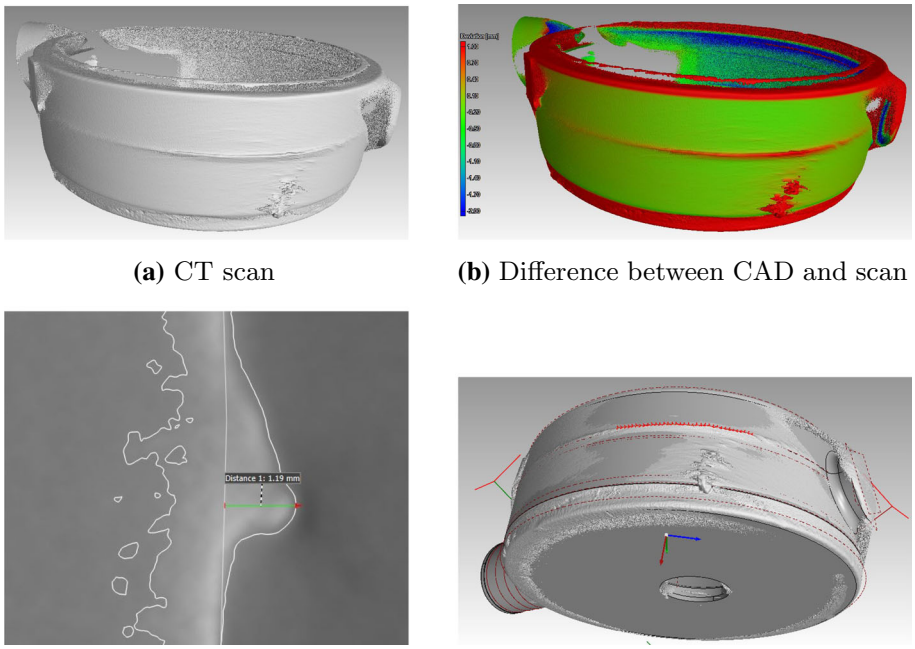
Graphics Software was used to analyze the scans, register the parts, and measure the burr height (Fig. 8).

Burr detection and size estimation evaluation

In the following, we compare our approach to estimate burr height with a high resolution industrial CT scanning machine. The workpiece materials in this study is shown in Fig. 9. Using CT scanning machine as described in "Burr size evaluation data" section, burr height of discrete 100 points were measured. Using CT scanning machine, the burr height is measured to be 0.79 ± 0.17 mm for the sampled discrete points. Figure 10 shows the region of interest and the distribution of burr height as estimated using our approach. To compare our estimate with high resolution CT scan result, we conducted a two sample z-test. The result shows the proposed approach provides similar result with the industrial standard burr height estimation technique ($z = 0.279$).

Discussion and conclusion

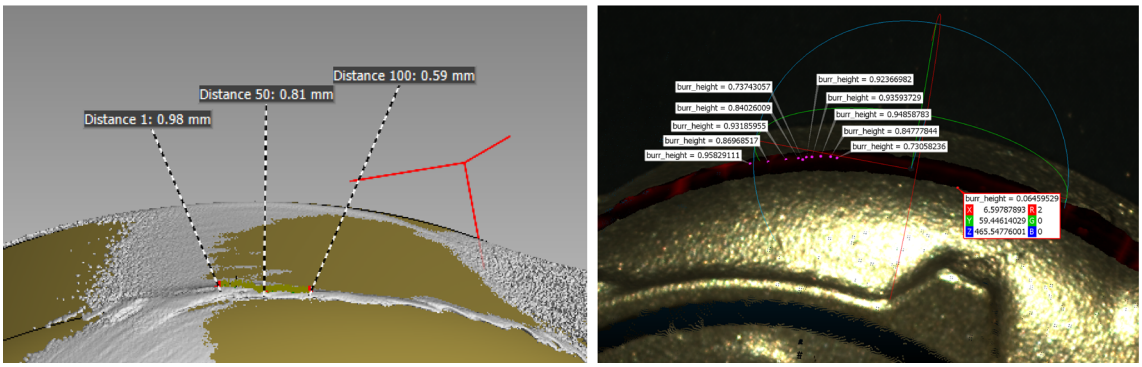
In this paper, we investigated burr size estimation and registration problem using 3D vision sensors. For the CAD models of the workpieces, the average translation and rotation errors of CAD models to real Zivid scans are shown in Tables 1 and 2 . The result reveals that the maximum translation error occurs for a workpiece 2P-1. This could be due the fact that the scans of 2P-1 have extra feeding component that is not part of the objects CAD model. Furthermore, workpieces that are symmetric and lack geometric variations (e.g. 3B-2) are challenging as it is not possible to find unique points for registration resulting in a large rotation error (Choy et al., 2020). While the focus of this work is mainly on burr height estimation and registration given the CAD and scan, the approach could further be improved by including an algorithm for CAD model retrieval (finding the most similar model). One extension of the proposed method would be to retrieve the CAD model given the scan. This could be accomplished by design-



(a) CT scan **(b)** Difference between CAD and scan

(c) Measured burr height **(d)** Measured burr region

Fig. 8 CT scan. The raw CT scan is shown in (**a**). In **b**, the scan is registered with the CAD model. The colors indicate the difference between CAD and scan. The height of the burr is measured as the distance from the CAD model and the top of the burr (**c**). A burr region was measured for comparison with the Zivid scans (**d**) (Color figure online)

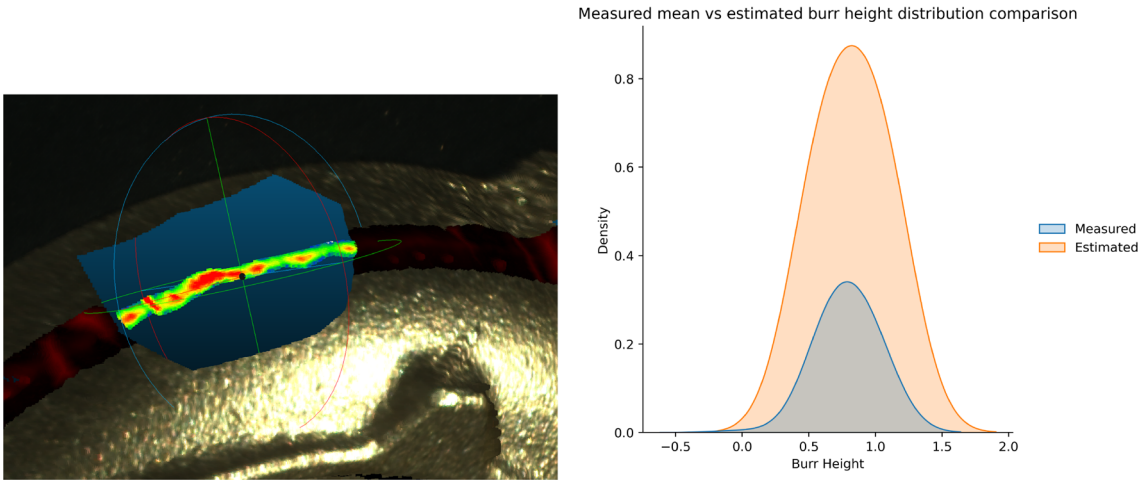


(a) Ground truth burr height **(b)** Estimated burr height

Fig. 9 Comparison of ground truth burr height as explained in "Burr size evaluation data" section and the estimated burr height using our approach

ing a neural network architecture that is specifically trained on shape similarity between scan and CAD geometry.

Our primary contribution is to provide a high-resolution characterization of burrs using 3D vision sensors. To do so, we developed a deep learning model that is trained unsupervised only on CAD models by mimicking real scans. The developed model is robust to differences between CAD and real world scans such as variations to point density as well as missing data in the scans. Figure 11 shows a side by side comparison of CAD model, RGB image and captured point cloud highlighting the domain gap between CAD and real scans. The proposed method is general in that it works with multiple CAD models and able to handle variations in physical parts due to solidifation process. Therefore, applicable in industrial settings. Together with registration, our burr height estimation approach is able to estimate burr height similar to high resolution CT scans with Z-statistic ($z = 0.279$).



(a) Burr height sampled region

(b) Estimated burr height distribution comparison

Fig. 10 Estimated burr height for region of interest and histogram of burr (325 sample points on the scan)

(a) Original CAD **(b)** Synthetic casting burr **(c)** Detected burr

Fig. 11 Result on missing/noisy data: The ID2 cad model is shown **a** and **b** shows the RGB image of one of the physical workpieces of ID2 part standing on rotating workspace with camera facing the backside of the workpiece, see Fig. 3. In **c** we show the scanned point cloud with sensor noise and missing parts with the detected burr. Note that with the learned features, we are able to register the clean CAD model with the noisy scan shown in **c** and detect burrs

Acknowledgements The work reported in this paper was based on activities within centre for research based innovation SFI Manufacturing in Norway, and is partially funded by the Research Council of Norway under contract number 237900. We would like to thank Mjøs Metallvarefabrikk AS www.mjosmetall.no for providing access to data and valuable feedback.

Funding Open access funding provided by SINTEF.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Aurich, J. C., Dornfeld, D., Arrazola, P., et al. (2009). Burrs-analysis, control and removal. *CIRP Annals, 58*(2), 519–542. https://doi.org/10.1016/j.cirp.2009.09.004

Bahçe, E., & Özdemir, B. (2021). Burr measurement method based on burr surface area. *International Journal of Precision Engineering and Manufacturing-Green Technology, 8*(4), 1287–1296. https://doi.org/10.1007/s40684-020-00228-0

Béarée, R., Dieulot, J. Y., & Rabaté, P. (2011). An innovative subdivision-icp registration method for tool-path correction applied to deformed aircraft parts machining. *The International Journal of Advanced Manufacturing Technology, 53*(5), 463–471. https://doi.org/10.1007/s00170-010-2875-0

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV: Control paradigms and data structures, SPIE*, (pp. 586–606). https://doi.org/10.1117/12.57955.

Choi, S., Zhou, Q. Y., & Koltun, V. (2015). Robust reconstruction of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5556–5565). https://doi.org/10.1109/cvpr.2015.7299195.

Choy, C., Park, J., & Koltun, V . (2019). Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference*

on computer vision (pp. 8958–8966). https://doi.org/10.1109/iccv.2019.00905

Choy, C., Dong, W., & Koltun, V . (2020). Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2514–2523). https://doi.org/10.1109/cvpr42600.2020.00259.

Dai, A., Chang, A. X., & Savva, M., et al. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5828–5839). https://doi.org/10.1109/cvpr.2017.261.

Du, S., Liu, J., Zhang, C., et al. (2015). Probability iterative closest point algorithm for md point set registration with noise. *Neurocomputing, 157*, 187–198. https://doi.org/10.1016/j.neucom.2015.01.019

Elbaz, G., Avraham, T., & Fischer, A. (2017). 3D point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4631–4640). https://doi.org/10.1109/cvpr.2017.265.

Erwin Aertbeliën. (2009). Development and acquisition of skills for deburring with kinematically redundant robots. PhD thesis, Catholic University of Leuven.

Franke, V., Leitz, L., & Aurich, J . (2010). Burr measurement: a round robin test comparing different methods. In *Burrs-analysis, control and removal* (pp 167–178). Springer, https://doi.org/10.1007/978-3-642-00568-8_18.

Huang, W., Mei, X., & Jiang, G., et al. (2021). An on-machine tool path generation method based on hybrid and local point cloud registration for laser deburring of ceramic cores. *Journal of Intelligent Manufacturing* (pp. 1–16). https://doi.org/10.1007/s10845-021-01779-y.

Ioannou, Y., Taati, B., & Harrap, R., et al. (2012). Difference of normals as a multi-scale operator in unorganized point clouds. In *2012 second international conference on 3D imaging, modeling, processing, visualization & transmission, IEEE* (pp. 501–508), https://doi.org/10.1109/3dimpvt.2012.12.

ISO 13715:2019,. (2020). *Edges of undefined shape-indication and dimensioning*. International Organization for Standardization, Geneva, CH: ISO standard.

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*.

Khoury, M., Zhou, Q. Y., & Koltun, V. (2017). Learning compact geometric features. In *Proceedings of the IEEE international conference on computer vision* (pp. 153–161). https://doi.org/10.1109/iccv.2017.26.

Kosler, H., Pavlovčič, U., Jezeršek, M., et al. (2016). Adaptive robotic deburring of die-cast parts with position and orientation measurements using a 3d laser-triangulation sensor. *Strojniški vestnik-Journal of Mechanical Engineering, 62*(4), 207–212. https://doi.org/10.5545/sv-jme.2015.3227

Kuss, A., Drust, M., & Verl, A. (2016). Detection of workpiece shape deviations for tool path adaptation in robotic deburring systems. *Procedia CIRP, 57*, 545–550. https://doi.org/10.1016/j.procir.2016.11.094

Lai, Z., Xiong, R., & Wu, H., et al. (2018). Integration of visual information and robot offline programming system for improving automatic deburring process. In *2018 IEEE international conference on robotics and biomimetics (ROBIO), IEEE* (pp. 1132–1137). https://doi.org/10.1109/robio.2018.8665148

Onstein, I., Semeniuta, O., & Bjerkeng, M. (2020). Deburring using robot manipulators: A review. In *2020 3rd international symposium on small-scale intelligent manufacturing systems (SIMS), IEEE* (pp. 1–7). https://doi.org/10.1109/SIMS49386.2020.9121490

Open3d.org. (2015). Global registration-open3d latest (664eff5) documentation. http://www.open3d.org/docs/latest/tutorial/Advanced/global_registration.html.

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation, IEEE* (pp. 3212–3217). https://doi.org/10.1109/robot.2009.5152473.

Song, H. C., & Song, J. B. (2013). Precision robotic deburring based on force control for arbitrarily shaped workpiece using cad model matching. *International Journal of Precision Engineering and Manufacturing, 14*(1), 85–91. https://doi.org/10.1007/s12541-013-0013-2

Tellaeche, A., & Arana, R. (2016). Robust 3d object model reconstruction and matching for complex automated deburring operations. *Journal of Imaging, 2*(1), 8. https://doi.org/10.3390/jimaging2010008

Villagrossi, E., Cenati, C., Pedrocchi, N., et al. (2017). Flexible robot-based cast iron deburring cell for small batch production using single-point laser sensor. *The International Journal of Advanced Manufacturing Technology, 92*(1), 1425–1438. https://doi.org/10.1007/s00170-017-0232-2

Xiong, R., Lai, Z., & Guan, Y., et al. (2018). Local deformable template matching in robotic deburring. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE* (pp. 401–407). https://doi.org/10.1109/robio.2018.8665325.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.