Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 134 (2025) 437–442

www.elsevier.com/locate/procedia

Proceedings of the 58th CIRP Conference on Manufacturing Systems 2025

A data augmentation algorithm for surface inspection in point cloud data

Daan Büchner[a,b]*, Ole Schmedemann[b], Thorsten Schüppstuhl[b]

[a]3D.aero GmbH, Billhorner Deich 96, 20539 Hamburg, Germany
[b]Hamburg University of Technology, TUHH, Institute of Aircraft Production Technology, Denickestr. 17, 21073 Hamburg, Germany

* Corresponding author. Tel.: +49-151-57166723. *E-mail address:* dbuechner@3d-aero.com

Abstract

Due to the high standards in aircraft maintenance, high resolution sensors, such as white light interferometers, are needed. Those sensors scan surfaces in nanometer scale and generate point clouds. This data can be used to detect surface defects. Such anomalies should be identified during the inspection process to assess the current condition of the workpiece. Deep learning algorithms can be used to evaluate the data. However, in the domain of 3D data, the challenge of obtaining training data is amplified due to the time-consuming labeling process. Therefore, this work introduces an algorithm that combines surface features, like cracks, into surface data to generate new labeled training data. The resulting dataset is then used to train a deep learning algorithm to segment the cracks in the point cloud data. The results indicate that the augmented data enhances the training process.

Keywords: Machine learning; 3d data; surface inspection; data augmentation

1. Introduction

In an aircraft, the most expensive components are jet engines. Therefore, the maintenance has the most cost effectiveness by repairing parts from the jet engine instead of substituting them. In regular maintenance intervals the jet engines are disassembled, and subcomponents are inspected to locate defects in the submillimeter range [1]. For this task high-resolution sensors are essential. One such sensor is the white light interferometer, an optical device capable of acquiring 3D data at the nanometer scale. This capability enables automated inspection of components for defects at this scale [2]. In addition, algorithms are required to effectively analyze the huge among of the acquired 3d data to ensure accurate assessment.

Conventional algorithms, such as the one developed by Otto [3], are available for defect detection. This algorithm, for instance, is specifically designed to detect cracks in rotationally symmetric combustion chambers scanned by a white light interferometer. However, this approach is highly specialized for crack detection and is not suitable for identifying general defects; similar structures are often misclassified as cracks. Recent advances in deep learning offer a promising alternative, providing solutions to overcome the limitations associated with conventional data evaluation algorithms.

A major challenge for deep learning algorithms is obtaining high-quality training data [4, 5]. This requires a balanced distribution of classes, comprehensive representation of each class's features, and a sufficiently large dataset. In defect detection, creating such a training set is particularly difficult due to the considerable variability in defect appearances, the uneven occurrence rates of certain defects, and the need for expert labeling. Additionally, labeling 3D data is notably time-consuming, as it requires examination from multiple perspectives, which further increases the cost and complexity of generating a training set for 3D data.

An effective approach to increasing variability in training datasets for deep learning is data augmentation, which has shown promising results [6]. In this process, existing data are modified to generate new samples, enhancing the dataset

2212-8271
10.1016/j.procir.2025.02.142

without requiring additional data collection. Data augmentation can involve simple techniques, such as applying jitter to the data, or more advanced methods, like combining different data samples to create novel representations [7].

In this work, a data augmentation algorithm for point clouds is presented that inserts defects into surfaces to create new training samples. The training samples should be used to enhance the training process for deep leaning algorithms in the field of defect detection in point clouds of part surfaces. A training dataset with ground truth data representing surface cracks was created and compared to training with unaltered real data samples, enabling evaluation of the algorithm's effectiveness in enhancing defect detection performance.

2. Related work

2.1. Defect Detection in Point Clouds

Algorithms for point cloud analysis can be categorized into conventional and deep learning approaches. Conventional approaches are rule-based, applying predefined rules to classify structures within a point cloud, such as identifying defects [8, 9, 10, 11].

In recent years, deep learning methods have gained significant attention and have been increasingly applied to defect detection [12]. For instance, Zhang et al. [13] employed a deep learning approach to detect cracks in pavement by first projecting point cloud data onto an image plane before processing it with a deep learning model. Similarly, Zhao et al. [14] projected 3D data into a plane to detect surface defects in continuous casting products, and Lafiosca et al. [15] used an image plane projection to identify dents in aircraft parts. In contrast, Wang et al. [16] proposed a method that directly operates on point clouds, utilizing graph-based deep learning techniques for defect detection.

2.2. Deep Learning Algorithms for Point Clouds

While deep learning has been extensively explored for 2D data, there is growing interest in developing algorithms for 3D data, specifically for semantic segmentation in point clouds. One of the pioneering algorithms for point cloud semantic segmentation was PointNet by Qi et al. [17]. This approach was later enhanced by Qi et al. [18] with PointNet++, which introduced a hierarchical spatial structure to improve performance on local features. Qian et al. [19] further refined this approach with novel training and scaling strategies, enhancing segmentation accuracy. Xiong et al. [20] introduced a graph-based structure in their deep learning model for point clouds, while algorithms such as KPConv [21] and PointConv [22] employed weighted convolutions for more effective point cloud processing.

In recent years, transformer models have significantly impacted natural language processing and 2D data classification, sparking interest in their application to point clouds. This led to the development of PointTransformer by Zhao et al. [23]. Another model, PointMLP by Ma et al [24] , introduced a residual MLP network that, unlike other models, does not rely on a complex local geometry extractor. Many algorithms, however, struggle with scalability as point cloud size increases. To address this, Hu et al. [25] proposed RandLA-Net, which incorporates random sampling between layers to maintain efficiency with larger point clouds.

2.3. Data Augmentation for Point Clouds

Data augmentation for point clouds can employ simple techniques such as affine transformations, where data is shifted, rotated, flipped, or scaled. Additional techniques include adding jitter to introduce slight noise or randomly dropping points to increase model robustness.

In the last years, more advanced techniques for data augmentation have been investigated in recent years. Sheshappanavar et al. [26] applied simple augmentation techniques with varying parameters on specific patches of point cloud data, while Choi et al. [27] implemented a part-aware approach applicable only to outdoor datasets to prevent object overlap. Kim et al. [28] introduced locally weighted transformations for data augmentation. Rather than local augmentations, Wang et al. [29] developed a patch-mixing method, using a scoring module to generate contextually relevant targets, and Shi et al. [30] placed ground-truth samples into scenes with multimodal context guidance. Zhang et al. [31] proposed a general approach, combining point clouds by optimally aligning and replacing points in one sample based on another's optimal assignment. Leng et al. [32] extended training data by using pseudo- labeled objects placed in scenes. Leveraging generative models, Reichardt et al. [33] created 3D models with a text-to-3D generator for integration into point cloud scenes. Inspired by image interpolation, Chen et al. [34]. interpolated between two point clouds to generate new samples. Global augmentation techniques include Vu et al. [35], who used upsampling and reconstruction methods, and Zhang et al. [36], who employed principal component analysis to align point clouds and enhance rotation-invariant feature learning.

For data augmentation, it is advantageous to take the data domain into account. Xiao et al. [37] developed a data augmentation algorithm that leverages the data structure of Lidar by merging point clouds using polar coordinates. Similarly, Qiu et al. [38] proposed an augmentation approach for Lidar data involving smooth deformations. Furthermore, Hasecke et al. [39] focused on maintaining context information within the autonomous driving domain when combining point clouds.

Native sensor data from the 3D domain is still frequently projected into 2D space to facilitate the application of deep learning algorithms. While some methods can process native 3D data directly, the challenge of obtaining sufficient training data remains a significant limitation. Data augmentation is a widely adopted strategy to enhance available datasets in such cases. In this work, we introduce a novel data augmentation algorithm for point clouds to enable deep learning algorithms for the surface inspection task with a white light interferometry sensor. To leverage contextual information, the algorithm is specifically designed consideration of the sensor's characteristics, ensuring the preservation of its native data structure.

3. Method

This section discusses the data augmentation algorithm implemented for point clouds. First, the conceptual framework of the algorithm is introduced, followed by a detailed explanation of its implementation.

3.1. Concept for the data augmentation algorithm

The data augmentation algorithm is tailored for the surface inspection of point clouds. Fig. 1 presents a sample dataset featuring a defect on a surface; in this example, the defect is specifically identified as a crack.

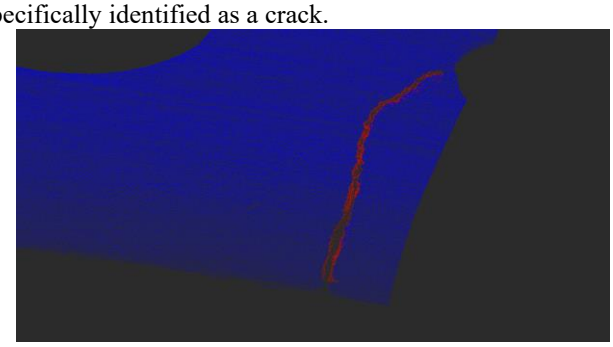

Fig. 1. Example of a point cloud of a surface with a crack acquired by a white light interferometer. Blue: surface; red: crack: black: background.

To generate augmented data, 3D point cloud defect segments are inserted into defect-free surface samples, both sourced from real-world objects. This approach supports various defect types and is therefore divided into two components. The first component is the defect preprocessor, which normalizes each defect sample by positioning it at the origin, aligning it, and projecting it onto a plane. This standardization enables the defect surface combiner to process defects in a uniform format. As the defect preprocessor is dependent on defect type, it requires tailored implementation; here, we present the implementation for cracks. Conversely, the defect surface combiner operates independently of defect type. The full conceptual workflow is illustrated in Fig. 2.

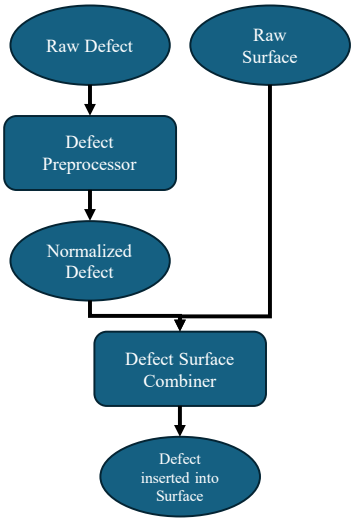

Fig. 2. Concept of the data augmentation for the defect detection task in point clouds

3.2. Defect Preprocessor

The goal of the preprocessor is to standardize each defect by shifting it to the origin, aligning it, and projecting it onto a plane. This process produces a generalized output that serves as a consistent input for the subsequent data augmentation algorithm, regardless of defect type. However, the preprocessor itself remains dependent on the specific defect type and must be individually implemented for each one.

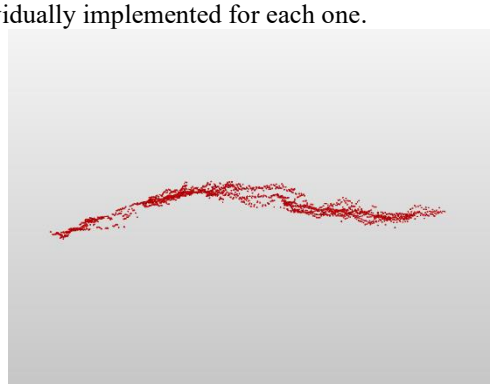

Fig. 3. Unprocessed crack sample

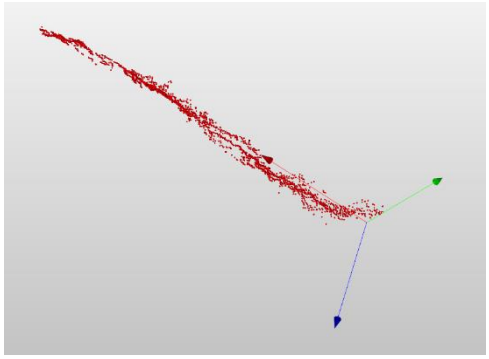

Fig. 4. Processed crack sample

The following section presents the implementation for the crack defect type, one of the most common defects in surface inspection tasks using white light interferometers [3]. Fig. 3 illustrates an unprocessed crack, while Fig. 4 shows the crack after processing by the defect preprocessor.

The first step shifts the crack data to the world origin, using the mean of the point cloud to center it. This simplification facilitates subsequent steps. Next, the point cloud is aligned with the global axes by applying eigenvector analysis. The eigenvector with the largest eigenvalue indicates the primary direction of the crack, and the points are rotated to align this eigenvector with the x-axis. Following this, the orientation of the crack normal, which matches the surface normal, is determined. Points with z-values exceeding the mean z-value by a specified threshold are selected as surface or near-surface points. A plane fit is then applied to these points, and the resulting normal vector is calculated. Finally, the crack points are rotated to align the z-axis with the computed surface normal.

Some cracks occur in curved regions, resulting in a curved crack profile that complicates seamless insertion into a flat surface. To correct this, a polynomial surface fit is applied to

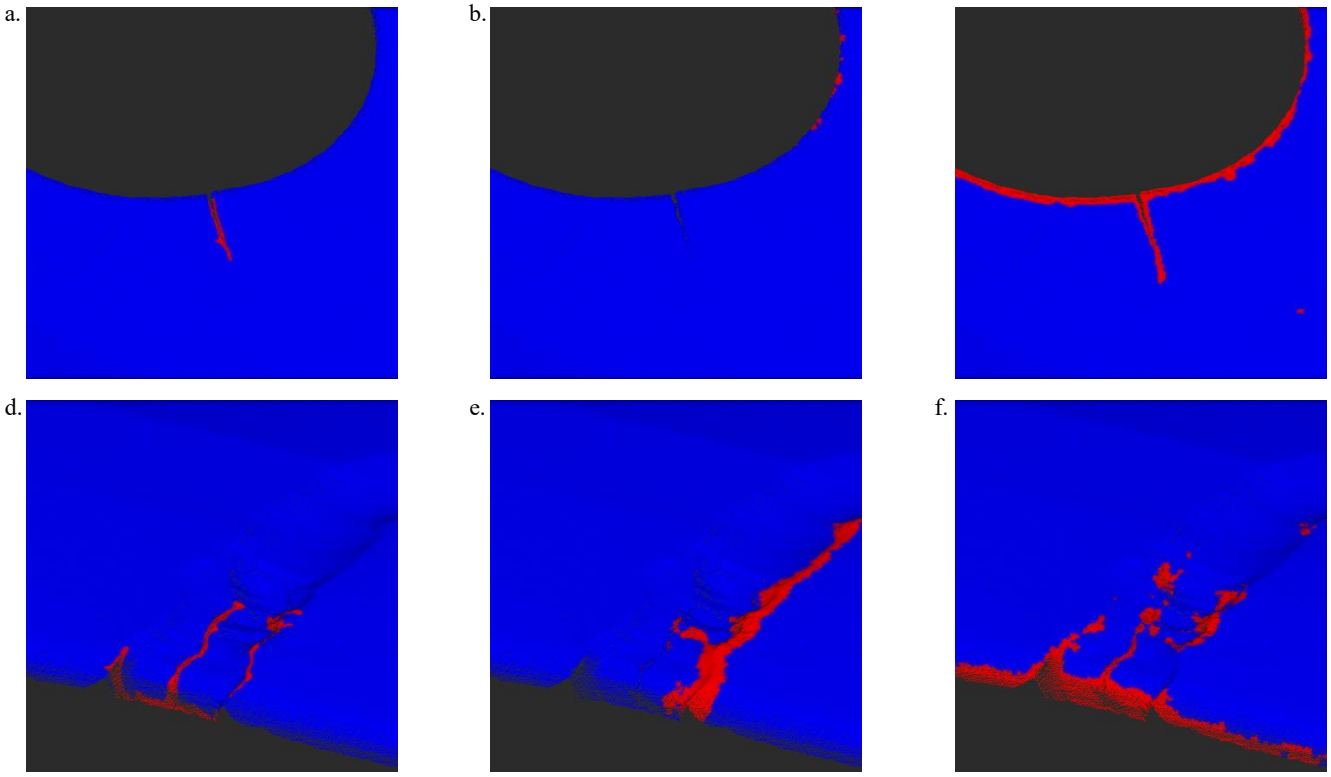

Fig. 5. Comparison of samples from the validation dataset for the different model. (a) Ground truth sample. (b) Sample from (a) evaluated by the model trained on the ground truth dataset. (c) Sample from (a) evaluated by the model trained on the augmented dataset. (d) Ground truth sample of a defect at an imprinted area. (e) Sample from (d) evaluated by the model trained on the ground truth dataset. (f) Sample from (d) evaluated by the model trained on the augmented dataset.

align the crack to the x-y plane. All crack points are used to generate this polynomial fit, and each point's z-coordinate is then adjusted by subtracting the corresponding z-value of the polynomial surface at that position.

The next step is estimating the crack origin, defined as the point where crack growth likely initiated. This position typically corresponds to the widest part of the crack. Points near the maximum and minimum x-values are selected, and the mean y-value of each set is calculated. Since the crack width is measured along the y-axis, the y-values of each set are compared to the mean y-value. The set with the greater y-axis deviation indicates the crack's starting point. Thus, the crack origin is defined by the largest or smallest x-value of the set with the greater y-axis deviation, the mean y-value of the corresponding set, and the average z-value of the initial points in the set.

The final step in the preprocessor is extracting the crack boundary, an important input for the subsequent data augmentation algorithm. To achieve this, the crack is sampled along the x-axis. At each sampling step, the points with the minimum and maximum value on the y-axis are identified, generating two arrays: one containing the points with the minimum y-values and the other with the maximum y-values. The boundary is defined by starting with the array of maximum y-values, ordered from the smallest to the largest x-value, and then continuing with the array of minimum y-values, ordered from the largest to the smallest x-value.

The output is a point cloud where the crack origin is positioned at the world origin, aligned with the global axes. The surface curvature has been corrected so that the crack lies within the x-y plane, and the boundary points have been identified. This processed data serves as one of the inputs for the defect surface combiner in the next stage.

3.3. Defect surface combiner

The data augmentation algorithm is detailed in this section, with the primary objective of seamlessly integrating a surface defect, such as a crack, into a given surface. The input consists of a point cloud representing the surface, along with the point cloud of the defect feature, which is aligned and transformed to a defined defect origin, including its boundary points.

Initially, surface normals are computed for the point cloud of the surface. This computation is performed only once for a given surface point cloud. If the same surface is later used in conjunction with a different defect feature, the normal computation step is not repeated.

A random point on the surface is selected as the seed for the defect, which is then randomly rotated about the z-axis. The defect's origin is shifted to the selected surface point. A region of interest (ROI) is defined around the defect, and points within this ROI are used to fit a polynomial surface. The defect points are subsequently shifted to align with the fitted surface. A k-nearest neighbors (k-NN) search is performed to identify points within or near the defect. The defect boundaries, estimated in a prior step, are converted into a polygon, and the inclusion of each surface point within this polygon is determined using the Jordan curve theorem. Surface points inside the polygon are transformed based on the neighboring defect points, with transformations weighted by the distance between the surface

point and the defect point, such that closer points exert a greater influence. Surface points within the defect that lack neighboring defect points are removed. This approach maintains the original data structure of the white light interferometer, ensuring that the defect is seamlessly integrated into the surface data. Finally, the labels of the surface points are updated to reflect their inclusion within the defect region. A representative result of this process is illustrated in Fig. 6.

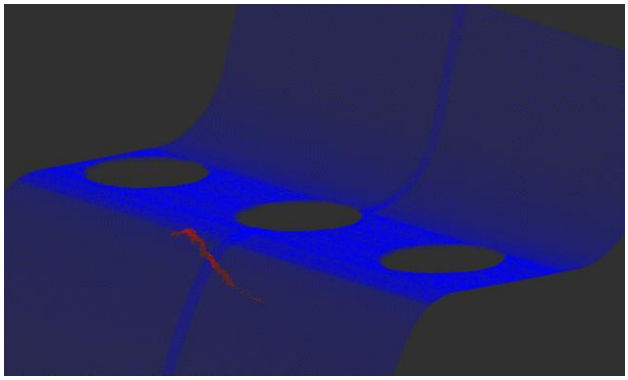

Fig. 6. Example of a point cloud of a surface with a crack acquired by a white light interferometer

4. Tests & Results

For testing purposes, a ground truth dataset comprising 54 samples of white light interferometer scans of real surface cracks is utilized. This dataset contains approximately 57 million points labeled as surface and 65,900 points labeled as cracks. From this dataset, 10 samples are reserved for the validation set.

The training data includes 36 labeled cracks from the ground truth dataset. For surface data, 100 defect-free surface samples are sourced from a separate dataset, also acquired using a white light interferometer. Each scan in this dataset contains approximately 80,000 points, with two to four scans being stitched together to create point clouds ranging from 119,000 to 329,000 points. The total number of points per sample varies due to structural features such as holes or edges.

Using the 100 surface samples and 36 cracks, a data augmentation process generates a total of 589 training samples. This augmented dataset ensures a diverse and representative sample set for training the model.

4.1. Training

To evaluate the proposed approach, the RandLa-Net model was employed due to its suitability for semantic segmentation of large point clouds. The model processes inputs of 80,000 points through five hidden layers, making it well-suited for this application. As no publicly available point cloud datasets closely aligned with surface inspection exist, the model was trained from scratch. A batch size of 2 was utilized to accommodate the high memory requirements inherent in processing point clouds. Hyperparameter tuning was conducted to optimize the learning rate, with values of 1e-1, 1e-2, and 1e-3 tested to identify the best configuration.

Two separate models were trained: one using the ground truth dataset and another using only the augmented dataset. The models were evaluated on the validation set after achieving their respective peak performances. The model trained on the ground truth dataset reached optimal performance after 120 epochs with a learning rate of 1e-3, achieving an overall validation accuracy of 82.1%, with a precision of 58.8% and recall of 77.14%. In comparison, the model trained on the augmented dataset achieved its best performance after 150 epochs with a learning rate of 1e-2, reaching a validation accuracy of 86.2%, with a precision of 52.8% and recall of 79.2%. These results demonstrate that the data augmentation approach enhances the model's performance.

Qualitative results, shown in Fig. 5, highlight the improvements provided by the augmented dataset. In particular, the edge case (Fig. 5 (d) – (f)) demonstrates that the model trained on the augmented dataset better learns the general characteristics of a crack, enabling improved segmentation of previously unseen defects compared to the model trained on the ground truth dataset alone.

5. Conclusion and outlook

A data augmentation algorithm was developed and implemented to enhance surface inspection capabilities using a white light interferometer. The method involves embedding a normalized point cloud representing a defect into a defect-free surface point cloud while preserving the sensor-specific data structure.

The algorithm was evaluated in the context of crack detection on metal surfaces, demonstrating that data augmentation introduces greater variation to the training data, thereby improving model performance. Future work will focus on extending this approach to other defect types, general surface features and integrating it into a training data generation pipeline to address issues like the domain gap or unavailable data for certain defect variation. A key limitation of the method is the dependency on sufficient variability in defect data, as some defect types are rare in practical scenarios. To overcome this limitation, the integration of synthetic data generation will be investigated to supplement the dataset with additional defect variations, thereby ensuring robust model training.

CRediT author statement

Daan Büchner: Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Ole Schmedemann:** Writing - Review & Editing. **Thorsten Schüppstuhl:** Supervision

Acknowledgements

This research was funded by the German Federal Ministry for Economic Affairs and Climate Action under grant number 20D1912B.

References

[1] R. Saltoğlu, N. Humaira und G. İnalhan, „Aircraft Scheduled Airframe

Maintenance and Downtime Integrated Cost Model," *Advances in Operations Research,* Bd. 2016, p. 1–12, 2016.

[2] T. Domaschke, „Automatisierung der Weißlichtinterferometrie zur Inspektion rotationssymmetrischer Triebwerksbauteile," Hamburg, 2017.

[3] M.-A. Otto, „Automated High-Precision Crack Detection in Airplane Combustion Chamber Liners Using White Light Interferometry," Siegen, 2020.

[4] O. Schmedemann, S. Schlodinski, D. Holst und T. Schüppstuhl, „Adapting synthetic training data in deep learning-based visual surface inspection to improve transferability of simulations to real-world environments," in *Automated Visual Inspection and Machine Vision V*, 2023.

[5] O. Schmedemann, M. Baaß, D. Schoepflin und T. Schüppstuhl, „Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection," *Procedia CIRP,* Bd. 107, p. 1101–1106, 2022.

[6] A. Mumuni und F. Mumuni, „Data augmentation: A comprehensive survey of modern approaches," *Array,* Bd. 16, p. 100258, December 2022.

[7] Q. Zhu, L. Fan und N. Weng, „Advancements in point cloud data augmentation for deep learning: A survey," *Pattern Recognition,* Bd. 153, p. 110532, September 2024.

[8] P. Lafiosca, I.-S. Fan und N. P. Avdelidis, „Automated Aircraft Dent Inspection via a Modified Fourier Transform Profilometry Algorithm," *Sensors (Basel, Switzerland),* Bd. 22, 2022.

[9] E. T. Lee, Z. Fan und B. Sencer, „A new approach to detect surface defects from 3D point cloud data with surface normal Gabor filter (SNGF)," *Journal of Manufacturing Processes,* Bd. 92, p. 196–205, April 2023.

[10] I. Jovančević, H.-H. Pham, J.-J. Orteu, R. Gilblas, J. Harvent, X. Maurice und L. Brèthes, „3D Point Cloud Analysis for Detection and Characterization of Defects on Airplane Exterior Surface," *Journal of Nondestructive Evaluation,* Bd. 36, p. 1–17, 2017.

[11] M. Makuch und P. Gawronek, „3D Point Cloud Analysis for Damage Detection on Hyperboloid Cooling Tower Shells," *Remote Sensing,* Bd. 12, p. 1542, 2020.

[12] L. Huo, Y. Liu, Y. Yang, Z. Zhuang und M. Sun, „Review: Research on product surface quality inspection technology based on 3D point cloud," *Advances in Mechanical Engineering,* Bd. 15, p. 168781322311595, March 2023.

[13] A. Zhang, K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li und B. Li, „Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet," *Journal of Computing in Civil Engineering,* Bd. 32, p. 04018041, 2018.

[14] L. Zhao, F. Li, Y. Zhang, X. Xu, H. Xiao und Y. Feng, „A Deep-Learning-based 3D Defect Quantitative Inspection System in CC Products Surface," *Sensors (Basel, Switzerland),* Bd. 20, 2020.

[15] P. Lafiosca, I.-S. Fan und N. P. Avdelidis, *Automatic Segmentation of Aircraft Dents in Point Clouds (SAE Paper 2022-01-0022),* 2022.

[16] Y. Wang, W. Sun, J. (. Jin, Z. (. Kong und X. Yue, „MVGCN: Multi-View Graph Convolutional Neural Network for Surface Defect Identification Using Three-Dimensional Point Cloud," *Journal of Manufacturing Science and Engineering,* Bd. 145, December 2022.

[17] C. R. Qi, H. Su, K. Mo und L. J. Guibas, *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,* 2016.

[18] C. R. Qi, L. Yi, H. Su und L. J. Guibas, „PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems*, 2017.

[19] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny und B. Ghanem, „PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies," in *Advances in Neural Information Processing Systems*, 2022.

[20] S. Xiong, B. Li und S. Zhu, „DCGNN: a single-stage 3D object detection network based on density clustering and graph neural network," *Complex & Intelligent Systems,* Bd. 9, p. 3399–3408, December 2022.

[21] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette und L. J. Guibas, *KPConv: Flexible and Deformable Convolution for Point Clouds,* 2019.

[22] W. Wu, Z. Qi und L. Fuxin, „PointConv: Deep Convolutional Networks on 3D Point Clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] H. Zhao, L. Jiang, J. Jia, P. Torr und V. Koltun, *Point Transformer,* 2020.

[24] X. Ma, C. Qin, H. You, H. Ran und Y. Fu, *Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework,* arXiv, 2022.

[25] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni und A. Markham, „RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, p. 11108–11117.

[26] S. V. Sheshappanavar, V. V. Singh und C. Kambhamettu, „PatchAugment: Local Neighborhood Augmentation in Point Cloud Classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2021.

[27] J. Choi, Y. Song und N. Kwak, „Part-Aware Data Augmentation for 3D Object Detection in Point Cloud," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[28] S. Kim, S. Lee, D. Hwang, J. Lee, S. J. Hwang und H. J. Kim, *Point Cloud Augmentation with Weighted Local Transformations,* arXiv, 2021.

[29] Y. Wang, J. Wang, J. Li, Z. Zhao, G. Chen, A. Liu und P. A. Heng, „PointPatchMix: Point Cloud Mixing with Patch Scoring," *Proceedings of the AAAI Conference on Artificial Intelligence,* Bd. 38, p. 5686–5694, March 2024.

[30] P. Shi, H. Qi, Z. Liu und A. Yang, „Context-guided ground truth sampling for multi-modality data augmentation in autonomous driving," *IET Intelligent Transport Systems,* Bd. 17, p. 463–473, September 2022.

[31] J. Zhang, L. Chen, B. Ouyang, B. Liu, J. Zhu, Y. Chen, Y. Meng und D. Wu, „PointCutMix: Regularization strategy for point cloud classification," *Neurocomputing,* Bd. 505, p. 58–67, September 2022.

[32] Z. Leng, S. Cheng, B. Caine, W. Wang, X. Zhang, J. Shlens, M. Tan und D. Anguelov, „PseudoAugment: Learning to Use Unlabeled Data for Data Augmentation in Point Clouds," *ECCV 2022 (pp. 555-572). Springer, Cham,* p. 555–572, October 2022.

[33] L. Reichardt, L. Uhr und O. Wasenmüller, *Text3DAug – Prompted Instance Augmentation for LiDAR Perception,* arXiv, 2024.

[34] Y. Chen, V. T. Hu, E. Gavves, T. Mensink, P. Mettes, P. Yang und C. G. M. Snoek, *PointMixup: Augmentation for Point Clouds,* arXiv, 2020.

[35] T.-A. Vu, S. Sarkar, Z. Zhang, B.-S. Hua und S.-K. Yeung, „Test-Time Augmentation for 3D Point Cloud Classification and Segmentation," in *2024 International Conference on 3D Vision (3DV)*, 2024.

[36] C. Zhang, A. Li, D. Zhang und C. Lv, „PCAlign: a general data augmentation framework for point clouds," *Scientific Reports,* Bd. 14, September 2024.

[37] A. Xiao, J. Huang, D. Guan, K. Cui, S. Lu und L. Shao, „PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds," in *Advances in Neural Information Processing Systems*, 2022.

[38] S. Qiu, J. Chen, C. Lai, H. Lu, X. Xue und J. Pu, „Leveraging Smooth Deformation Augmentation for LiDAR Point Cloud Semantic Segmentation," *IEEE Transactions on Intelligent Vehicles,* Bd. 9, p. 3316–3329, February 2024.

[39] F. Hasecke, M. Alsfasser und A. Kummert, „What Can be Seen is What You Get: Structure Aware Point Cloud Augmentation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022.

[40] S. Lin, C. Xu, L. Chen, S. Li und X. Tu, „LiDAR Point Cloud Recognition of Overhead Catenary System with Deep Learning," *Sensors (Basel, Switzerland),* Bd. 20, 2020.