Defect Characterization on Complex Shape Aeronautical Parts via 3D Point Cloud Processing

Marija Džaković, Igor Jovančević, Velibor Došljak, Jean-José Orteu

Defect Characterization on Complex Shape Aeronautical Parts via 3D Point Cloud Processing

Marija Džaković
Faculty of Natural Sciences and Mathematics
University of Montenegro
Cetinjska 2, 81000 Podgorica, Montenegro
Institut Clément Ader (ICA); Université de Toulouse;
CNRS, IMT Mines Albi, INSA, UPS, ISAE
Campus Jarlard, 81013 Albi, France
marijapgd@gmail.com

Igor Jovančević
Faculty of Natural Sciences and Mathematics
University of Montenegro
Cetinjska 2, 81000 Podgorica, Montenegro
igorj@ucg.ac.me

Velibor Došljak
Faculty of Natural Sciences and Mathematics
University of Montenegro
Cetinjska 2, 81000 Podgorica, Montenegro
Institut Clément Ader (ICA); Université de Toulouse;
CNRS, IMT Mines Albi, INSA, UPS, ISAE
Campus Jarlard, 81013 Albi, France
velibord@ucg.ac.me

Jean-José Orteu
Institut Clément Ader (ICA); Université de Toulouse;
CNRS, IMT Mines Albi, INSA, UPS, ISAE
Campus Jarlard, 81013 Albi, France
jean-jose.orteu@mines-albi.fr

Abstract—**This paper presents an approach to characterizing defects - a process that consists of accurate measurement of their geometric properties such as depth and surface area, assuming that the defect detection process has been successfully performed previously. Our methodology for addressing this problem involves three key steps. The first is to reconstruct an ideal or defect-free surface using scattered points obtained from a point cloud scan of the inspected part. Subsequently, the distance from each cloud point to this surface is computed, and points at a distance higher than a specified threshold are identified as defect points. The maximum of these distances corresponds to the depth of the defect. Finally, the minimal 3D bounding box encapsulating the defect points is determined, where the two largest dimensions of this box represent the length and width of the identified defect.**

***Index Terms*—3D point clouds, computer vision, defect characterization, ideal surface approximation**

I. Introduction

Detriment on the surface of an industrial product or a structure can indicate a serious issue regarding its safety and structural integrity. Timely detection and accurate measurements of the geometric parameters of present damages, such as depth, width, and length, provide an approximate prediction of the remaining lifespan of the structure and can help in mitigating associated safety risks. This process can be automated using autonomous surveying robots with built-in real-time software. Hence, the aircraft maintenance, repair, and overhaul (MRO) industry is progressively adopting 3D scanning for dent inspection, replacing the more error-prone and time-consuming manual control methods.

The proposed methodology strongly relies on [2], with the advantage of a bigger evaluation dataset and alteration in the application of a more suitable surface reconstruction method for more complex surfaces.

Our work is seen as a continuation of the DECADOM project (2020-2023) [1] which dealt with detecting and characterizing defects on mechanical surfaces using 2D/3D vision. The authors focused on the application of their proposed software in the field of aeronautics. They claim that it is essential to perform periodic inspections, whether on structural parts before the assembling phase, on assembled sections before delivery, or a pre-flight inspection of an aircraft in service.

In [5] T. Reyno, C. Marsden and D. Wowk propose a 3D scanning method for measuring surface damage on aircraft structural panels. Dent depths were quantified by the difference between a point cloud rendering of the damaged surface and a surface fit approximating the original, undamaged surface.

Both their approach and ours present the novelty of recreating models of both the damaged and undamaged parts without any pre-existing Computer-Aided Design (CAD) information. They used image processing to measure dent length and area, and the obtained results showed that their method is more efficient and reliable compared to manual methods.

In [8] the only OEM-certified 'Go/No-Go' tool for 3D dent mapping is introduced. dentCHECK measures and analyzes

dents, bumps, lightning strikes, and blend-outs on metallic, composite, curved, and flat surfaces. All results are instantly projected (superimposed) on the aircraft surface.

II. Problem Statement and Available Data

In this work, the focus is on damage characterization, with the assumption that the defect detection module previously identified and localized defects on the inspected surfaces, particularly mechanical components of aircraft, made of metal or composite materials. Characterization is performed by processing the 3D point clouds, utilizing traditional methods of computer vision. The software aims to characterize identified dents or cracks by assessing their volume (depth, width, and length). The principal goal is precision with a mean absolute error smaller than 0.2mm for depth estimation.

We used a dataset that was collected during the DECADOM project [1]. The 3D scanner used in point cloud acquisition is 'Artec Space Spider', it was manually carried by an operator or by a cobot (automated control). 'Artec Space Spider' is a high-resolution 3D scanner based on blue light technology. It is perfect for capturing small objects or intricate details of large industrial objects in high resolution, with steadfast accuracy and brilliant color.

In total, there are 29 defects on scanned surfaces (see Fig. 1).

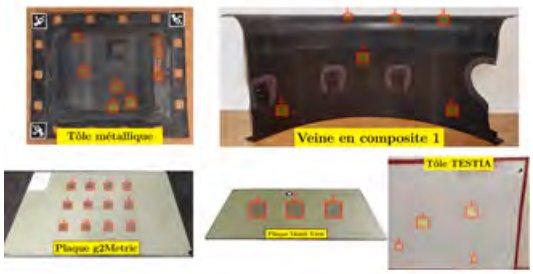


Fig. 1: Aircraft mechanical parts with annotated defects.

The mechanical parts exhibit various shapes. Some surfaces are planar, others are slightly curved, while certain parts, from the metal sheet component, have more complex geometric features (see Fig. 2a). Those complex shapes require a different approach in the surface reconstruction phase.

III. Overview of the Proposed System

Our method can be divided into four steps:

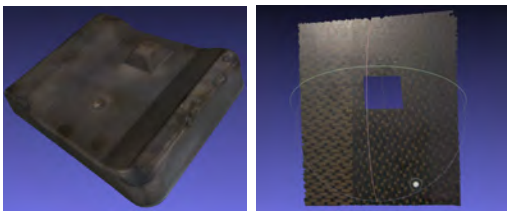
1) Pre-processing point clouds - down-sampling and outlier removal
2) Ideal surface reconstruction
3) Defects' depth calculation
4) Defects' surface area calculation

The details of each step will be explained in the subsequent subsections.

In the implementation, we extensively used well-documented and light-weight Python libraries: Open3D [7] for point cloud processing, Numpy [10], Matplotlib [11] and SciPy [9].

A. Pre-processing

As we aim to determine the equation of the ideal surface upon which the defect lies, it is crucial to exclude points belonging to the defect from this process. Hence, the code for surface reconstruction utilized point clouds from which the defect region had been removed, as illustrated in Fig. 2b.



(a) (b)

Fig. 2: (a) Metal sheet mechanical part, (b) Example of a point cloud with removed defect points.

We have noticed that in the cases where the underlying surface is planar, the residual between the fitted ideal surface and the used points was small and tolerable. However, when dealing with curved surfaces, integrating points within a large area around the defect into the approximation process results in a notable rise in the residual (by approximately 20 to 30 millimeters). This happens because the scanned surface is not always perfectly resembling a surface with a (quadratic) polynomial equation. To improve the approximation and enhance the accuracy of measurements, we opted to utilize points within a smaller vicinity around the defect in the surface reconstruction of highly curved surfaces found on a particular mechanical component.

We used statistical outlier removal functionality from Open3D [7]. After removing outliers, point clouds underwent down-sampling through the uniform down-sampling function available in the Open3D library [7]. Down-sampling and outlier removal parameters were found experimentally. These parameters remain constant across all point clouds processed with the same surface reconstruction method, but they are adjusted for each method in order to get best results.

B. Surface Reconstruction

Since CAD models of inspected parts are not available, we need to reconstruct a surface approximating the original undamaged geometry of the part. In other words, we want to create a virtual model of the part. Further, we can compare point cloud data from the in-service, damaged part with the reconstructed virtual model. Measuring the variance between the original and damaged geometries results in an assessment of the defect's depth.

A vast amount of research literature is available on the methods for surface reconstruction from 3D point clouds. Reference [6] provides a comprehensive examination of these methods. The authors classify a large number of methods according to various factors such as the characteristics of input

data, referred to as point cloud artifacts, input requirements, and the form of output representation. They categorize them by considering conventional surface smoothness priors as well as more specialized priors.

As specified, the outcome of reconstruction algorithms is significantly influenced by the characteristics of the input point clouds, such as sampling density, noise level, outliers, and missing data. In our case, the most challenging property of point clouds is missing data, since we extract the defect from the point cloud, which leaves us with no points in that particular region.

1) Weighted Least Squares: For all point clouds except for defects on the metal sheet, Weighted Least squares approximation [3] was used. This method proves suitable as it prioritizes points around defects, which is preferable for our specific needs. By utilizing the given points, it reconstructs the surface by inferring geometric properties in regions lacking data. After outlier removal and down-sampling we have N points $p_i = (x_i, y_i, z_i)$, $i \in \{1 \dots N\}$ located at certain positions in R^3 space.

We wish to obtain a globally defined, quadratic bivariate polynomial $f : R^2 \to R$, defined in (2), that best approximates the cloud points. The goal is to generate a function such that the distance between the scalar data values z_i and the function evaluated at the points $f(x_i, y_i)$ is as small as possible, with the error functional

$$\sum_{f \in \Pi_m^d} \theta\left(\|\bar{p} - p_i\|\right) \cdot \|f\left(x_i, y_i\right) - z_i\|^2 \tag{1}$$

Where $\bar{p}$ is a fixed point in R^3, in our case, the mean value of all defect points. The error is weighted by $\theta(d_i)$ where d_i are the Euclidian distances between $\bar{p}$ and the positions of data points p_i.

The equation of quadratic bivariate function f is:

$$f(x, y) = a \cdot x^2 + b \cdot y^2 + c \cdot x \cdot y + d \cdot x + e \cdot y + f \tag{2}$$

We used this polynomial because it performs well in both cases: when the input point cloud resembles a plane or a quadratic surface. In the first case, coefficients a, b, and c are close to zero. Hence, no apriori knowledge about the shape of the surface is needed. Moreover, the results that we got by fitting the linear function and quadratic function, show that a smaller residual is obtained in the second case. A plane fit is insufficient because, in general, the surface of most parts is not entirely flat.

For non-negative weighting function θ we chose $\theta(d) = \frac{1}{d^2+\epsilon^2}$, where ϵ is a constant, set to 0.15. The weighting functions have the property of assigning higher weights to the points that are in the proximity of the defect, and lower weights to the points far away from it.

2) Poisson surface reconstruction: Since the metal sheet part has a more complex shape, with a non-uniform profile (see Fig. 2a), the surface can not be approximated with a quadratic model or another small degree polynomial surface. For these point clouds we used method *create_from_point_cloud_poisson* from Open3D library [7] which is a wrapper of Kazhdan et al. Poisson surface reconstruction algorithm [4]. Poisson surface reconstruction is chosen due to its versatility in reconstructing various surface types, its speed, and its integration into the Open3d library. For this method, we found that the outlier removal did not yield improved results. Therefore, we chose not to employ it.

Following the generation of the mesh using points from the undamaged part of the surface we need to find the distances from defect cloud points to it.

C. Calculating depth of defects

The next step involves calculating the distances between the defect points in the point cloud being analyzed and the surface. To achieve this, the cropped defect region was used. Note that the defect point cloud was first cropped and then down-sampled by keeping every fifth point, ensuring that the execution time remains reasonable.

To determine the distances, following the reconstruction of the surface using WLS, we used the Lagrange multipliers method. We implemented it using the SciPy library [9].

Looping through every point $p = (p_x, p_y, p_z)$ in defect point cloud, the objective was to find the minimum of squared distance function $f(x, y, z) = (x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2$ subject to the constraint given by the implicit quadratic surface equation.

In the case of Poisson surface reconstruction, we used *RaycastingScene* class from Open3D, which provides *compute_distance* method for determining the distances between defect cloud points to the generated mesh. The *compute_distance* method converts the mesh to an implicit function representation to efficiently determine distances from points to the mesh surface without explicitly iterating over all mesh vertices or faces.

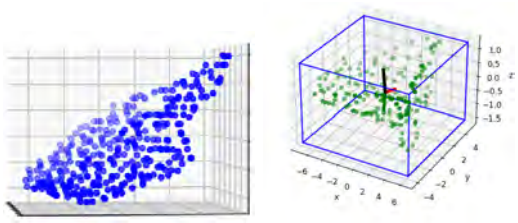
Every point that has a distance greater than the predefined threshold value, is appended to the set of defect points, and its distance to the set of distances. The maximum distance of all distances is the depth of the defect. Points representing defect were illustrated in Fig. 3a. After determining which points belong to the defect we proceed as described further.

D. Width and length of defects

The Boeing 737-400 SRM, here taken as reference, provides the only accepted definition of dent's measures: "The length of the dent is the longest distance from one end to the other end. The width of the dent is the second longest distance across the dent, measured at 90 degrees to the direction of the length". In our approach, we adhere to this particular definition.

In geometry, the minimum bounding box (MBB) for a point set S in N dimensions is the box with the smallest measure (area, volume, or hypervolume in higher dimensions) within which all the points lie (see Fig. 3b).

To determine the dimensions of the defect, namely its width and length, (since the depth is already calculated) we decided to enclose it within a minimal 3D bounding box. The two largest dimensions of this box represent the length and width of the inspected defect, respectively. To accomplish this, we utilized the Principal Component Analysis (PCA) algorithm. The bounding box is only created in the final step for the purpose of visualization (see Fig. 3b).



(a) (b)

Fig. 3: (a) plotted defect points, b) Defect points translated and rotated to align with Cartesian basis vectors and MBB around them.

TABLE I: Results

Defects	Threshold	Our method		Ground truth	
		w x l	*depth*	*w x l*	*depth*
D1	0.5	8.13×12.75	3.58	8.10×12.75	3.5
D2	0.5	6.55×12.97	1.17	2.40×7.01	1.51
D3	0.5	3.17×9.71	0.7	3.13×9.98	0.64
D4	0.5	9.41×13.15	3.13	9.41×13.15	3.13
D5	0.5	15.47×15.87	1.21	14.78×15.28	1.21
D6	0.5	13.35×14.04	2.59	22.52×22.71	1.27
D7	0.3	6.58×7.26	0.38	8.13×8.68	0.31
D8	0.3	10.15×10.66	0.55	10.36×10.51	0.43
D9	0.5	11.09×11.19	1.08	9.73×10.07	0.92
D10	0.5	25.29×26.47	1.49	25.17×25.89	1.61
D11	0.5	28.35×29.24	2.14	26.27×30.52	1.98
D12	0.2	7.03×10.24	0.31	7.54×11.12	0.42
D13	0.2	9.65×10.43	0.47	13.38×14.94	0.53
D14	0.15	17.47×17.56	0.34	$21,41 \times 22.75$	0.24
D15	0.1	12.36×14.27	0.2	12.35×12.62	0.2
D16	0.2	12.34×12.62	0.43	$12,62 \times 12,88$	0.37
D17	0.15	9.45×9.65	0.28	$9,78 \times 9,94$	0.29
D18	0.3	18.72×18.96	0.62	19.14×19.40	0.53
D19	0.3	16.05×16.14	0.47	16.56×16.69	0.41
D20	0.3	13.38×13.4	0.44	13.80×13.91	0.40
D21	0.1	5.98×10.28	0.18	$6,37 \times 10,47$	0.23
D22	0.5	17.32×18.14	0.64	18.64×18.84	0.66
D23	0.5	15.84×22.86	1.47	$16,26 \times 19,50$	1.52
D24	0.2	13.64×17.11	0.61	$13,45 \times 18,08$	0.73
D25	0.2	6.63×13.04	0.57	9.41×13.15	0.55
D26	0.5	7.45×8.58	0.76	4.47×5.91	0.55
D27	0.5	2.53×5.99	0.69	3.33×4.00	0.48
D28	0.5	8.01×9.63	1.77	8.60×9.21	1.26
D29	0.5	17.03×17.5	2.25	17.66×17.75	1.99

IV. Results

To assess the method's accuracy, we conducted a comparison between the results we obtained and manually acquired measures of defects by an operator during the DECADOM project, which we considered as ground truth. Table I displays results, listing measures of defects, including width, length (denoted as $w \times l$), and depth in millimeters.

The mean absolute error was calculated to estimate the accuracy of the method, yielding a value of 0.15 for the depth dimension, which falls below the specified margin, a value of 1.36 for the width, and 1.55 for the length.

The best results were achieved when dealing with planar surfaces featuring sphere-like defects as well as slightly curved surfaces with dents. However, estimations on highly curved and complex parts may have lower accuracy.

Conclusion

We propose a method that has the following advantages:

1) it provides a robust framework that can extract precise information about the defects' measures;
2) characterizes various types of defects without any prior knowledge about the surface shape or size of the defect;
3) the estimation of depth as the most important property of defects has the highest accuracy
4) it is not limited to aircraft parts, but can be used on other damaged surfaces

Through experimentation with the mentioned dataset, we have noticed which surface reconstruction method performs best with certain properties of input point clouds. For nearly flat surfaces, utilizing a broader defect environment yields the best results using WLS. Conversely, when applying WLS on curved quadratic models, a smaller number of neighboring points around the defect should be used for the most accurate approximation. Employing Poisson reconstruction with a wider point of view is preferable for highly curved, complex surfaces that deviate significantly from polynomial shapes. This approach allows Poisson reconstruction to effectively learn the surface characteristics necessary for reconstructing missing sections.

The parameter with the highest sensitivity in the proposed methodology is the threshold value utilized to differentiate defect points from surface points. This parameter significantly influences the estimation of the defect's width and length, as lowering it can result in the identification of more points as defects while raising it yields the opposite effect. Additionally, its effectiveness is influenced by the quality of the surface reconstruction process. For Poisson surface reconstruction the size of the point cloud is the most influential tunable factor. Furthermore, while Poisson reconstruction is known for its speed and applicability to various surface types, it is sensitive to the presence of missing data.

Further work

The selection of the surface reconstruction method should be automated, by examining the residual. If the fitted surface using WLS results in a residual higher than a certain threshold, then the Poisson reconstruction should be applied.
Additionally, determining the threshold that distinguishes defect points from surface points should be automated. This step is critical, as the distance threshold parameter significantly influences the estimation of the defect's width and length.

Acknowledgment

This work is a continuation of the DECADOM [1] project (2020-2023) which dealt with the detection and characterization of defects on mechanical structures by using 3D vision. The

DECADOM research work has been carried out within the framework of the joint research laboratory ”Inspection 4.0” between IMT Mines Albi/ICA and the company *Diota* specialized in the development of numerical tools for Industry 4.0. The DECADOM project was financially supported by the French “Région Occitanie”. The authors express their gratitude to the members of the DECADOM project and to French CETIM Sud-Ouest for providing the data for this research work.

The work was partially supported by Erasmus+ Project No. 2022-1-PL01-KA220-HED-000088359 entitled ”The Future is in Applied Artificial Intelligence” (FAAI) [12], which aims to join together Higher Education Institutions (HEI) and businesses. In this context, this project has to bridge the current artificial intelligence (AI) skills gap, build an AAI ecosystem of key partners, promote AI business opportunities, and support the creation of internship programs in AI. The FAAI project activities focus on HEI trainers, undergraduate and postgraduate students, and business managers. Furthermore, the project is promoting among businesses and young people the enormous opportunities provided by AI to build the ecosphere of modern society. The given work was performed within the framework of the FAAI work package 4 entitled ”Artificial Intelligence framework for training in HE” and presents a real use case that is offered for studying applied AI.

REFERENCES

[1] Zakaria Belbacha, Hamdi Ben Abdallah, Igor Jovančević, Jean-José Orteu, Ludovic Brethes. Detection and characterization of defects on mechanical structures by using 3D vision. QCAV’2023 - the 16th international conference on quality control by artificial vision, Jun 2023, Albi, France. 7 p. https://doi.org/10.1117/12.3000509/. https://hal.science/hal-04168243/.

[2] Igor Jovančević, Huy-Hieu Pham, Jean-José Orteu, Rémi Gilblas, Jacques Harvent, et al.. 3D Point Cloud Analysis for Detection and Characterization of Defects on Airplane Exterior Surface. Journal of Nondestructive Evaluation, 2017, 36 (4), pp.74. https://doi.org/10.1007/s10921-017-0453-1/. https://hal.science/hal-01622056/

[3] Nealen, Andrew. (2004). An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation.

[4] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing (SGP ’06). Eurographics Association, Goslar, DEU, 61–70.

[5] T. Reyno, C. Marsden, D. Wowk, Surface damage evaluation of honeycomb sandwich aircraft panels using 3D scanning technology, NDT & E International, Volume 97, 2018, Pages 11-19, ISSN 0963-8695,

[6] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Gael Guennebaud, et al.. A Survey of Surface Reconstruction from Point Clouds. Computer Graphics Forum, 2016, pp.27. https://doi.org/10.1111/cgf.12802. https://hal.science/hal-01348404v2.

[7] Zhou, Q., Park, J., & Koltun, V. (2018). Open3D: A Modern Library for 3D Data Processing. ArXiv. /abs/1801.09847

[8] 4D Technology, DentCheck;‘Go/No-Go’ Dent & Bump Mapping with real-time Visual Feedback, august, 2023, https://www.8-tree.com/dentcheck/

[9] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, CJ Carey, İ. Polat, Yu Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, Ch. R Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272

[10] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2.

[11] J. D. Hunter, ”Matplotlib: A 2D Graphics Environment”, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[12] The Future is in Applied Artificial Intelligence (FAAI). (2022-2024), https://faai.ath.edu.pl.