3D Point Cloud Analysis for Detection and Characterization of Defects on Airplane Exterior Surface

Igor Jovančević, Huy-Hieu Pham, Jean-José Orteu, Rémi Gilblas, Jacques Harvent, Xavier Maurice, Ludovic Brèthes

3D point cloud analysis for detection and characterization of defects on airplane exterior surface

Igor Jovančević[1] · Huy-Hieu Pham[1] · Jean-José Orteu[1] · Rémi Gilblas[1] · Jacques Harvent[1] · Xavier Maurice[2] · Ludovic Brèthes[2]

Abstract Three-dimensional surface defect inspection remains a challenging task. This paper describes a novel automatic vision-based inspection system that is capable of detecting and characterizing defects on an airplane exterior surface. By analyzing 3D data collected with a 3D scanner, our method aims to identify and extract the information about the undesired defects such as dents, protrusions or scratches based on local surface properties. Surface dents and protrusions are identified as the deviations from an ideal, smooth surface. Given an unorganized point cloud, we first smooth noisy data by using Moving Least Squares algorithm. The curvature and normal information are then estimated at every point in the input data. As a next step, Region Growing segmentation algorithm divides the point cloud into defective and non-defective regions using the local normal and curvature information. Further, the convex hull around each defective region is calculated in order to englobe the suspicious irregularity. Finally, we use our new technique to measure the dimension, depth, and orientation of the defects. We tested and validated our novel approach on real aircraft data obtained from an Airbus A320, for different types of defect. The accuracy of the system is evaluated by comparing the measurements of our approach with ground truth measurements obtained by a high-accuracy measuring device. The result shows that our work is robust, effective and promising for industrial applications.

✉ Igor Jovančević
igorjovan@gmail.com

[1] Université de Toulouse; CNRS, INSA, UPS, Mines Albi, ISAE; Institut Clément Ader (ICA);
Campus Jarlard, F-81013 Albi, France.

[2] KEONYS, 5 avenue de l'escadrille Normandie-Niemen, 31700 Blagnac, France.

List of symbols

In this paper, we use the following notation:

$P_N = \{p_1, p_2, ..., p_N\}$	A set of N points, p_i is the ith data point.
$p_i = (x_i, y_i, z_i)$	A point in three-dimensional space.
$P^K = p_1, p_2, ..., p_K$	The set of points which are located in the k-neighborhood of a query point p_i.
$\bar{p}$	The centroid of the data e.g., given a set of points P_N, we have: $\bar{p} = \frac{1}{N}(\sum\limits_{i=1}^{N} x_i, \sum\limits_{i=1}^{N} y_i, \sum\limits_{i=1}^{N} z_i)$
n_i	A surface normal estimated at a point p_i.
$\cdot$	The dot product.
$\times$	The cross product.
$\| \circ \|$	The Euclidean norm of $\circ$.

1 Introduction

In the aviation industry, one of the most important maintenance tasks is aircraft surface inspection. The main purpose of fuselage inspection process is to detect the undesired defects such as dents, protrusions or cracks. This is a difficult task for a human operator, especially when dealing with small defects hardly or not at all visible to the naked eye. In order to speed-up the inspection process and reduce human error, a multi-partners research project is being carried on to develop a collaborative mobile robot named Air-Cobot, with integrated automatic vision-based aircraft inspection capabilities.

Currently, coordinate measuring machines (CMMs) are widely used in the field of three-dimensional (3D) inspection. However, the inspection systems based on CMM machines have extremely low scanning speed; these systems are not suitable for working with the large objects such as an airplane. Instead, the recent advances of laser scanning technologies now allow the development of new devices to acquire the 3D data. Various types of 3D scanner have been developed for the inspection applications and the use of laser sensors in 3D part measurement process has introduced a significant improvement in data acquisition process regarding time and cost [18]. Therefore, Air-Cobot uses a 3D scanner that is capable of collecting point cloud within a short time at high rate of accuracy and under different illumination conditions. In order to get infor-

mation about the airplane exterior surface, we need to develop a robust inspecting technique for processing the scanned point cloud data.

In this paper, we present a robust approach for detecting and characterizing undesired deformation structures from 3D data. It mainly consists of two processes: detection process and characterization process. Firstly, the point cloud is pre-processed to remove measurement errors and outliers. The proposed approach then analyses the point cloud for identifying the defects and their positions. For this purpose, we focus on developing a segmentation algorithm in which the defect regions are segmented based on local features including local curvature and normal information. After isolating the defective regions, they are analyzed to find their dimensions, depths and orientations.

Our proposed method has the following advantages: (1) provides a robust framework which is able to detect and extract detailed information about the defects; (2) detects various types of defects without any prior knowledge of size or shape; (3) fully automates inspection process.

The rest of the paper is organized as follows: Sect. 2 contains a review of the related work. The dataset, context, and our approach are explained in Sect. 3. Sect. 4 shows a few empirical experiments of the proposed approach and discusses about experimental results. Finally, in Sect. 5, some future directions are presented and the paper is concluded.

2 Related work

Over the last few decades, visual inspection has received a great interest from the aviation industry. The majority of the existing systems have been developed for aircraft surface inspection. For instance, C. Seher et al. [46] have developed a prototype robot for non-destructive inspection (NDI) based on 3-D stereoscopic camera. M. Siegel et al. [48,49] have introduced the surface crack detection algorithm for aircraft skin inspection. This algorithm is based on determining region of interest (ROI) and edge detection technique. B. S. Wong et al. [57] have also developed an algorithm based on ROI and edge detection, but using a digital X-ray sensor. R. Mumtaz et al. [32] proposed a new image processing technique using neural network for classifying crack and scratch on the body of the aircraft. Wang et al. [54] developed a mobile platform for aircraft skin crack classification by fusing two different data modalities: CCD camera image and ultrasonic data. They designed features which they further used to train multi-class support vector machine in order to accomplish classification of cracks. In the literature, to

our knowledge, there is no much work that concerns the point cloud analysis for aircraft inspection. However, we can find some similar studies for different purposes. For instance, V. Borsu et al. [4] analyzed the surface of the automotive body panel and determined the positions and type of deformations of interest. P. Tang et al.[52] have developed a flatness defect detection algorithm by fitting a plane against point clouds and calculating the residuals of each point. Recently, R. Marani et al. [30] have presented a system based on a laser triangulation scanner that allows to identify the surface defects on tiny objects, solving occlusion problems.

The main purpose of our work is the defects detection and characterization by analyzing the surface structure in point cloud data. Specifically, this study is closely related to surface segmentation. Deriving defected surfaces from a set of 3D point clouds is not a trivial task as the cloud data retrieved from 3D sensor are usually incomplete, noisy, and unorganized. Many authors have introduced approaches and algorithms for segmenting 3D point cloud. We refer the reader to [56, 23,34] for a global review of 3D cloud segmentation strategies. In the literature, region-based method is one of the most popular approaches for 3D data segmentation. This segmentation technique is proposed by Besl and Jain in 1988 [2]. It is a procedure that groups points or subregions into larger regions based on homogeneity measures of local surface properties [12,19, 26,20,53,40,7,43,37,35]. Many of edge-based segmentation methods have been used to segment point cloud data. The principle of these methods is based on the determination of contours and then identification of regions limited by these contours [13,50,44]. Some local information of point cloud should be calculated such as normal directions [3,1], geometric and topological information [22]. In addition, the authors also use model-based approaches [45,36] and graph-based approaches [11,51,58].

3 Methodology for defect detection and characterization

3.1 Overview of the proposed system

Fig. 1 illustrates all the steps of our approach. We use a personal computer for processing point clouds acquired from a structured light 3D scanner. First, a defect detection module identifies and localizes the presence of defects or deformations on the airplane exterior surface. Then, we analyse the status of all the defected regions and extract information about the defects size, depth and orientation. We termed this second phase as defect characterization process.

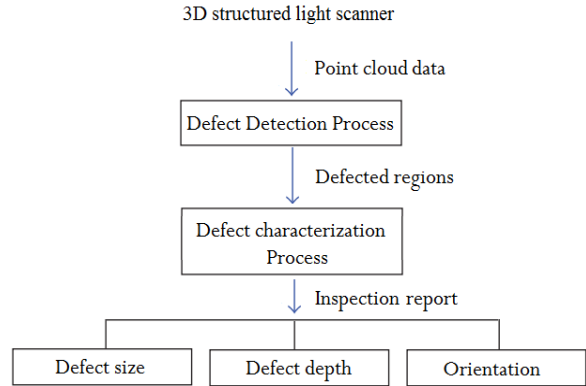The 3D data processing program must ensure robust-

Fig. 1 Overview of proposed system architecture

ness for industrial applications. In other words, it must be able to detect different types of defects with different properties.

3.2 Data acquisition

Our approach is applied to inspect the fuselage of real Airbus A320 airplane. The dataset is captured using a 3D scanner mounted on Air-Cobot (see Fig. 2a and 5b). The process is fully automatic and performs inspection of the body as the Air-Cobot moves following a predetermined trajectory like Fig. 2b. In order to test the

Fig. 2 (a) Air-Cobot and Airbus A320 airplane; (b) Illustration of moving map of Air-Cobot

robustness of our approach, we collected data of various types of defects such as undesired dents or scratches under different light and weather conditions. Few examples of our dataset are shown in Fig. 3.

3.3 Defect detection process

In this section, we introduce the defect detection process as illustrated in Fig. 4. The process is divided into

(a)   (b)

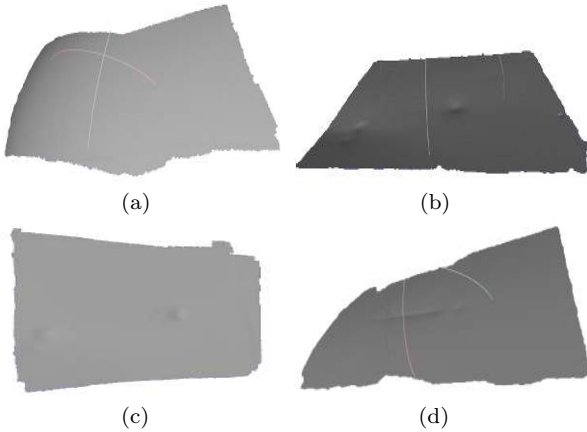(c)   (d)

Fig. 3 (a) Point cloud of surface without defect; (b) point cloud with large and small dents (c) point cloud with small dents;(d) point cloud with a long scratch

five steps. First, 3D point cloud is acquired using a 3D scanner. Next, it is smoothed by Moving Least Squares (MLS) algorithm. Further, we calculate the normal and curvature information of each point in the point cloud. We employ Region-Growing for segmenting the point cloud into two sets of points: (1) defected regions and (2) non-defected regions. Finally, these two sets are accordingly labeled for visualization.
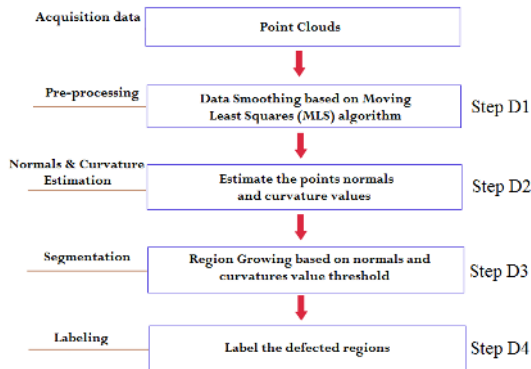

Fig. 4 Overview of the detection phase

Step 1 (data): With the advances of 3D scanning technologies, various types of 3D sensors have been developed for acquiring 3D data of high quality. This technology is very useful for material inspection and quality control. It allows to collect a lot of 3D data about the object surface and its size. Different 3D scanners such as FARO Focus 3D®, Trimble®, Artec Eva®, or Handyscan 3D® can be used for our work. After analyzing the data quality of different types of scanner, we

decided to use Artec Eva 3D scanner (see Fig. 5a). It scans quickly, in high resolution (0.5 mm) and accuracy (0.1 mm). Artec 3D scanner is also very versatile. It is recommended to keep the distance between the scanner and the object in the range $0.4 - 1m$. The scanner has field of view up to $536 \times 371mm$ (for furthest range) and frame rate of 16 frames per second. It should be noted, however, that the fundamental part of our system does not need to be changed if we want to use another type of 3D scanner.

(a)   (b)

Fig. 5 (a) Artec Eva 3D scanner; (b) Air-Cobot with the scanner mounted on a pantograph

Step 2 (Pre-processing): Although the quality of 3D scanners has been improved greatly, we still get inevitable measurement errors and outliers in point cloud. The goal of this step is to smooth and re-sample point cloud data. This pre-processing step is important because it gives more accurate local information. We use Moving Least Squares (MLS) for smoothing the surface. MLS is a method of reconstructing a surface from a set of unorganized point data by higher order polynomial interpolations in the neighborhood of a fixed point. This technique was proposed by Lancaster and Salkauskas in 1981 [27] and developed by Levin [28,29]. We are approximating our cloud with a polynomial of second degree in $\mathbf{R^n}$, since airplane fuselage is closest to this type of surface. The mathematical model of MLS algorithm is described as follows:

Consider a function $f : \mathbf{R^n} \mapsto \mathbf{R}$ and a set of points $S = \{x_i, f_i | f(x_i) = f_i\}$ where $x_i \in \mathbf{R^n}$ and $f_i \in \mathbf{R}$. The Moving Least Square approximation of the point x_i is the error functional:

$$f_{MLS}(x_i) = \sum_i (\| f(x_i) - f_i \|)^2 \Theta(\| x - x_i \|) \qquad (1)$$

We achieve the weighted least-square error at $\widehat{f}$ where:

$$\widehat{f} = \min(f_{MLS}(x_i)) = \min(\| f(x_i) - f_i \|)^2 \Theta(\| x - x_i \|)$$

In equation (1), the function Θ is called weighting function. Authors have proposed different choices for this function. For example, in [29] the author used a Gaussian function: $\Theta(d) = e^{\frac{-d^2}{h^2}}$. By applying the MLS algorithm, we can remove the small errors and further estimate the intrinsic properties of the surface such as normal and curvature (see Fig. 6).
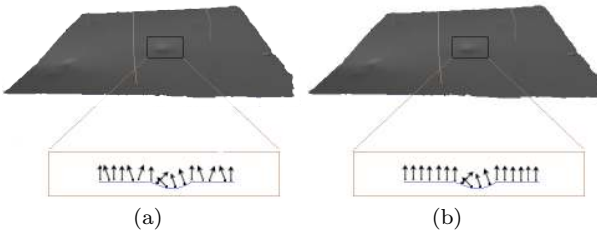

Fig. 6 Surface normal estimation on the: (a) original point cloud before resampling and (b) after resampling using Moving Least Squares algorithm

Step 3 (Normals and Curvature Estimation): In 3D geometry, a surface normal at a point is a vector that is perpendicular to the surface at that point. The surface normals are important information for understanding the local properties of a geometric surface. Many different normal estimation techniques exist in
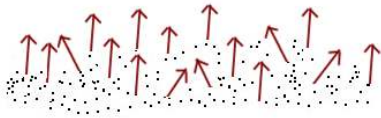

Fig. 7 Illustration of surface normals

the literature [24,8,31]. One of the simplest methods to estimate the normal of a point on the surface is based on estimating the normal of a plane tangent to the surface [41].

Given a point cloud P_N , we consider the neighboring points P^K of a query point p_q. By using a least-square plane fitting estimation algorithm as introduced in [47], we can determine the tangent plane S represented by a point x and a normal vector n_x. For all the points $p_i \in P^K$, the distance from p_i to the plane S is defined

as :

$$d_i = (p_i - x) \cdot n_x$$

S is a least-square plane if $d_i = 0$.

If we set x as a centroid of P^K:

$$x = \overline{p} = \frac{1}{K} \sum_{i=0}^{K} (p_i)$$

in order to estimate n_x, we need to analyze the eigenvalues λ_j and eigenvectors v_j $(j = 0, 1, 2)$ of the 3×3 covariance matrix A formed by the points $p_i \in P^K$:

$$A = \frac{1}{K} \sum_{i=0}^{K} (p_i - \overline{p}).(p_i - \overline{p})^T \tag{2}$$

The eigenvector v_0 corresponding to the smallest eigenvalue λ_0 is the approximation of n [41].

Another surface property that we are using in defect detection is curvature. In computer graphics, there are many ways to define the curvature of a surface at a point such as Gaussian curvature ($K = k_1 k_2$), or Mean Curvature ($H = \dfrac{k_1 + k_2}{2}$) [10] where k_1 and k_2 are the principal curvatures of the surface. In the literature, these methods are widely used for calculating curvature information [39]. Some other techniques have been proposed by the authors in [59,25]. The above approaches are accurate but very sensitive to noise and unable to estimate the curvature from a set of points directly (mesh representation required). We estimate the curvature information at a specific point by analysing the eigenvalues of covariance matrix defined in equation 2.

The curvature value of a point P_j is estimated as:

$$c\{P_j\} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{3}$$

where $\lambda_0 = \min (\lambda_{j=0,1,2})$ [38].

To resume, we estimate surface normals and curvature of each point in the cloud. This information is used in the next step.

Step 4 (Segmentation): In order to detect the damaged regions on airplane exterior surface, we need to segment the 3D points cloud data into regions that are homogeneous in terms of calculated surface characteristics, more specifically normal vector angles and curvature differences. By this way, we can divide original point cloud into two principal parts: damaged regions and non-damaged regions. The objective of this step is to partition a point cloud into sub-point clouds based on normal and curvature information which are calculated in step 3.

Let P represent the entire input point cloud, the region-based segmentation divides P into n sub-point clouds $R_1, R_2, R_3, ... R_i ..., R_n$ such that:

(1) $\bigcup\limits_{i=1}^{n} R_i = P$

(2) R_i is connected region ($i = \overline{1,n}$)

(3) $R_i \cap R_j = \oslash$ for all i and j, $i \neq j$

(4) $LP(R_i) = $ True for $i = \overline{1,n}$

(5) $LP(R_i \cup R_i) = $ False for any adjacent regions R_i and R_j

$LP(R_i)$ is a logical predicate defined on the points $p \in R_i$. Condition (4) indicates that the differences in surface properties (normal and curvature in our case) in a segmented region must be below certain threshold. Condition (5) regulates the difference between adjacent regions which should be above the threshold. The algorithm starts with random points (P_{seeds}) representing distinct regions and grow them until they cover the entire cloud. For region growing, we need a rule for checking the homogeneity of a region after each growth step. In this paper, we have used surface normals and curvatures to merge the points that are close enough in terms of the smoothness constraint. The picked point is added to the set called seeds. In each iteration a seed point is chosen from the set of unlabeled points. Seed point is always selected as a point with the lowest curvature in the current set of unlabeled points. For every seed point, the algorithm finds neighboring points (30 in our case). Every neighbor is tested for the angle between its normal and normal of the current seed point. If the angle is less than a threshold value, then current point is added to the current region. Further, every neighbour is tested for the curvature value. If the curvature is less than threshold value c_{th}, then the point is added to the seeds [42]. The criteria is shown in Eq. 4:

$$arccos(n, n_k) \leq \alpha_{th}, \qquad (4)$$

where n and n_k are normals of the seed point p and current tested point p_k, respectively.

By this way, the output of this algorithm is the set of clusters, where each cluster is a set of points that are considered to be a part of the same smooth surface. We finish by obtaining one vast cluster which is considered background and a lot of small clusters only in the defected regions. Admittedly, we obtained several clusters within the same defect, but we solve this by simply merging adjacent clusters. Our defects are never close to each other so this merging step is safe to be done.

The segmentation algorithm presented in step 4 can be described as following:

Algorithm 1: Point cloud segmentation based on surface normal and curvature

---

Input: Point cloud $P = p_1, p_2 ...., p_N$; Point normals N; Point curvatures C ; Angle threshold α_{th}; Curvature threshold c_{th}; Neighbour finding function $F(\cdot)$

Process:

1: Region list $\{R\} \longleftarrow \oslash$
2: Available points list $\{L\} \longleftarrow \{1..|P|\}$
3: While $\{L\}$ is not empty do
4:     Current region $\{R_c\} \longleftarrow \oslash$
5:     Current seeds $\{S_c\} \longleftarrow \oslash$
6:     Point with minimum curvature in $\{L\} = P_{\min}$
7:     $\{S_c\} \longleftarrow \{S_c\} \cup P_{\min}$
8:     $\{R_c\} \longleftarrow \{R_c\} \cup P_{\min}$
9:     $\{L\} \longleftarrow \{L\} \setminus P_{\min}$
10:     For $i = 0$ to size ($\{S_c\}$) do
11:         Find nearest neighbors of current seed point $\{B_c\} \longleftarrow F(S_c\{i\})$
12:         For $j = 0$ to size ($\{B_c\}$) do
13:             Current neighbor point $P_j \longleftarrow B_c\{j\}$
14:             If $P_j \in L$ and $arccos(|(N\{S_c\{i\}\}, N\{S_c\{j\}\})|) < \alpha_{th}$ then
15:                 $\{R_c\} \longleftarrow \{R_c\} \bigcup P_j$
16:                 $\{L\} \longleftarrow \{L\} \setminus P_j$
17:                 If $c\{P_j\} < c_{th}$ then
18:                     $\{S_c\} \longleftarrow \{S_c\} \cup P_j$
19:                 End if
20:             End if
21:         End for
22:     End for
23:     Global segment list $\{R\} \longleftarrow \{R\} \bigcup \{R_c\}$
24: End while
25: Return the global segment list $\{R\}$

Outputs: a set of homogeneous regions $R = \{R_i\}$.

Step 5 (Labeling): The previous algorithm allows determining the regions which contain points that belong to defects. The defects are labeled by the algorithm in order to show them on the original point cloud. The resulting labeling is shown in red color as in Fig. 8:

3.4 Defect characterization process

Next step is to characterize the defects by estimating their size and depth. For that, we use the result of the defect detection process.
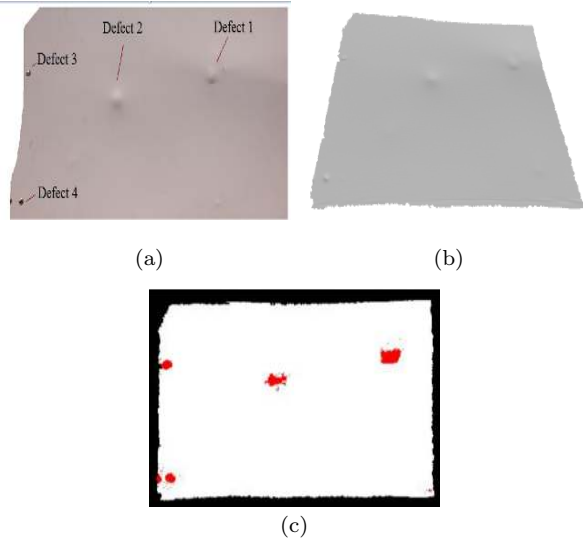
(a) (b)

(c)

Fig. 8 (a) Part of the fuselage; (b) Acquired point cloud (visualized with MeshLab).; (c) The detected defects on the original mesh are shown in red color

The purpose of this process is to extract and show the most important information about each detected defect. In our study, we propose an approach that allows estimating three main information about a defect, including size (bounding box), the maximum depth, and the principal orientation of a defect. Orientation is useful in the case of scratch-like defects (ex. Fig. 12a).

Our global approach can be viewed as a 4-step process (Fig. 9): (1) projection of the 3D point cloud onto the fronto-parallel 2D image plane (2) data preparation, (3) reconstruction, and (4) extracting information about the defects. Further on we will explain each of the steps.
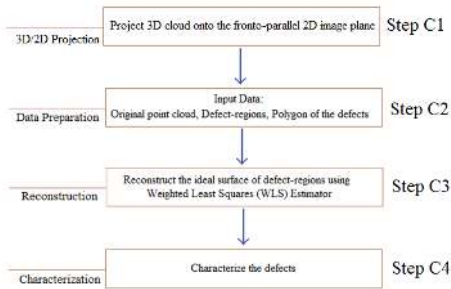

Fig. 9 Global approach of characterization process

3.4.1 Step C1 : 3D/2D projection

We are reducing our problem from 3D to 2D by projecting our 3D cloud onto the fronto-parallel 2D image plane placed on a certain distance from the cloud. We do this in order to reduce computational cost and also to facilitate operations such as neighbors search in characterization phase. We do not lose information because our clouds are close to planes. After this process, each 3D point can be referenced by its 2D projection (pixel).

Planar geometric projection is mapping 3D points of a 3D object to a two-dimensional plane called projection plane. It is done by passing lines (projectors) through 3D points and calculating their intersections with projection plane. Depending on the center of projection (COP), there are two principal kinds of projection: *parallel* and *perspective* projection [6]. When the COP is placed on a finite distance from the projection plane, perspective projection is obtained. In the case of parallel projection, the COP is considered to be at infinity and projectors are parallel. *Orthographic* projection is a subclass of parallel projection which is obtained when the projectors are orthogonal to the projection plane. If the scale is introduced in a uniform manner, it is said that *scaled orthographic projection* is performed. Scale is added in a way that the whole object is uniformly decreased/increased after being projected. This type of projection is also called *weak perspective projection*. It assumes that relative depths of object points are negligible compared to the average distance between the object and COP.

In our work, we are performing a scaled orthographic projection of our point cloud. The projection plane is placed on a certain distance d from the cloud and oriented approximately parallel to the cloud. The point cloud points are represented by their (x, y, z) coordinates in scanner reference system. We are expressing these points in the new coordinate system which enables the projection to be straightforward. This new coordinate system is placed in the mean point of the cloud with mean normal of the cloud as its z axis (O'_{rf} in Fig. 11). Finally, this system is translated for length d along its z axis. The process consists of 3 steps.

Step C1.1 (Find the mean normal of the point cloud)

The notion of centroid can apply to vectors. Let V be a set of N normal vectors in all the points of the cloud:

$$V = \{n_1, n_2...n_N\} \text{ with } n_i = [x_{n_i}, y_{n_i}, z_{n_i}]^T$$

The mean normal is calculated as:

$$\overline{n} = \frac{1}{N} \sum_{i=1}^{N} n_i = (x_{\overline{n}}, y_{\overline{n}}, z_{\overline{n}})$$

The mean normal is then normalized:

$$\widehat{n} = \frac{\overline{n}}{\|\overline{n}\|} = \left(\frac{x_{\overline{n}}}{\|\overline{n}\|}, \frac{y_{\overline{n}}}{\|\overline{n}\|}, \frac{z_{\overline{n}}}{\|\overline{n}\|} \right)$$

where $\|\overline{n}\| = \sqrt{x_{\overline{n}}^2 + y_{\overline{n}}^2 + z_{\overline{n}}^2}$.

Step C1.2 (Calculate the rotation and transformation matrix)

When the point cloud is created, it is defined in the reference system of the scanner O_{rf}. We define a new reference system O'_{rf} in which $z_{O'_{rf}} = \widehat{n}$ where $z_{O'_{rf}}$ is a unit vector along z axis of the new reference system O'_{rf}. The origin of O'_{rf} is unchanged. Further, we find the rotation matrix which aligns two unit vectors $z_{O_{rf}} = [0, 0, 1]$ and $z_{O'_{rf}} = \widehat{n}$. This task can be solved as follows.

It should be noted that the 3D rotation which aligns these two vectors is actually a 2D rotation in a plane with normal $z_{O_{rf}} \times \widehat{n}$ by an angle Θ between these two vectors:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since $\cos\theta = z_{O_{rf}} \cdot \widehat{n}$ and $\sin\theta = \|z_{O_{rf}} \times \widehat{n}\|$, we further obtain:

$$R = \begin{bmatrix} z_{O_{rf}} \cdot \widehat{n} & -\|z_{O_{rf}} \times \widehat{n}\| & 0 \\ \|z_{O_{rf}} \times \widehat{n}\| & z_{O_{rf}} \cdot \widehat{n} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 0 \\ x_2 & y_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

With R we defined a pure z-rotation which should be performed in the reference frame whose axes are $(z_{O_{rf}}, \frac{\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}}{\|\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}\|}, z_{O_{rf}} \times \frac{\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}}{\|\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}\|})$. It can be easily verified that this is an orthonormal basis. If we denote $z_{O_{rf}}$ with A and $\widehat{n}$ with B, the axes are illustrated in Fig. 10 where $^B P_A$ is the projection of vector B onto the vector A.
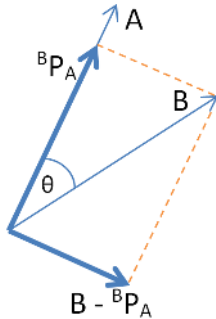

Fig. 10 Constructing the new orthonormal base. Thick blue vectors denote x and y vectors of new reference frame (not yet normalized).

Matrix for changing basis is then:

$$C = (z_{O_{rf}}, \frac{\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}}{\|\widehat{n} - (z_{O_{rf}} \cdot \widehat{n}) z_{O_{rf}}\|}, z_{O_{rf}} \times \widehat{n})^{-1}.$$

Further, we multiply all the cloud points with $C^{-1}RC$. With C we change the basis, with R we perform the rotation in the new basis and C^{-1} brings the coordinates back to the original basis. After this operation we have our cloud approximately aligned with xy plane of the original frame and approximately perpendicular to the z axis of the same frame.

Step C1.3 (Orthographic projection and translation in image plane)

Once the cloud is rotated, orthographic projection on the xy plane means just keeping x and y coordinates of each point.

$u = x; v = y$

Some of these values can be negative. In that case, we are translating all the 2D values in order to obtain positive pixel values and finally create an image. Let $p_{neg} = (u_{p_{neg}}, v_{p_{neg}})$ be the most negative 2D point in the set of projected points. We are translating all the points as follows:

$u_i = u_i + \|u_{p_{neg}}\|$

$v_i = v_i + \|v_{p_{neg}}\|$

The projection process is illustrated in Fig. 11. Examples of two point clouds and their projections are shown in Fig. 12. The projection is better visible in the Fig. 13a.
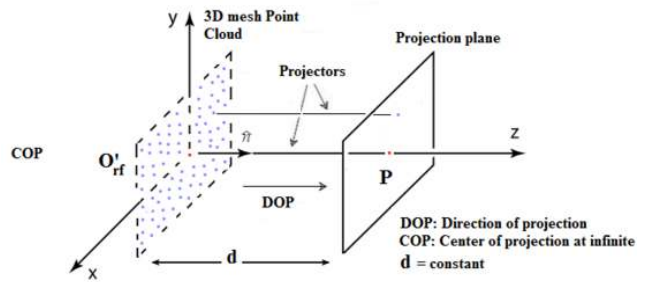

Fig. 11 Orthographic projection from 3D point cloud to 2D plane

As the last step in C1 phase, in the image space, we perform resampling of projected pixels (Fig. 13). After projection, pixels are scattered (Fig. 13a). Resampling is done in order to have regular grid of projected
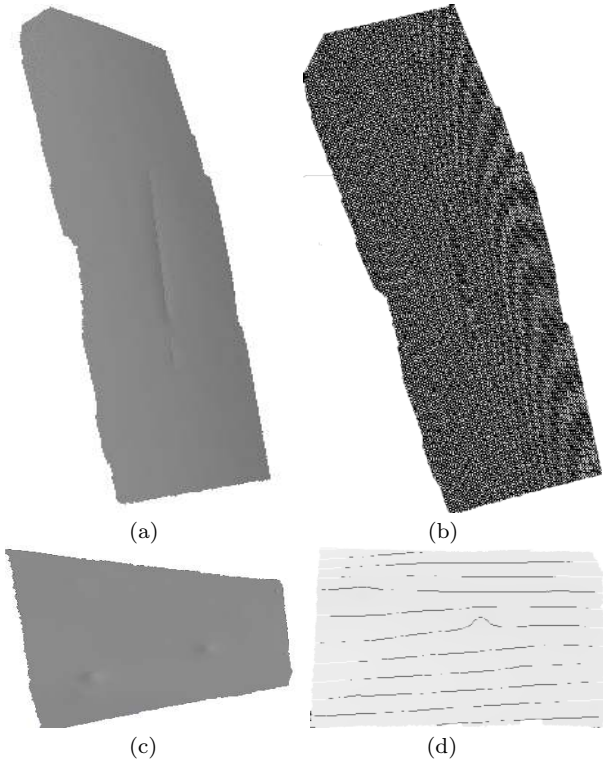
Fig. 12 (a),(c) 3D mesh of original point cloud; (b),(d) 2D image after projecting

points. Regular grid, shown in Fig. 13b, makes neighbors search faster by directly addressing neighboring pixels with their image coordinates instead of searching among scattered points.
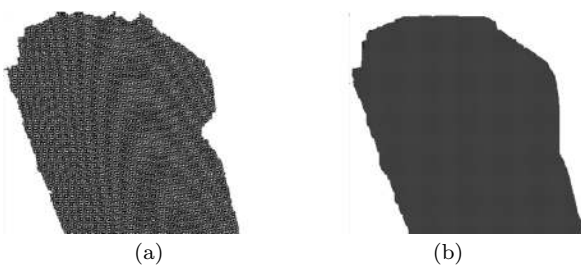

Fig. 13 (a) scattered pixels after projection (b) regular grid after resampling

The same as the whole input point cloud, the defected regions are separately projected onto another 2D image. An example is shown in Fig. 14. Note that these images have the same size as the projection of the original point cloud.

3.4.2 Step C2 : Data preparation

The second step of the characterization process is the preparation of data. There are three different types of data which are essential for this process: (1) the original point cloud, (2) identified points belonging to the defect-regions, and (3) the polygon surrounding each defect. The point cloud and all the defect-regions are available from Sec. 3.3.

In order to obtain the surrounding polygon of a defect, we start from the binary image with all projected defect points after the projection process (Fig. 14b). Note that the input data can contain one or several defects. For the defects located in close proximity, we group these defects into one by using the mathematical morphology operation called dilation [14]. This operator also allows to enlarge the boundaries of defect-regions (Fig. 14c).

After dilating the defect-regions, we identify connected components [15] on binary image (see Fig. 14d). Each of the connected components corresponds to a damage. Further, contours are extracted for each defect (see Fig. 14e). The convex hull [16] of the defect is then determined as in Fig. 14f and taken as the polygon surrounding the points which belong to the defect.

3.4.3 Step C3 : Reconstruction

Our main idea in this section is to reconstruct the ideal surface of the 3D data. This ideal surface is further used as a reference to extract the information about the status of defect by comparing the variance between the z-coordinate value of each point in the ideal surface and the corresponding point in the original data. The concept is illustrated in Fig. 15.

In order to reconstruct the ideal surface of the 3D data, we use a method called Weighted Least Squares (WLS) [33]. We are fitting a quadratic bivariate polynomial $f(u, v) : R^2 \longrightarrow R$ to a set of cloud points which are out of the polygonal defect area. We justify this by the shape of the airplane fuselage which is close to the quadratic surface.

We start with a set of **N** points $(u_i, v_i) \in \mathbf{R^2}$ with their z-values $z_i \in \mathbf{R}$. All these values are obtained in the projection phase. We search for a globally-defined function $f(u, v) = z$, that best approximates the samples. The goal is to generate this function such that the distance between the scalar data values z_i and the function evaluated at the points $f(u_i, v_i)$ is as small as possible. This is written as:

$$\min = \sum_{i=1}^{N} \theta(\| (\overline{u}, \overline{v}) - (u_i, v_i) \|) \| f(u_i, v_i) - z_i \| \quad (5)$$
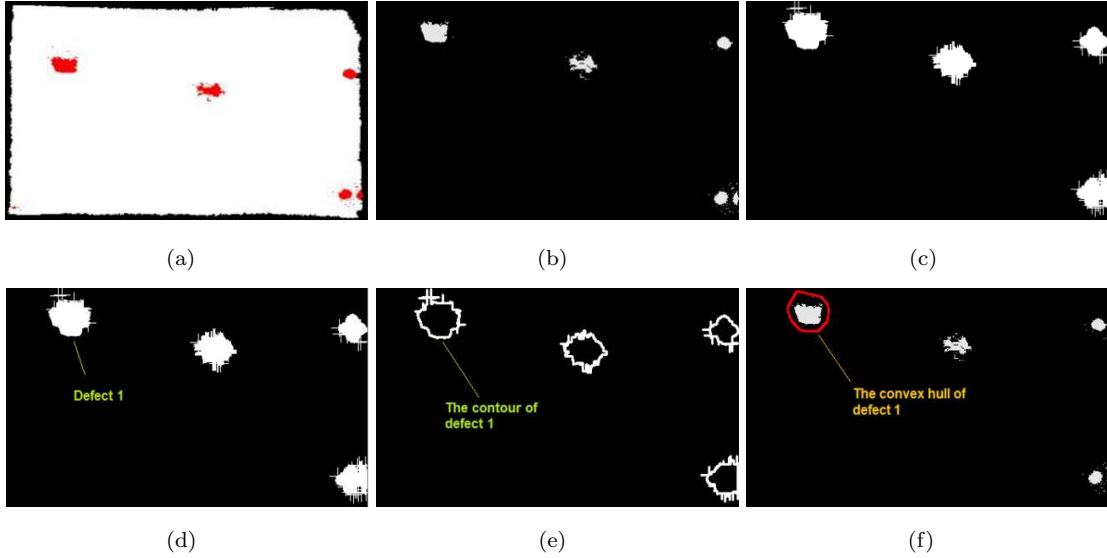
Fig. 14 (a) Labeled defects after detection; (b) Binary image after projecting defects onto the plane; (c) Defect regions after dilation; (d)Identifying each connected component as one defect; (e) Contours of the enlarged defects; and (f) Convex hull of each defect
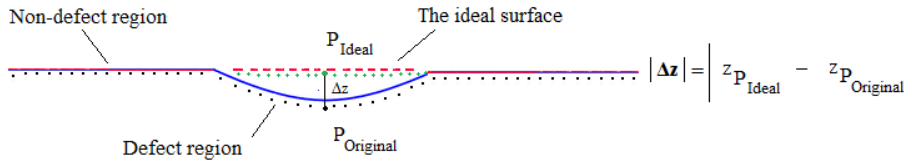

Fig. 15 An illustration of the approach for calculating defect depth

where $(\overline{u}, \overline{v})$ is a fixed point, for ex. center of mass of the defect region. We can find many choices for the weighting function $\theta(d)$ in the literature such as a Gaussian [29] or the Wendland function [55]. It is a function which is favorizing the points which are in the proximity of the defect, while assigning lower weights to the points far away from the fixed point $(\overline{u}, \overline{v})$.

3.4.4 Step C4 : Extracting information about the defects

The lowest point

For each point in a defect region, we estimate the values $\Delta z(p_i) = z_{P(ideal)} - z(p_i)$. Here, p_i is a point belonging to a defect region. We do not consider p_i as a defect point if $\left|\Delta z(p_i)\right|$ is lower than a predefined threshold. The lowest point of the defect is determined by $\max\{\left|\Delta z(p_i)\right|\}$ among all the points from that defect region. The sign of $\Delta z(p_i)$ determines if defect is a dent or a protrusion. A dent is detected when $\Delta z(p_i)$

is positive and a protrusion is detected when $\Delta z(p_i)$ is negative.

The dimension and orientation of defect

In order to show the size and the orientation of the defect, we construct an oriented bounding-box (OBB) [17]. We rely on Principal Component Analysis (PCA) [21]. Let $\mathbf{X}$ be a finite set of $\mathbf{N}$ points in $\mathbf{R^2}$. Our problem consists of finding a rectangle of minimal area enclosing $\mathbf{X}$.
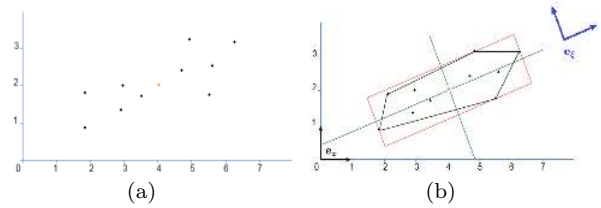

Fig. 16 Illustration of the PCA bounding-box of a set of points $\mathbf{X} \in \mathbf{R^2}$

The main idea of PCA is to reduce the dimensionality of a data set based on the most significant directions or principal components. For performing a PCA on $\mathbf{X}$, we compute the the eigenvectors of its covariance matrix and choose them as axes of the orthonormal frame e_ξ (see Fig. 16b). The first axis of e_ξ is the direction of largest variance and the second axis is the direction of smallest variance [9]. In our case, given a finite set of points in the defect-regions, we first calculate the center of mass of the defect and then apply the PCA algorithm for determining e_ξ. We continue by searching the end points along two axes of e_ξ. These points allow us to draw an oriented bounding-box of the defect as we can see for ex. in Fig. 17c .

4 Experiments and discussion

The proposed method has been tested on 15 point clouds, both with and without defective regions. The items which have been used to test and illustrate our approach are: radome, static port with its surrounding area and some parts of the fuselage. This set is considered representative since the radome (airplane nose) has a significant curvature (Fig. 22a) while static port (Fig. 22c) and fuselage (Fig. 20a) are the surfaces relatively flat. We obtained promising results which will further be illustrated. We acquired point clouds using the Artec Eva 3D scanner at Air France Industries tarmac and Airbus hangar in different lighting conditions. We acquired scans of aircraft surface with multiple defects. The same parameters of the detection algorithm are used for most of the input clouds. The scanner was placed $60 - 100$ cm from the surface. Specifically, we choose angle threshold $\alpha_{th} = 0.25$ and the curvature threshold $c_{th} = 0.3$. The original point clouds, detected defects and the corresponding characterization results for each defect are shown in Fig. 17, Fig. 18, Fig. 19, Fig. 20, Fig. 21, and Fig. 22.

The parameters we use in our algorithm play an important role in detecting the defects. The most important one is the angle threshold α_{th}. In our experiments, we have used α_{th} in the range $\{0.2 \sim 1\}$ degrees. In most cases, we have set $\alpha_{th} = 0.25$. When we reduced the value of angle threshold α_{th}, the sensitivity of the algorithm increased. Fig. 23 shows the influence of the value α_{th} on the area of detected defect.

For curvature threshold c_{th}, we test the algorithm on our dataset and we set it to $c_{th} = 0.3$. This study also indicates that the performance of the program is influenced by various factors, as scanning mode, scanning distance, density of point cloud and dimensions of the defects (depth, area).
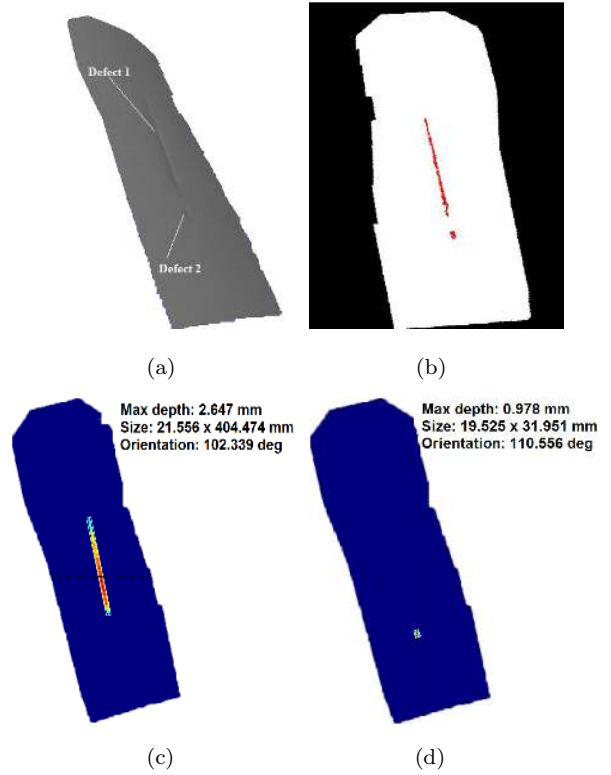

(a)   (b)

(c)   (d)

Fig. 17 Scratch on fuselage. (a) Original point cloud; (b) Defects detected; (c) Information about defect 1; (d) Information about defect 2
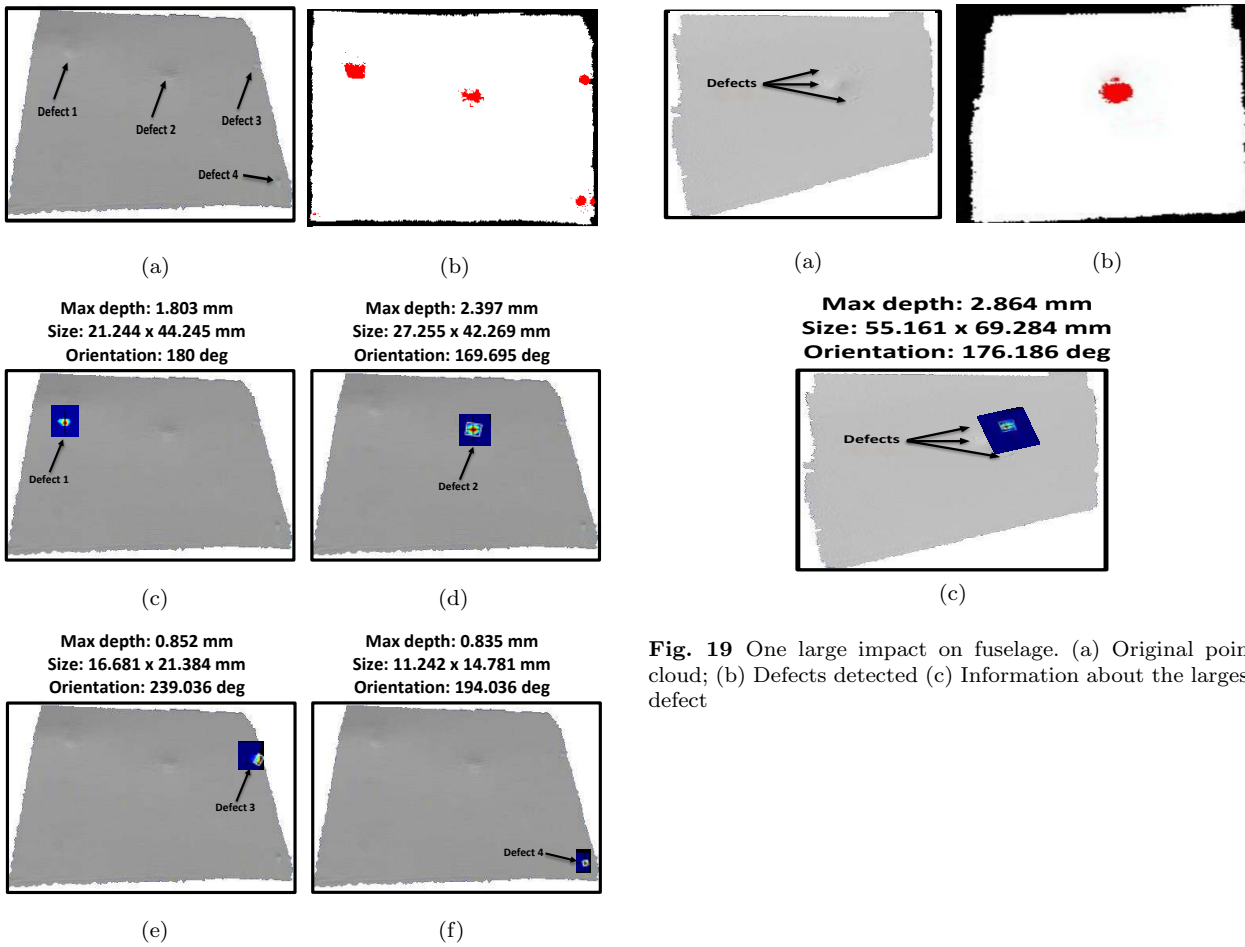
(a)

(b)

Max depth: 1.803 mm
Size: 21.244 x 44.245 mm
Orientation: 180 deg

(c)

Max depth: 2.397 mm
Size: 27.255 x 42.269 mm
Orientation: 169.695 deg

(d)

Max depth: 0.852 mm
Size: 16.681 x 21.384 mm
Orientation: 239.036 deg

(e)

Max depth: 0.835 mm
Size: 11.242 x 14.781 mm
Orientation: 194.036 deg

(f)

Fig. 18 Four impacts on fuselage. (a) Original point cloud; (b) Defects detected; (c) Information about defect 1; (d) Information about defect 2; (e) Information about defect 3; (f) Information about defect 4

(a)

(b)

Max depth: 2.864 mm
Size: 55.161 x 69.284 mm
Orientation: 176.186 deg

(c)

Fig. 19 One large impact on fuselage. (a) Original point cloud; (b) Defects detected (c) Information about the largest defect

(a)

(b)

Max depth: 2.190 mm
Size: 25.994 x 48.267 mm
Orientation: 153.435 deg

Max depth: 0.827 mm
Size: 11.304 x 14.730 mm
Orientation: 180 deg

(c)

(d)

Max depth: 2.994 mm
Size: 30.242 x 40.375 mm
Orientation: 106.699 deg

Max depth: 1.547 mm
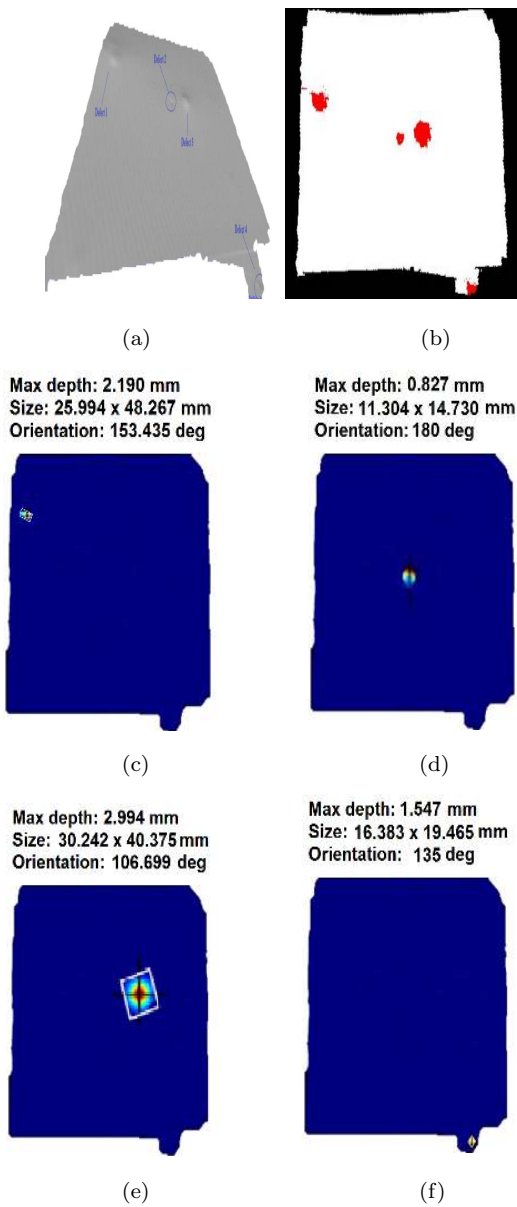Size: 16.383 x 19.465 mm
Orientation: 135 deg

(e)

(f)

Fig. 20 Four defects on fuselage. (a) Original point cloud; (b) Defects detected; (c) Information about defect 1; (d) Information about defect 2; (e) Information about defect 3; (f) Information about defect 4
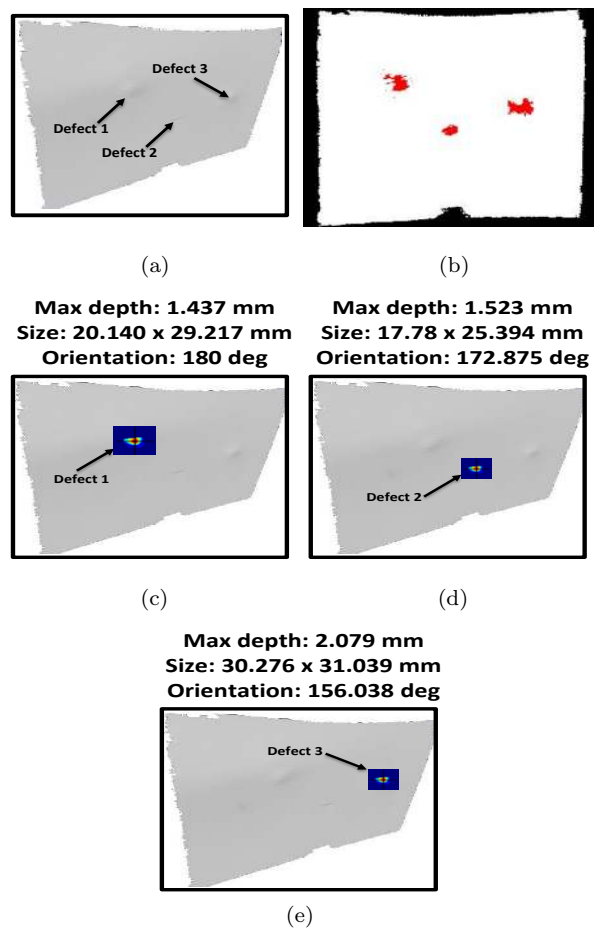

(a)

(b)

Max depth: 1.437 mm
Size: 20.140 x 29.217 mm
Orientation: 180 deg

Max depth: 1.523 mm
Size: 17.78 x 25.394 mm
Orientation: 172.875 deg

Defect 1

Defect 2

(c)

(d)

Max depth: 2.079 mm
Size: 30.276 x 31.039 mm
Orientation: 156.038 deg

Defect 3

(e)

Fig. 21 (a) Original point cloud; (b) Defects detected; (c) Information about defect 1; (d) Information about defect 2; (e) Information about defect 3
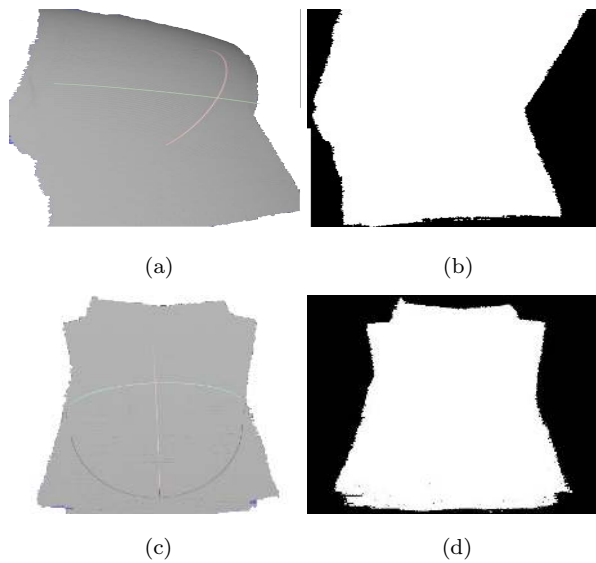

(a)

(b)

(c)

(d)

Fig. 22 Examples of point clouds without defects: (a) Radome; (c) Static port ; (b) and (d) Detection result

$(\alpha_{th} = 0.2)$ $(\alpha_{th} = 0.25)$

$(\alpha_{th} = 0.3)$ $(\alpha_{th} = 0.35)$

$(\alpha_{th} = 0.4)$ $(\alpha_{th} = 0.45)$

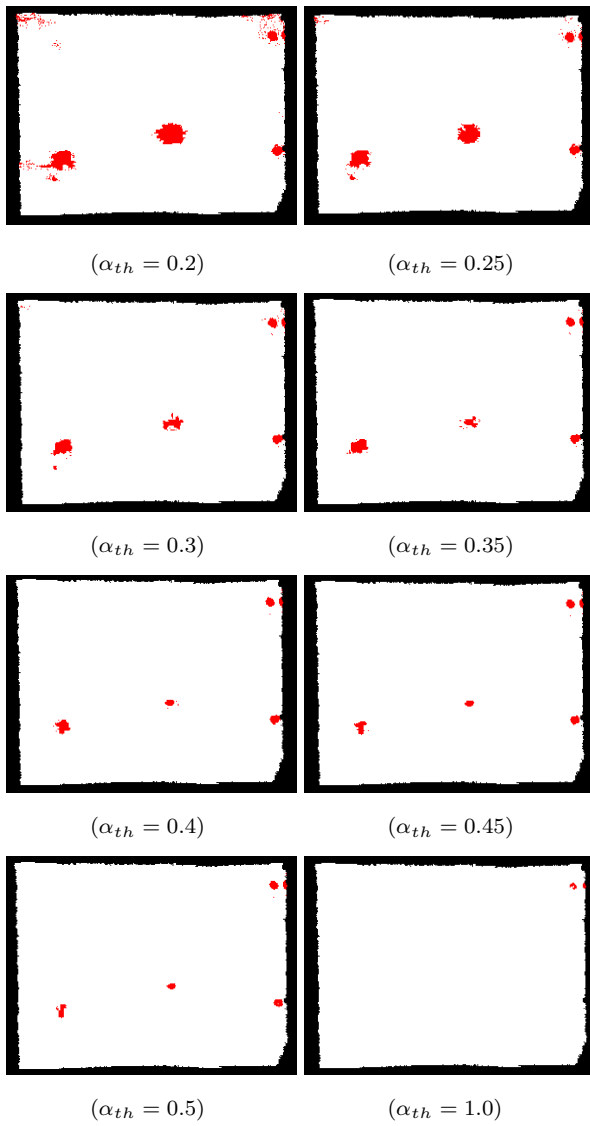$(\alpha_{th} = 0.5)$ $(\alpha_{th} = 1.0)$

Fig. 23 The influence of the value α_{th} on the detection results

4.1 Evaluation using dial gauge ground truth

In practice, the fuselage inspection is done manually by a quality manager who first examines the surface using a low angle light in order to detect defects. Next, the zone around the detected defect is demarcated with a marker pen. The zone is further examined using a dial gauge, also named dial indicator. This instrument is shown in Fig. 24a and its functioning principle is illustrated in Fig. 24b. The probe is traversing the defective area until surface contact occurs.

Obvious drawback of this method is that it depends on the expertise and mood of the person operating the equipment. Another flaw appears in the case of larger defects, such as those in Fig. 18c and Fig. 18d. Having a measuring stand with a fixed standardized diameter, the gauge can dive into the defect and report a lower depth than the real one (Fig. 25). An advantage of our method is that it can characterize defects of any size.
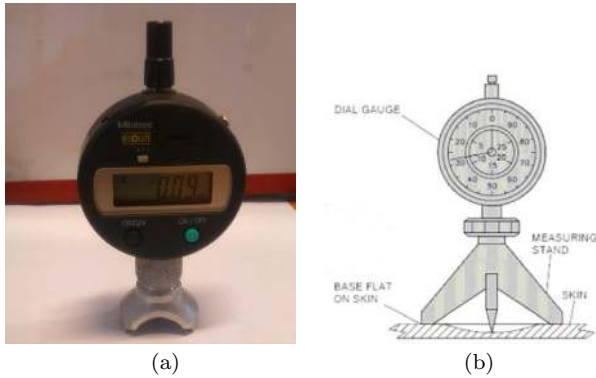

Fig. 24 (a) AIRBUS standardized dial gauge; (b) Illustration of dial gauge functioning
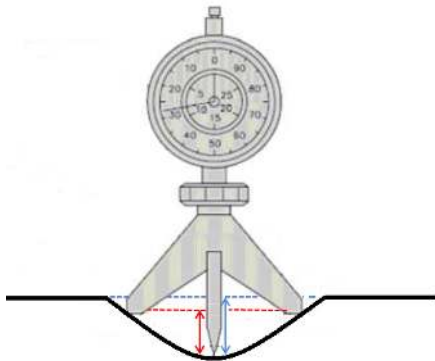

Fig. 25 Imprecision in measuring depth in the case of large defects. Red: depth measured by dial gauge; Blue: real depth.
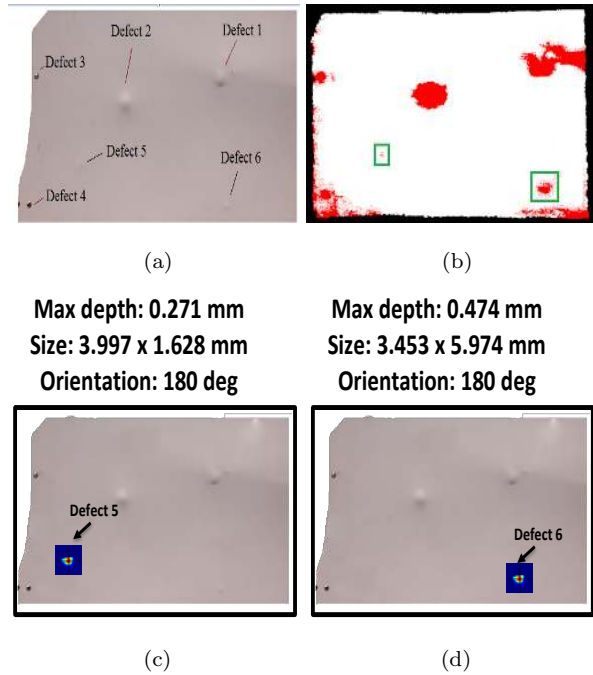

(a)   (b)

Max depth: 0.271 mm   **Max depth: 0.474 mm**
Size: 3.997 x 1.628 mm   **Size: 3.453 x 5.974 mm**
Orientation: 180 deg   **Orientation: 180 deg**

(c)   (d)

Fig. 26 (a) Part of the fuselage; (b) The detected defects are shown in red color; (c) Information about defect 5 (Dial gauge max depth: $0.31mm$); (d) Information about defect 6 (Dial gauge max depth: $0.48mm$)

In the case of small defects, we compared our method with the result obtained by AIRBUS experts using their standardized dial gauge (diameter of the measuring stand $34mm$) shown in Fig. 24a. Fig. 26a shows the same part of the fuselage as the one in Fig. 8, with indicated two additional defects (5 and 6), hardly visible to an eye. For detecting these shallow defects, α_{th} had to be decreased. For this reason, sensitivity of our detection phase increased. Consequently, we produced some false detections as well (Fig. 26b).

Fig. 26c and 26d show that the estimated maximal depths obtained by our approach are $0.27mm$ and $0.47mm$ while standardized AIRBUS dial gauge results are $0.31mm$ and $0.48mm$ respectively. The average discrepancy is around 8%.

For the reason of small diameter measuring stand, we could not obtain accurate results with the same dial gauge for neither of the defects larger than $34mm$. Therefore, we carried on the measuring in laboratory conditions. Our setup is shown in Fig. 27. Part of the fuselage was fixed on XY mobile table used for precise cutting of composite materials. The part was placed as parallel as possible with the table in order to minimize inclination. Dial gauge (with $0.01mm$ graduations) without limiting measuring stand was fixed by using magnetic base. Rectangular grid was drawn around each defect and the part was slowly moved along X and

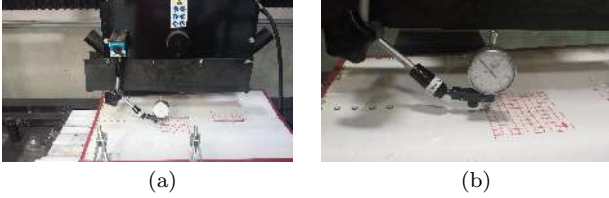Y axis of the table. In all the intersections points of the grid, the depth is measured by the dial gauge.

Fig. 27 Measuring the depth of defects with Dial gauge; (a) Measuring setup; (b) Dial gauge

This way we obtained $10cm$ long profile lines. Values read along middle lines are shown in Fig. 28 together with our results. In order to take into account possible inclination of the fuselage part, the depth is obtained by measuring the difference between the lowest point (black squares in Fig. 28) and the line obtained as average of end values on the profile (red lines in Fig. 28). The discrepancies between the Dial gauge measurements and our measured values (Fig. 18 c and d) are $e = |1.8 - 1.7| = 0.1mm$ (6%) and $e = |2.44 - 2.4| = 0.04mm$ (2%). The values obtained by the three measurement methods are given in Table 1. This table confirms our doubt that, in case of large defects (defects 1 and 2), AIRBUS gauge depth values are underestimated due to the measuring stand issue. The other tests that have been carried out so far on large defects have shown that the discrepancy is on average 5% and always below 10%. As per defects 3 and 4 from the same cloud (Fig. 18 e and f), it was impossible to measure them with dial gauge because those are two holes. However, having similar values for these two defects (0.85 and 0.84) is coherent since they are two identical screw holes produced in the manufacturing phase.
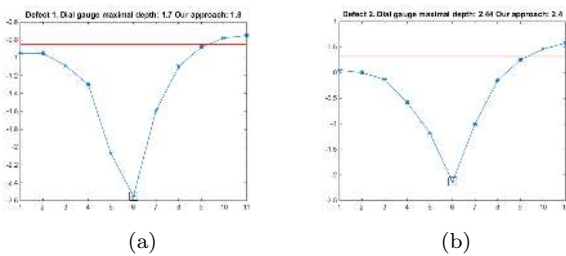

Fig. 28 (a) Profile for defect 1 (Fig. 18c); (b) Profile for defect 2 (Fig. 18d)

It should be noted that dial gauge method does not take into account the curvature of the fuselage which can affect the characterization process of defects above certain size. Contrary, with the ideal surface reconstruc-

tion explained in Sec. 3.4.3, our approach considers this aspect of the problem.

Table 1 Maximal depth of large defects shown in Fig. 18

	Our method	Dial gauge	AIRBUS dial gauge
Defect 1	1.80	1.70	1.42
Defect 2	2.40	2.44	1.73

4.2 Execution time

Execution time of the whole process is not easily quantifiable because it depends on density and size of the cloud (number of points) as well as on the number of defects. It should be noted that characterization process is performed for each detected defect sequentially. Also, in our process we are converting the input cloud from the scanner format to the format suitable for processing, which also takes time. However the total processing time which varies between 20s and 120s on our dataset, is acceptable for our application since the 3D inspection is planned to be done during more detailed and longer check, usually in the hangar. These values were obtained by testing non-optimized code on the PC with: 2.4 GHz Core(TM) i7 CPU, 8GB RAM with Microsoft Visual Studio 2013. The method was developed in C++ with the support of Point Cloud Library v.1.7.0 cite [42] and OpenCV v.3.0. library [5]. Approximately for a cloud with 30000 points, detection phase takes around $8 - 9s$ while characterization step takes $2 - 3s$ for each defect. Our time rises up to $120s$ because some of our clouds contain redundant information, caused by the longer exposure time. It is experimentally established that this scanning mode is not useful and "one shot" scanning mode is recommended. Typical cloud obtained with "one shot" scanning mode contains 30000 points. Therefore typical processing time is 20s, if we assume that typical number of detected defects is $3 - 5$.

5 Conclusions

In this paper, an original framework for the detection and characterization of defects in point cloud data has been presented. Proposed methodology is divided into two main processes. The first process is the defects detection. In this process, the Point Cloud is segmented to identify the defect regions and non-defect regions. A computer vision algorithm which is able to detect various undesired deformations on airplane surface was

developed using Region-Growing method with the local information about surface including points normal and curvature. In the next process, we developed a technique for characterizing the defects. This technique allows us to provide information about each defect such as the size, the depth and the orientation. Experiments are conducted on real data captured by 3D scanner on the fuselage of Airbus A320 airplane. This is a set of clouds encompassing various characteristics. The experimental results demonstrate that our approach is scalable, effective and robust to clouds with noise and can detect different types of deformation such as protrusions, dents or scratches. In addition, the proposed processes work completely automatically. Finally, a limitation of our approach is processing-time. In the future, we plan to reduce program execution time by optimizing our code. Thus, we believe that our results are promising for application in an inspection system. Not only limited to the context of airplane surface inspection, our approach can be applied in wide range of industrial applications. Our approach is also limited to plane-like surfaces. Strongly curved surfaces, such as wings and engine cowling, cause our characterization approach to fail. We propose cloud fitting to the available Computer Aided Design model of the airplane, in order to calculate ideal surface more precisely.

References

1. Benhabiles, H., Lavoué, G., Vandeborre, J., Daoudi, M.: Learning boundary edges for 3d mesh segmentation. In: Computer Graphics Forum, vol. 30, pp. 2170–2182. Wiley Online Library (2011)
2. Besl, P.J., Jain, R.C.: Segmentation through variable-order surface fitting. Pattern Analysis and Machine Intelligence, IEEE Transactions on **10**(2), 167–192 (1988)
3. Bhanu, B., Lee, S., Ho, C., Henderson, T.: Range data processing: Representation of surfaces by edges. University of Utah, Department of Computer Science (1985)
4. Borsu, V., Yogeswaran, A., Payeur, P.: Automated surface deformations detection and marking on automotive body panels. In: Automation Science and Engineering (CASE), 2010 IEEE Conference on, pp. 551–556. IEEE (2010)
5. Bradski, G.: Dr. Dobb's Journal of Software Tools
6. Carlbom, I., Paciorek, J.: Planar geometric projections and viewing transformations. ACM Computing Surveys (CSUR) **10**(4), 465–502 (1978)
7. Deng, H., Zhang, W., Mortensen, E., Dietterich, T., Shapiro, L.: Principal curvature-based region detector for object recognition. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, p. 18. IEEE (2007)
8. Dey, T.K., Li, G., Sun, J.: Normal estimation for point clouds: A comparison study for a voronoi based method. In: Pointbased graphics, 2005. Eurographics/IEEE VGTC symposium proceedings, pp. 39–46. IEEE (2005)
9. Dimitrov, D., Knauer, C., Kriegel, K., Rote, G.: On the bounding boxes obtained by principal component analysis. In: 22nd European Workshop on Computational Geometry, pp. 193–196. Citeseer (2006)
10. Dyn, N., Hormann, K., Kim, S., Levin, D.: Optimizing 3d triangulations using discrete curvature analysis. Mathematical methods for curves and surfaces **28**(5), 135–146 (2001)
11. Filin, S.: Surface clustering from airborne laser scanning data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences **34**(3/A), 119–124 (2002)
12. Fua, P., Sander, P.: Segmenting unstructured 3d points into surfaces. In: Computer Vision - ECCV'92, pp. 676–680. Springer (1992)
13. Golovinskiy, A., Funkhouser, T.: Randomized cuts for 3d mesh analysis. ACM transactions on graphics (TOG) **27**(5), 145 (2008)
14. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Third Edition, pp. 669-671 (2002)
15. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Third Edition, pp. 667-669 (2002)
16. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Third Edition, pp. 655-657 (2002)
17. Gottschalk, S., Lin, M.C., Manocha, D.: Obbtree: A hierarchical structure for rapid interference detection. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 171–180. ACM (1996)
18. Haddad, N.A.: From ground surveying to 3d laser scanner: A review of techniques used for spatial documentation of historic sites. Journal of King Saud UniversityEngineering Sciences **23**(2), 109–118 (2011)
19. Hoover, A., JeanBaptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D.B., Bowyer, K., Eggert, D.W., Fitzgibbon, A., Fisher, R.B.: An experimental comparison of range image segmentation algorithms. Pattern Analysis and Machine Intelligence, IEEE Transactions on **18**(7), 673–689 (1996)
20. Jin, H., Yezzi, A.J., Soatto, S.: Region-based segmentation on evolving surfaces with application to 3d reconstruction of shape and piecewise constant radiance. In: Computer Vision - ECCV 2004, pp. 114–125. Springer (2004)
21. Jolliffe, I.: Principal component analysis. Wiley Online Library (2002)
22. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts, vol. 22. ACM (2003)
23. Khan, W.: Image segmentation techniques: A survey. Journal of Image and Graphics **1**(4), 166–170 (2013)
24. Klasing, K., Althoff, D., Wollherr, D., Buss, M.: Comparison of surface normal estimation methods for range sensing applications. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 3206–3211. IEEE (2009)

25. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image and vision computing **10**(8), 557–564 (1992)
26. Köster, K., Spann, M.: Mir: an approach to robust clustering-application to range image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(5), 430–444 (2000)
27. Lancaster, P., Salkauskas, K.: Surfaces generated by moving least squares methods. Mathematics of computation **37**(155), 141–158 (1981)
28. Levin, D.: The approximation power of moving leasts-quares. Mathematics of Computation of the American Mathematical Society **67**(224), 1517–1531 (1998)
29. Levin, D.: Mesh-independent surface interpolation. In: Geometric modeling for scientific visualization, pp. 37–49. Springer (2004)
30. Marani, R., Roselli, G., Nitti, M., Cicirelli, G., D'Orazio, T., Stella, E.: A 3d vision system for high resolution surface reconstruction. In: Sensing Technology (ICST), 2013 Seventh International Conference on, pp. 157–162. IEEE (2013)
31. Mitra, N.J., Nguyen, A.: Estimating surface normals in noisy point cloud data. In: Proceedings of the nineteenth annual symposium on Computational geometry, pp. 322–328. ACM (2003)
32. Mumtaz, R., Mumtaz, M., Mansoor, A.B., Masood, H.: Computer aided visual inspection of aircraft surfaces. International Journal of Image Processing (IJIP) **6**(1), 38 (2012)
33. Nealen, A.: An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. URL: http://www. nealen. com/projects **130**, 150 (2004)
34. Nguyen, A., Le, B.: 3d point cloud segmentation: A survey. In: Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on, pp. 225–230. IEEE (2013)
35. Nurunnabi, A., Belton, D., West, G.: Robust segmentation in laser scanning 3d point cloud data. In: Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on, p. 18. IEEE (2012)
36. Nurunnabi, A., West, G., Belton, D.: Robust methods for feature extraction from mobile laser scanning 3d point clouds
37. Pauling, F., Bosse, M., Zlot, R.: Automatic segmentation of 3d laser point clouds by ellipsoidal region growing. In: Australasian Conference on Robotics and Automation (ACRA) (2009)
38. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient simplification of point-sampled surfaces. In: Proceedings of the conference on Visualization'02, pp. 163–170. IEEE Computer Society (2002)
39. Peng, J., Li, Q., Kuo, C.J., Zhou, M.: Estimating gaussian curvatures from 3d meshes. In: Electronic Imaging 2003, pp. 270–280. International Society for Optics and Photonics (2003)
40. Rabbani, T., Van Den Heuvel, F., Vosselmann, G.: Segmentation of point clouds using smoothness constraint. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36**(5), 248–253 (2006)
41. Rusu, R.B.: Semantic 3d object maps for everyday manipulation in human living environments. Ph.D. thesis, Technische Universität München (2009)
42. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China (2011)
43. Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M.: Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems **56**(11), 927–941 (2008)
44. Sappa, A.D., Devy, M.: Fast range image segmentation by an edge detection strategy. In: 3D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, pp. 292–299. IEEE (2001)
45. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point cloud shape detection. In: Computer graphics forum, vol. 26, pp. 214–226. Wiley Online Library (2007)
46. Seher, C., Siegel, M., Kaufman, W.M.: Automation tools for nondestructive inspection of aircraft: Promise of technology transfer from the civilian to the military sector. In: Fourth Annual IEEE DualUse Technologies and Applications Conference (1994)
47. Shakarji, C.M., et al.: Leasts-quares fitting algorithms of the nist algorithm testing system. Journal of ResearchNational Institute of Standards and Technology **103**, 633–641 (1998)
48. Siegel, M., Gunatilake, P.: Remote inspection technologies for aircraft skin inspection. In: Proceedings of the 1997 IEEE Workshop on Emergent Technologies and Virtual Systems for Instrumentation and Measurement, Niagara Falls, CANADA, pp. 79–78 (1997)
49. Siegel, M., Gunatilake, P., Podnar, G.: Robotic assistants for aircraft inspectors. Industrial Robot: An International Journal **25**(6), 389–400 (1998)
50. Simari, P., Nowrouzezahrai, D., Kalogerakis, E., Singh, K.: Multi-objective shape segmentation and labeling. In: Computer Graphics Forum, vol. 28, pp. 1415–1425. Wiley Online Library (2009)
51. Strom, J., Richardson, A., Olson, E.: Graph-based segmentation for colored 3d laser point clouds. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 2131–2136. IEEE (2010)
52. Tang, P., Akinci, B., Huber, D.: Characterization of three algorithms for detecting surface flatness defects from dense point clouds. In: IS&T/SPIE Electronic Imaging, pp. 72,390N–72,390N. International Society for Optics and Photonics (2009)
53. Tóvári, D., Pfeifer, N.: Segmentation based robust interpolation - a new approach to laser data filtering. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36**(3/19), 79–84 (2005)
54. Wang, C., Wang, X., Zhou, X., Li, Z.: The aircraft skin crack inspection based on different-source sensors and support vector machines. Journal of Nondestructive Evaluation **35**(3), 46 (2016). DOI 10.1007/s10921-016-0359-3. URL http://dx.doi.org/10.1007/s10921-016-0359-3
55. Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in computational Mathematics **4**(1), 389–396 (1995)
56. Wirjadi, O.: Survey of 3D image segmentation methods, vol. 35. ITWM (2007)
57. Wong, B.S., Wang, X., Koh, C.M., Tui, C.G., Tan, C., Xu, J.: Crack detection using image processing techniques for radiographic inspection of aircraft wing spar. InsightNonDestructive Testing and Condition Monitoring **53**(10), 552–556 (2011)
58. Yang, J., Gan, Z., Li, K., Hou, C.: Graph-based segmentation for rgb-d data using 3d geometry enhanced superpixels. Cybernetics, IEEE Transactions on **45**(5), 927–940 (2015)

59. Zhang, X., Li, H., Cheng, Z., Zhang, Y.: Robust curvature estimation and geometry analysis of 3d point cloud surfaces. J. Inf. Comput. Sci **6**(5), 1983–1990 (2009)