An Octree-Based Two-Step Method of Surface Defects Detection For Remanufacture

Yan He[a], Wen Ma[a], Yufeng Li [a,*], Chuanpeng Hao[a], Yulin Wang[b], Yan Wang[c]

[a] State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, 400030, China
[b] School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China
[c] Department of Computing, Engineering and Mathematics, University of Brighton, Brighton, BN2 4GJ, United Kingdom
* Corresponding author. State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400030, China (Yufeng Li).
E-mail address: liyufengcqu@cqu.edu.cn (Yufeng Li, Tel: +86 17783903035).

Abstract:

Accurate and quick detection has a significant bearing on overall productivity of remanufacture. 3D scanning technologies have been widely applied in defects detection by comparing the damaged model with the nominal model. In this process, a huge amount of point cloud data is required to ensure detection accuracy whereas resulting in large storage space and long processing time of detection. This paper proposed an efficient two-step method based on octree to detect defects accurately and quickly for remanufacturing. In this method, the damaged point cloud and the nominal point cloud are first registered. Then a two-step detection approach is developed to extract the surface defects, coarse detection and detailed extraction, where the octree method is applied to create an effective topology of discrete points and perform the Boolean operation for defects extraction. In coarse detection, rough location and size information of the defects are acquired from the whole point cloud data. Based on coarse detected boundary box containing defects, the detailed extraction step is applied to extract corresponding defects shape accurately. The feasibility of proposed method was validated by using a case to detect defects of a damaged turbine blade and the detection results can be used to generate restoration tool path. The results show that the proposed method outperforms state-of-art defects detection methods, which can reduce time by 74.03% and reduce error by 36.86%, respectively.

Keywords Octree · Remanufacturing repair · Point cloud registration · Defects extraction

1. Introduction

Remanufacturing is an industrial process of restoring worn-out products to “like new” condition, which offers significant economic, environmental and social benefits [1]. Nevertheless, the components in remanufacturing factory have highly uncontrollable difference in materials, structures and shape complexities. In addition, the failure conditions of remanufacturing products are usually unknown because there is predominantly no data about the information of defects. Therefore, the inspection procedure is a crucial stage to acquire information of damaged products.

Several researchers [2] indicated that typical surface defects of components are often found in the form of cracks, dents, burrs and abrasions. In addition, as the critical dimensions of the component no longer meet the design requirements, the components are also damaged [3]. After inspection, these above defects are detected by various methods and vital parameters of defects are calculated such as size and volume. These quantitative evaluation [4] of the surface defects is acquired for making the decision of

remanufacture, whether the part can be repaired or scrapped. Therefore, accurate and quick detection has significant bearing on overall productivity of remanufacture. Besides, the defect shape is vital for providing the exact missing geometry to generate the particular tool path for each defect so that each defect can be repaired in a targeted way [5]. The framework of detecting these defects is shown in Fig.1.

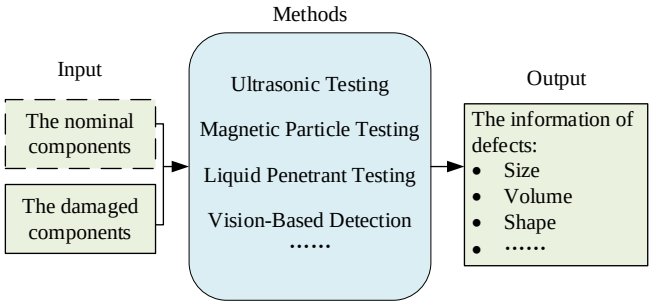


Fig.1 The framework of defects detection

Some methods using sensors are developed to detect defects of damaged components [6-8], such as ultrasonic testing, liquid penetrant testing, magnetic particle testing, 3D laser scanning detection, etc. Rannou et al. [6] use ultrasonic technology to continuously monitor the wall thickness of cold-rolled steel sheets. But this method is only adopted to the components which have simple geometrical shape. Abolfazl et al. [7] use magnetic particle nondestructive testing in the surface crack detection of welded components and this method is difficult to automate because of the harsh application conditions. Zolfaghari et al. [8] used visible liquid penetrant testing to detect surface cracks of welded components and this method is only suitable for surface-breaking defect. Because the quantity of components in remanufacture is large and the difference is great, the above-mentioned detection methods are not suitable for remanufacturing due to their respective limitations. It is necessary to develop an effect method of surface defects detection for remanufacturing.

Due to recently development of all kinds of 3D sensors and its advantages of non-contact, high precision and high adaptability [9], the defects detection methods led by 3D laser scanning technology attract wide attention. 3D laser scanning technology provides an accurate way to acquire abstraction and representation of the entity damaged component in format of dense point cloud [10]. In this process, huge amount of point cloud data is required to ensure the detection accuracy whereas resulting in large storage space and long processing time of detection. In addition, the collected 3D point cloud is a huge amount of discrete and nonuniform points which make it difficult to find out the defects area. The core of defects detection methods based on 3D laser scanning technology are mainly processing collected data by adding topology and using it to detect the defects by comparing nominal and damaged model. In the absence of a nominal model, researches always reconstructed nominal model from collected point cloud model, however, these strategies and algorithms are only developed for components which have similar cross section or curvature constraint rather then general components [11-13].

The current researches on defects detection for general components are all carried out in the scene of nominal models. In these methods, the topology structure of point cloud is usually built by triangular mesh reconstruction. For example, Zhang et al. [14] introduced a novel defects extraction methodology for general components based on voxel modeling. In this method, the point cloud is converted to stereolithography (STL) models for creating the topology, then converted to a voxel model and the voxel models are compared to extract defects. Zhang et al. [15] introduces a defect extraction algorithm benefits from tri-dexel modeling. The point cloud is converted to STL models as well firstly. However, 3D reconstruction method of point cloud data exists many problems, such as: 3D modeling efficiency is low,

which makes the detection process more complicated and time-consuming [16].

For the above problems, the methods of detecting defects based on point cloud directly without model reconstruction attract wide attention. Feng et al. [17] demonstrated a defect area extraction approach for complicated geometries by the scanned point cloud model with the nominal model. In this method, the topology of point cloud is created by calculating the distance among all the matching point pairs, and the calculation usually exhibit exponential growth with the number of points and the complexity of the objects in practical situations by this method. In order to reduce the processing time, there are many point cloud simplification [18] methods to reduce the number of the scanned points and thereby reducing the detecting time accordingly. However, the simplification of point cloud could cause information loss and undetection of defects. In general, two major challenges exist for processing of point cloud to extract defects in real industrial environments:

1) The collected 3D point cloud is a huge amount of discrete and nonuniform points which miss relationships. It is difficult to develop an effective method to create relationships between points.

2) The damaged components in remanufacturing factory exhibit highly uncontrolled differences and the scanned point cloud also exhibit highly uncontrolled differences in the size and shape complexities. It is necessary to develop an adaptive method for point cloud of general components.

To fill this research gap, octree is used to create topology and detect defects. Similar to voxels, octree nodes could used to perform Boolean operation to compare the damaged model and the nominal model to detect defects. Compared with voxel-based method, the octree is used to create topology point cloud directly rather than model reconstruction. During the creation of octree, node-to-node relationships rather than point-to-point relationships are established. The octree is obviously faster because the number of octree nodes is dramatically smaller than the number of points. Octree as a spatial data structure is not influenced by the density of point clouds so point cloud simplification is not required to reduce the processing time. What's more, the octree is easy to adapt to point cloud of general components in remanufacturing factory which have highly uncontrollable difference in components' shapes, structures and failure conditions.

In this study, a novel defect area extraction method based on octree was proposed. In the process, the damaged point cloud model is obtained by 3D scanners. The nominal point cloud is converted from the nominal CAD model. Next, fast point feature histogram and the iterative closest point algorithm are selected for initial and fine registrations, respectively, between above point clouds. And then, a two-step detection approach consisting of coarse detection and detailed extraction was developed to extract the surface defects, in which the octree is applied to create an effective topology of discrete points and the structure of octree is used to perform the Boolean operation for defects extraction. In coarse detection, rough location and size information of the defects are acquired from the whole point cloud data. Based on coarse detected boundary box containing defects, the detailed extraction step is applied to extract corresponding defects shape accurately.

When repairing the damaged components, the restoration tool path can be generated to repair the damaged components to its original performance. Existing methods all need reconstruct surface to STL which is an intermediate form of 3D models for the generation of tool path. With the development of advanced machining technology, the technology of generating tool path directly from point cloud has been gradually developed in recent years [22-25]. Based on these methods, the tool path for remanufacturing is generated directly from point clouds without model reconstruction.

The main contributions of this paper are summarized as:

(1) An effective detection method is developed for surface defects by comparing nominal and

damaged point cloud model.

(2) For collected discrete point cloud model, the octree is used for adding topology effectively.

(3) A two-step approach is developed for effective detection and extraction. The location and size of defects were detected by coarse detection and the accurate shape of defects was extracted by detailed extraction.

(4) The tool path for remanufacture is generated directly from point clouds without model reconstruction.

The rest of this paper is organized as follows. Section II introduces octree for adding topology of point clouds. In Section III, the details of the proposed method for defects extraction are introduced, including point cloud registration and two-step defects detection approach. In Section IV, a case about blade is introduced, in which the defects are extracted and the tool path is generated. In Section V, two experiments are designed to validate effectiveness of the proposed method. Finally, Section VI concludes the paper with conclusions.

2. Materials and Methods

This section comprehensively describes the developed method for detecting and extracting of surface defects. The framework of the method for surface defects detection and extraction is shown in Fig. 2. It is composed of two main parts, including point cloud registration and defects detection. Once the damaged point cloud and the nominal point cloud are obtained, fast point feature histogram (FPFH) and iterative closest point (ICP) algorithm are used for registering. Based on the registration results, a two-step detection approach based on octree is developed to detect and extract the surface defects, where the defects can be detected and extracted accurately and quickly. The detailed procedure of the method is explained in Sub-section 2.1 and Sub-section 2.2.

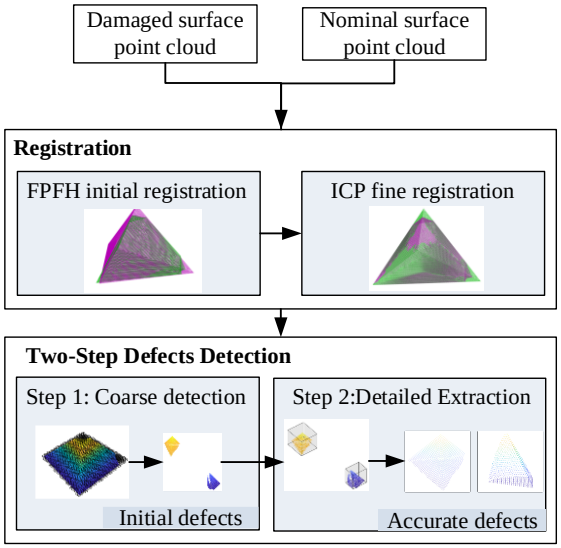


Fig. 2 Framework for surface defects detection and extraction

2.1 Point Cloud Registration

3-D point cloud registration is important for extracting surface defects [27]. The damaged point cloud is acquired by scanning the defective part and the nominal point cloud model represents the undamaged geometry part which can be acquired from the CAD model. In practice, these nominal and damaged point clouds are acquired in different ways making them in random position and orientation, as shown in Fig.3a. The purpose of point cloud registration is mainly to convert the arbitrary point clouds

into a same coordinate system that makes point clouds have the same position and orientation through rotation, translation, and other transformations. Point cloud registration directly affects the accuracy of the results of defects detection. Therefore, it is necessary to registration at first for extracting surface defects [21]. In this study, FPFH is selected for initial registration and ICP is selected for fine registration between point clouds.

Initial registration is based on FPFH aiming at rough alignment between the nominal and damaged point clouds to make the point clouds have better initial positions, which can improve the accuracy and efficiency of ICP algorithm registration. The FPFH initial registration process obeys the following steps and the effect after registration is shown in Fig.3b [27].

Step 1: Extracted feature points from the point cloud according to FPFH [29];

Step 2: Find four pairs of points based on several given constraints in regard to their features, distances, and location relationships;

Step 3: Based on previous four point-pairs, add point pairs sequentially until the registration requirements are meet;

Step 4: Calculate the rotation and translation matrix from matched point pairs.

For fine registration, the ICP and related algorithm still play a superior role in the area of accurate registration of point clouds. In our study, the ICP algorithm is utilized to further enhance the registration precision between the nominal and damaged models after initial registration. The ICP fine registration process obeys the following steps the effect after registration is shown in Fig.3c.

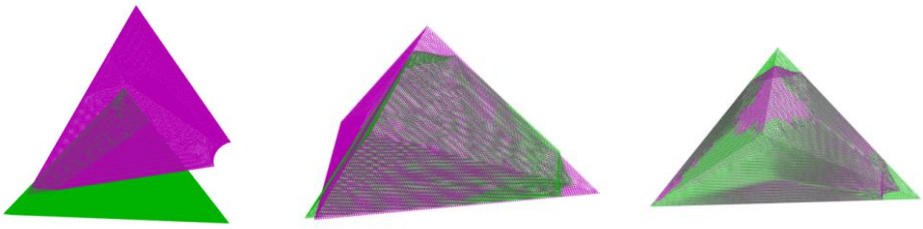
Step 1: Seek for a list of point pairs (x_i, y_i). The points x_i are from the nominal point cloud Q_n and the corresponding points y_i from damaged point cloud Q_d are selected by calculating the nearest Euclidean distance;

Step 2: Calculate the transformation matrix $M_{dn} = [R_{dn}\ T_{dn}]$ by minimizing Eq. (1) based on above point pairs;

$$e(Q_d, Q_n) = \sum_{i=1}^{k} (R_{dn}x_i + T_{dn} - y_i)^2 \quad (1)$$

Step 3: Calculate the registration error between the nominal and damaged point clouds. If the registration error is not meet requirement, the aforementioned steps are repeated until the error is meet.

Finally, these damaged and nominal point clouds are in a same coordinate system with same position and orientation using the FPFH and ICP algorithm.



(a) (b) (c)

Fig.3 Model registration. **a** unaligned position of nominal and damaged models; **b** Initial registration based on FPFH; **c** Fine registration based on ICP

2.2 Defects Detection Approach

2.2.1 Octree for Creating Topology

Before defects detection, the octree is created for adding topology and the spatial structure of octree is used to perform Boolean operation to detect and extract surface defects.

Octrees is a spatial data structure for storing 3-dimensional data similar to binary trees and quadtrees. A node of octree is segmented into eight child nodes, and each child node corresponds to an eighth of the volume and space of this node. Each node of octree represents the space surrounded by a cuboid, also for convenience, generally considered to be an axis aligned cube. All cubes without points are interpreted as empty space. These empty nodes usually means that the corresponding space is left out in order not to further subdivision. Only nodes that contain points are segmented continually to create child nodes. Moreover, when creating octree, a stop rule for the subdivision of octree must be defined. The minimal cube size or the maximal number of iterations is defined as stopping criteria which is equivalently the size of leaf nodes. If the cube size is below the minimal cube size or the number of iterations exceed the maximal number of iterations, the subdivision of octree is stopped and leaf nodes of octree are generated. Ultimately, a list of points is stored in each occupied leaf nodes. In addition, the point cloud is collected by scanning the surface of objects which leads to point cloud is only distributed on the surface of objects rather than fully volumetric. As a result, most space is not occupied, and most octree nodes only have few child nodes making octree store point clouds fast. The octree is therefore extremely befitting to store scanned point cloud efficiently.

A recursive refinement of a three-level octree is illustrated in Fig.4. The minimum bounding box of model is regarded as the root of octree. The node has up to eight children, and empty nodes are left out in order not to further subdivision. And the nodes of level 1-2 are regarded as inner nodes which need further subdivision. To subdivide the nodes of level 2, the nodes of level 3 which is the leaf nodes are created. The leaf nodes need not further subdivision.

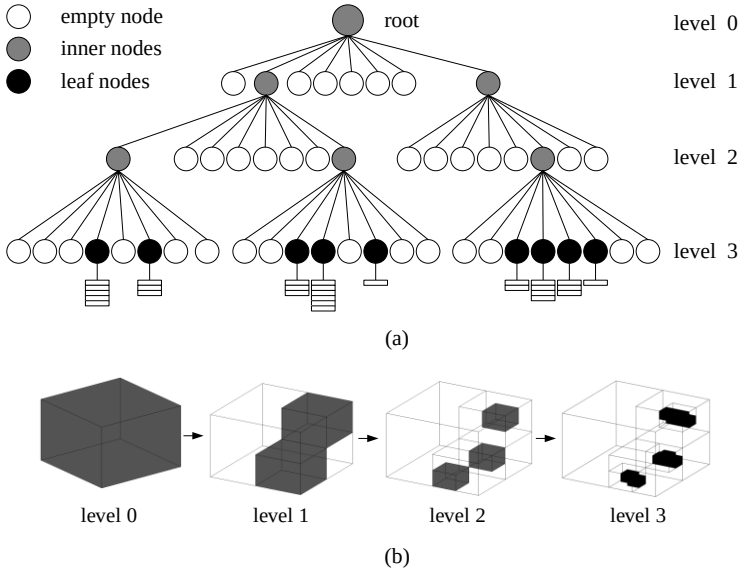


Fig.4 A three-level octree. **a** Spatial subdivision of three-level octree; **b** The corresponding tree structure of the sparse data structure. Occupied inner nodes are shaded grey. Occupied leaf nodes are shaded black.

An example showing the create process of octree for a point cloud is illustrated in Fig.5. First, the bounding box (X, Y, Z, x, y, z) of point cloud is calculated as shown in Fig.5c. Subsequently, the point cloud model with bounding box is used to create the octree. The child nodes are created for the point cloud model and leave the empty nodes out constantly until meet the stopping criteria. Then the points are stored in each occupied leaf nodes so that the octree for storing point cloud is created. The two-level octree of point cloud is shown in Fig.5d. The three-level octree of point cloud is shown in Fig.5e. The

five-level octree of point cloud is shown in Fig.5f.

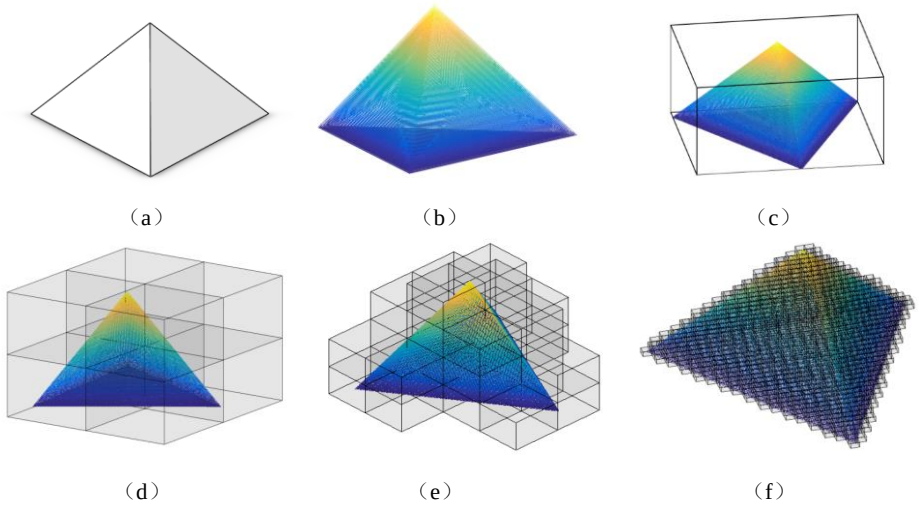


Fig.5 The octree creation. **a** CAD model; **b** point cloud model; **c** octree model with bounding box; **d** the two-level octree of point cloud; **e** the three-level octree of point cloud; **f** the five-level octree of point cloud

When selecting the number of iterations, there is a trade-off between the time and node size. Node size is the minimum defect size that the method could detect. The accuracy increases with the decrease of node size. Node size δ is dependent of the scale of the damaged components and the number of iterations which can be calculated as follows:

$$d = \max(X - x, Y - y, Z - z) \tag{2}$$

$$\delta = d * (\frac{1}{2})^n \tag{3}$$

where X, Y, Z represent the maximum value on each axis, and x, y, z represent the minimum value on each axis. The n represent the number of iterations.

Increasing the iterations can simultaneously improve the accuracy of detection, but can result in increasing storage space and processing time. Decreasing the iterations can cause undetection of defects. The effect of iterations of octree on node size and time is shown in Fig.6. If the octree is directly used to extract defect shape, it will bring a lot of unnecessary calculation and processing time because the undamaged area is subdivided continuously simultaneously. Therefore, a two-step defects detection approach was developed based on octree. Octree with a smaller number of iterations is used to detect defects firstly which will acquire a coarse defects information such as location and size. A method of point cloud segmentation is used to obtain isolated defects regions. Identify the bounding box of each defect and use octree to extract corresponding defect shape more accurately.

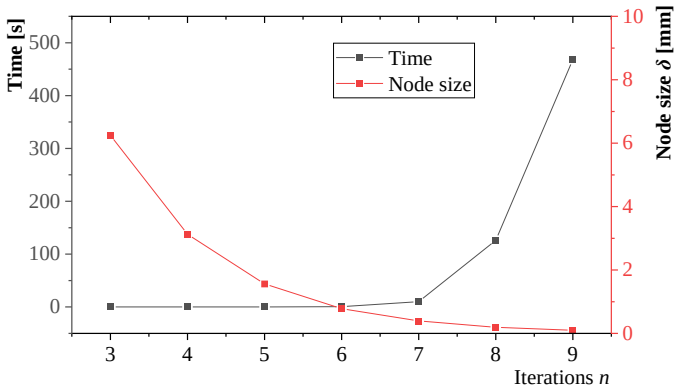


Fig.6 Effect of iterations of octree on node size and time

Algorithm Octree Creation of Point Cloud

Input: The point cloud model, M_P; the stopping rule for the subdivision of octree, S; a minimal cube size D or the maximal number of iterations N;

Output: Octree Model, M_O;

$n \leftarrow 1$
$V_1 \leftarrow M_P$
Calculate the boundary $B(xmax, ymax, zmax, xmin, ymin, zmin)$ of V_1;
$d \leftarrow max(xmax\text{-}xmin, ymax\text{-}ymin, zmax\text{-}zmin)$
while $n < N$ **or** $d < D$ **do**
  $K \leftarrow length[V]$
  $M \leftarrow \emptyset$
  Confirm the number of cube K;
  for $k = 1$ to K **do**
    Divide the bounding box V_i to eight child cubes $C_1, C_2 ... C_8$ along each axis respectively;
    $n \leftarrow n + 1$
    for $i = 1$ to 8 **do**
      Calculate the boundary $B(xmax, ymax, zmax, xmin, ymin, zmin)$ and cube size d of C_i;
      if C_i do not include the points **then**
        C_i is left out;
      else
        Keep the C_i and its boundary $B(xmax, ymax, zmax, xmin, ymin, zmin)$;
        $M \leftarrow [M; B]$
      end if
    end for
  end for
  $V \leftarrow M$
end while
$M_O \leftarrow M$

2.2.2 Coarse Detection

Coarse detection is used based on octree for quickly locating all defects from the whole point cloud data. at the first stage. The damaged point cloud model and its nominal CAD model is shown in Fig.7a and Fig.7b. The topology of the nominal point cloud is created by octree at first in Fig.7c. The spatial structure of octree is extracted to apply to damaged point cloud in Fig.7d, and the point out of the structure and empty nodes are filtered. The point in empty nodes from nominal model is part I. The point out of the structure is part II. In order to make it more intuitive, a cross-section is extracted from the point cloud model as shown in Fig.7e. The empty nodes are extracted as shown in the Fig.7f and the points in empty nodes from nominal point cloud are extracted as shown in the Fig.7g. The difference set between damaged point cloud and points in non-empty nodes are developed as shown in the Fig.7i. Therefore, the point cloud of defect area is extracted, which includes two parts, part I is from the nominal point cloud, part II is from the damaged point cloud model. In this step, the octree is created with a smaller number of iterations as the minimal cube size that is stopping criteria. The initial defects detection process based on octree is illustrated as follows. As shown in the Fig.7j, even though the defect is detected roughly, it is difficult to provide accurate quantitative evaluation of the surface defects. It is necessary to a further accurate detection. The information of defect including the location and size will be calculated for further extraction.

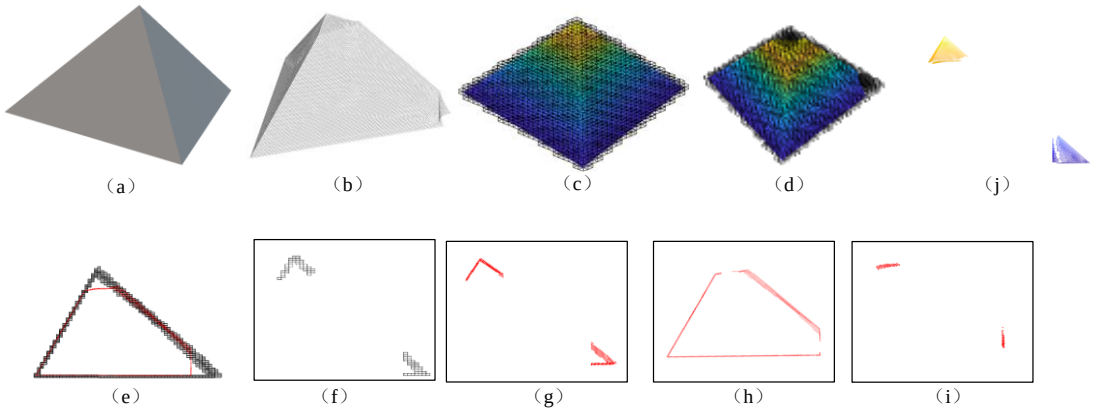


Fig.7 The procedure of coarse detection. **a** the nominal CAD model; **b** the damaged point cloud model; **c** the octree of nominal point cloud; **d** octree and damaged model; **e** a cross profile of d; **f** the empty nodes; **g** part I of defects; **h** the undamaged area; **i** part II of defects; **j** the coarse defects.

Algorithm Defects Detection

Input: The damaged point cloud model from 3D Scanner, M_d; the the nominal CAD model, M_n; the stopping rule for the subdivision of octree, S;

Output: The coarse defect area, M_b;

Convert the nominal CAD model M_n to point cloud model;
Create octree of the nominal point cloud M_{on};
$K \leftarrow length[M_{on}]$
$P_0 \leftarrow \emptyset$
$P_1 \leftarrow \emptyset$
for $i = 1$ to K **do**
    if V_i do not include the points from the damaged point cloud **then**
        Select the points list p in V_i from the damaged point cloud;
        $P_0 \leftarrow [P_0; p]$;
    else
        Select the points list p in V_i from the nominal point cloud;
        $P_1 \leftarrow [P_1; p]$;
    end if
end for
$P_2 \leftarrow M_d - P_0$;
$M_b \leftarrow [P_1; P_2]$

2.2.3 Detailed Extraction

Even though the defects are detected by previous method, the extracted shapes of defects are not closed completely which cause the loss of information. Hance, this step was used to extracted shape of defects accurately. The detailed extraction is divided to point cloud segmentation and defects extraction.

The reason of components damaging is complex and a damaged component often has more than one defect. The failure conditions of each defect are different which means it is necessary to analyze each defect individually. In addition, when defects are distributed in different locations of damaged parts, it is time-consuming to detailed extraction of defects directly. The main process of point cloud segmentation aims at grouping several initial defects point cloud into single regions to process individually. After coarse detection, the point cloud of defects area is extracted as shown in Fig.8a and fed to subsequent point cloud segmentation algorithms.

The point cloud segmentation under a certain distance threshold is used [28] which use distance information to combine neighboring points, thereby separating the point cloud to different isolated regions. The distance-based clustering method divides the point cloud into N classes [30]. At the time of initial detection, the octree of point cloud has been created. The points and nodes of defects are reserved to this step as shown in Fig.8b and octree nodes are used to represent defects. At first, a random node is selected as the seed node and the nodes in the threshold distance are selected into a group. Each node is

compared to the cluster in each iteration step, and the cluster centers change when absorbing nodes in the threshold distance. The iteration will continue until the number of remaining nodes is unchanged and these nodes are regarded as a cluster. Then, the next seed node is selected and the above steps are repeat for other nodes. The effect of point cloud segmentation is shown in Fig.8c.

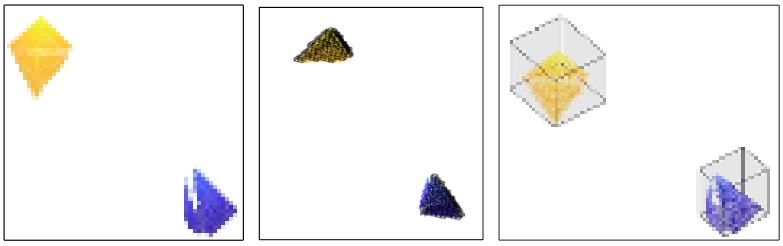


Fig.8 The procedure of point cloud segmentation. **a** the defects from coarse detection; **b** the nodes of defects; **c** the result of point cloud segmentation

In the previous steps, the defects are extracted roughly by coarse detection and separated by point cloud segmentation. The information of each defect including the location and size will be calculated and the boundary box as shown in Fig.9a is acquired. The defects from coarse detection are not whole which means the boundary box is smaller than true value and the boundary box is enlarged by a node size. Then the point in boundary box is extracted as shown in Fig.9b and Fig.9c to detailed extraction according to the algorithm of Defects Detection. Due to the size of the defect is different, the fix size of cube can't adapt to all defects. Therefore, a fixed number of iterations is chosen as the stop criteria of octree and the maximal leaf nodes size is also the accuracy of detection. Therefore, the point cloud of defect area is extracted as shown in Fig.9d, which includes two parts, first one is from the damaged point cloud model, second one is from the nominal point cloud. The fine defects detection process based on octree is illustrated as follows.

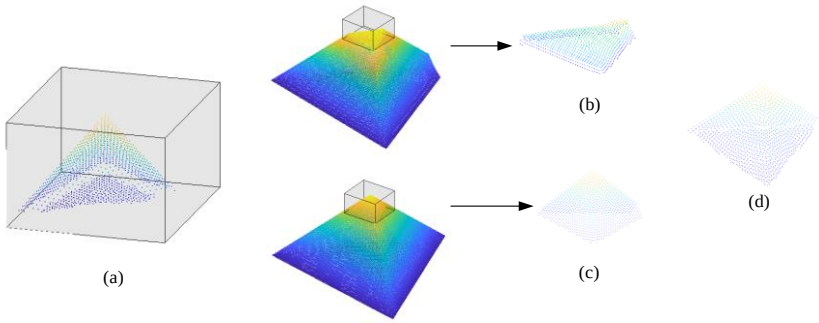


Fig.9 The Detailed Extraction procedure of Defect I. **a** the boundary box of Defect I; **b** the damaged point in boundary box; **c** the nominal point in boundary box; **d** the result of detailed extraction

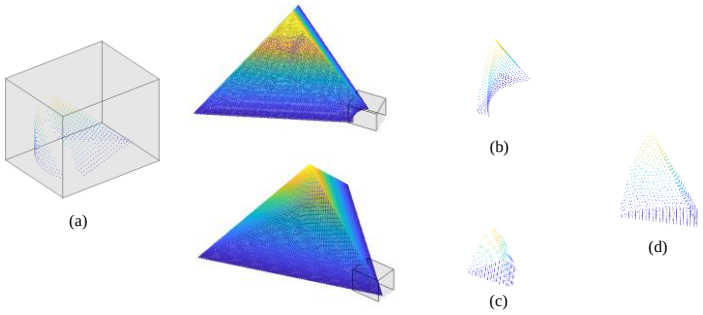


Fig.10 The Detailed Extraction procedure of Defect II. **a** the boundary box of Defect I; **b** the damaged point in boundary box; **c** the nominal point in boundary box; **d** the result of detailed extraction

3. Case Study

Aero-engine blades are typical high-value metallic components. The remanufacturing of aero-engine blades has been studied widely due to the benefits in economy, environment, and society. Therefore, the aero-engine blade is used as case study to detect its defects. For the feasibility study of detecting defects, the defective blade model used for experimental purposes were created artificially. This model was developed by generating "virtual defects" into a non defective model using Solidworks [20]. Based on the characteristics of blade defects, two typical defects (tip missing and abrasion) [21] are generated on the blade. The feasibility of the developed method is validated implemented in MATLAB, which run on a PC with a 1.60 GHz Intel Core CPU.

3.1 Defects Detection

To mimic such defects, a turbine blades CAD model was created in Fig.11a and defects were generated on the turbine blades in Fig.11b. In actual application, the damaged model is scanned by laser scanner to acquire the damaged point cloud. In this case, the damaged CAD model is converted to point cloud for following work an shown in Fig.11d. The nominal CAD models is converted to point cloud as well in Fig.11c. The registration operation of point clouds was required for detecting defects. The results of FPFH initial registration and ICP fine registration are respectively shown in Fig12b and c.

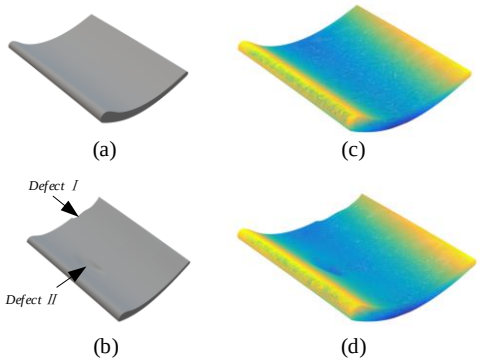


Fig.11 The turbine blade model. **a** the nominal CAD model; **b** the damaged CAD model; **c** the nominal point cloud; **d** the damaged point cloud

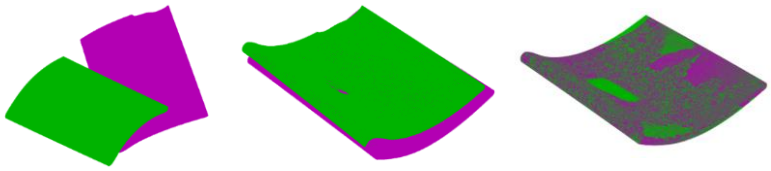


Fig.12 The registration process between the nominal and damaged point cloud. **a** the position before registration; **b** the result of initial registration; **c** the result of fine registration

After registration, the two-step approach is used to extracted defects of blade. At first, the octree of nominal point cloud is created to build topology in Fig.13a and the minimal cube size is set to 0.5. The spatial structure of octree is extracted to apply to point cloud of damaged blade in Fig.13b, and the point out of the structure and empty nodes are filtered. The intersection points in empty nodes and nominal model are part I in Fig.13c. The point out of the structure is part II in Fig.13d. The result of coarse detection is the combination of part I with part II as shown in Fig.13e and segmented to two defects in

Fig.14b. In order to ensure extraction accuracy, the boundary box is enlarged by a node size. Then the point in boundary box is extracted as shown in Fig.15b and Fig.15c to detailed extraction according to the algorithm of Defects Detection. a fixed number of iterations 5 is chosen as the stopping criteria of detailed extraction. The detailed extraction result of Defect I is shown in Fig.15f. The situation of Defect II is shown in Fig.16.

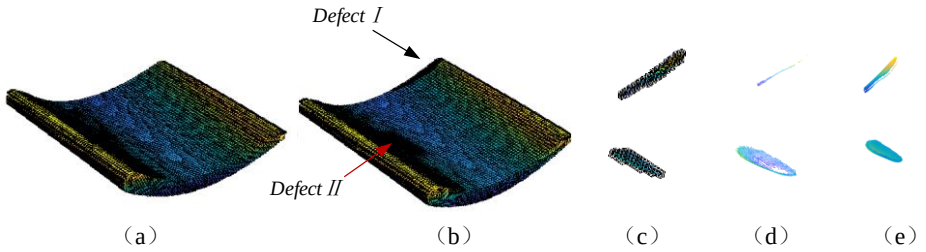


Fig.13 The blade defects detection procedure. **a** the spatial structure of octree for nominal point cloud; **b** the spatial structure with damaged point cloud; **c** part I of defects area; **d** part II of defects area; **e** result of coarse detection

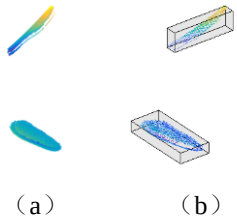


Fig.14 The result of point cloud segmentation. **a** the defects before segmentation; **b** the segmented defects

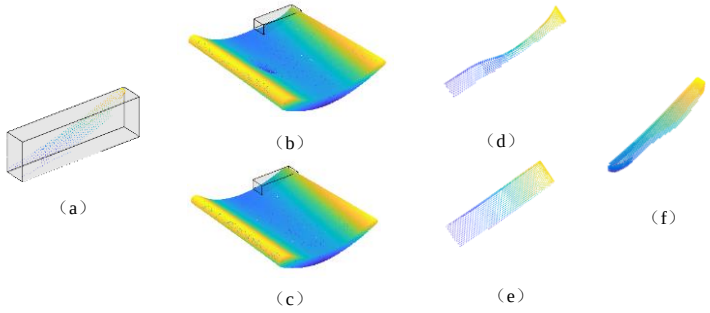


Fig.15 The blade detailed extraction procedure of Defect I. **a** the boundary box of Defect I; **b** the boundary box applied to damaged point cloud; **c** the boundary box applied to nominal point cloud; **d** the damaged points in boundary box; **e** the nominal points in boundary box; **f** the result of detailed extraction

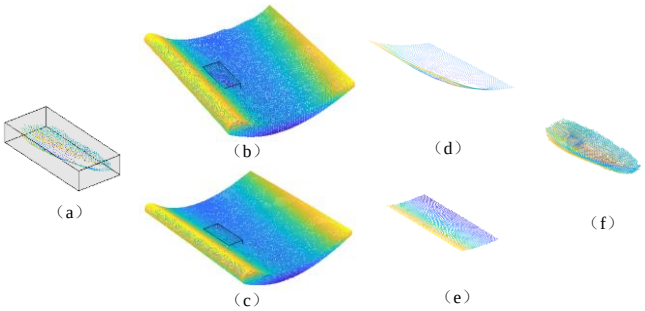


Fig.16 The blade detailed extraction procedure of Defect II. **a** the boundary box of Defect II; **b** the boundary box applied to damaged point cloud; **c** the boundary box applied to nominal point cloud; **d** the damaged points in boundary box; **e** the nominal points in boundary box; **f** the result of detailed extraction

3.2 Generation of tool path

Once the surface defects are obtained, the restoration tool path can be generated to repair the damaged components to their original performance. There are three steps for repairing. The first stage is the pre-processing of defects area to provide a smooth surface area and a tool-accessible area. It is because defects in components are often found in the form of cracks, dents, burrs, and abrasions. The damaged area is always small so that the cracks are not accessible by additive manufacturing tool. In the second step, the tool path of AM is necessary to deposit material into the pre-machined area to return the components to original shape [27]. In the third step, the tool path of the post process is necessary to polish or grind the AM surface to meet the quality requirement.

Existing methods for tool path generation all need to reconstruct surface to STL which is an intermediate form of 3D models by computer-aided manufacturing software. The repairing process is complicated and inefficient because the reconstruction process requires a large amount of computation recourse. With the development of advanced machining, the technology of constructing machining trajectories directly with measured data has been gradually developed in recent years [22-26]. The codes based on above researches were wrote by MATLAB to generate tool paths directly from the point cloud. The tool-paths were formatted into machine-readable g-code for experimental validation.

The implementation of the pre-process tool paths is shown in following four steps. The pre-machining toolpath of blades is shown in Fig.17c and d.

Step 1: Identification of the smallest bounding of pre-processing based on the defect point cloud. The bounding should meet the defined inclination angle and filleted corner to facilitate additive manufacturing;

Step 2: Slice the bounding of pre-processing. Identification of the 2D bounding of each layer waiting to be processed;

Step 3: Splitting the bounding with equidistant lines. Find intersection points between all the lines with the boundary of the 2D area;

Step 4: Connection of intersection points of each neighbor to form a continuous path.

Based on the pre-processed model, the processing area of AM is the boundary after pre-processing. Similar to the process of pre-machining tool path, the tool path of AM is generated by slicing, splitting the boundary with equidistant lines and connection of points. The AM toolpath of blades is shown in Fig.17e and f.

Even though the AM process can restore the shape of the damaged components, the surface quality of AM process is still far from satisfactory. It is necessary to polish or grind the AM surface to meet the quality requirement. The tool path of post process is generated based on the nominal point cloud. Based on cross section algorithm [25], points are adjusted and arranged. The tool path could be generated by sorting all points by layers and connecting them in turns. The toolpath of post processing blades is shown in Fig.17g and h.

The proposed toolpath generation algorithm was verified through virtual repairing the defective turbine blade in the VERICUT software. No CAD or CAM software is used in the process of tool path generation. The extraction of defect area and generation of the tool paths do not require additional manual intervention.

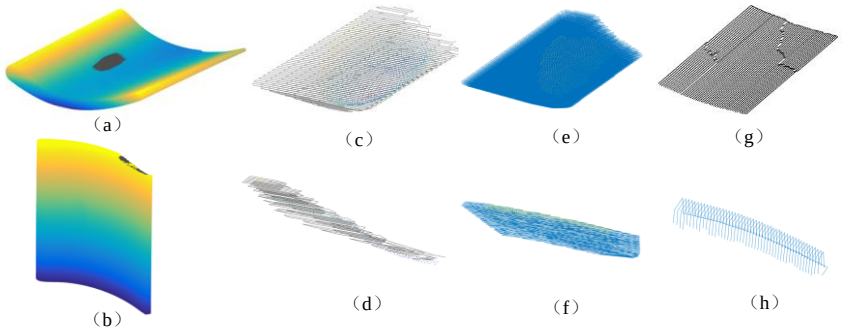


Fig.17 Generation of tool path for repairing blade. **a** the blade to repair Defect II; **b** the blade to repair Defect I; **c** pre-machining process of Defect II; **d** pre-machining process of Defect I; **e** AM process of Defect II; **f** AM process of Defect I; **g** post process of Defect II; **h** post process of Defect I.

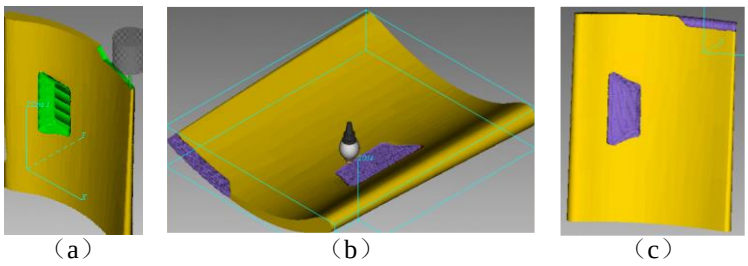


Fig.18 The simulation result of tool path in VERICUT. **a** the simulation result of pre-machining; **b** the simulation result of AM process; **c** the simulation result of post process.

4. Results and Discussion

Two experiments are designed to validate the accuracy and effectiveness of the proposed defect detection method. In the first experiment, the result of the proposed method is compared to the other two methods to validate the detection time and detection accuracy of the method. The second experiment proves the adaptability and stability of the proposed algorithm to different parts by detect multiple damaged components.

4.1 Comparison and Verification of detection time and accuracy

To compare and verify the detection time and accuracy of the developed octree-based detection method, the developed method is compared to the voxel-based method [14] and the distance-based filtering method [17]. Feng et al. [17] demonstrated a distance-based method to detect and extract defects. First, for a point pair, a point-to-point distance between one point in the nominal model and another point in the damaged model is estimated. Furthermore, an average distance among all the matching point pairs is computed and set a distance threshold that is usually 3 to 5 times the average distance. All the point pairs which are greater than the distance threshold are retained to represent the area of the defects. The accuracy of distance-based method is the distance threshold, which is also the minimum defect size that the method could detect. Zhang et al. [14] introduced a novel defects extraction methodology based on voxel modeling. Nominal and damaged models are converted to STL format. Then STL models are transferred to voxel models. For any voxel of the nominal model, the operation is conducted to check if the voxel of the damaged model belongs to it or not. In this way, all the voxels that do not belong can form the damaged volume. The accuracy of voxel-based method is the size of voxel, which is also the minimum defect size that the method could detect. The size of voxel is pre-set. The accuracy of developed method is the size of leaf nodes which is also the minimum defect size that the method could detect. The

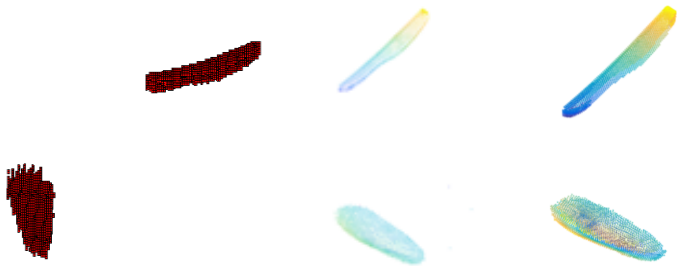
result of above methods is shown in Fig.19. All the cases are conducted with MATLAB 2016a on Intel(R) processor (1.60-GHz clock).

To validate the effectiveness of our proposed method, the proposed method is compared with the voxel-based method and distance-based method in terms of time. In Fig.20, the runtime of above methods to process different point clouds is compared at the same accuracy (0.3mm). As can be seen from the graph, the developed method takes less time than others at each point cloud. It is because the proposed method is based on octree, which created the topology of discrete points by node-to-node relationships. However, the voxel-based method and the distance-based method create topology by point-to-point relationships. The number of octree nodes is dramatically smaller than the number of points. Hence, the creation of topology for discrete points is faster than other methods. Secondly, a two-step defects detection approach is used including coarse detection and detailed extraction. By this mean, unnecessary time processing the undamaged area is also reduced.

In addition, as the number of points increases, the advantage of the developed method is more obvious. The time of the voxel-based method is greatly increased while the time of the proposed method is not affected for example the time of the voxel-based method is 81 times more than the time of the proposed method at the point cloud having 1073457 points and the time of the distance-based method is 3 times more. This is owing to that the octree structure is not affected by point cloud density. Therefore, the developed method could process the point cloud collected by 3D scanning devices and it is not necessary to point cloud simplification.

In Fig.21, the runtime of the above methods to process point clouds is compared at different accuracy (1.5mm, 0.75mm, 0.3mm and 0.2mm). As can be seen from the graph, the developed method takes less time than others at each accuracy. As the accuracy increases, the time of the voxel-based method is increased dramatically which is unacceptable in practical application. But the trend of the proposed method is gentle which is always within the acceptable range.

To reduce the detecting time, point cloud simplification methods were widely used to reduce the number of the scanned points. However, it is found that point cloud simplification will result in error of surface detection. To validate the accuracy of our proposed method as well as the corresponding processing time, the proposed method is compared with four ways, they are the voxel-based method on original point cloud, the voxel-based method on simplified point cloud, the distance-based method on original point cloud and the distance-based method on simplified point cloud in Fig.22. The error is evaluated by comparing the volume of the detected defect which is acquired by Solidworks. As in Fig. 22, the ways 1-4 can not achieve the optimal accuracy and time simultaneously. By comparison, the developed method has less time and error than other four ways. As shown in the graph, except for our methods the third way is optimal on original point cloud and compared with it [17], the time of developed method is improved by 74.03% and the error is reduced by 36.86%.



（a） （b） （c）

Fig.19 The result of mentioned methods. **a** the result of voxel-based method; **b** the result of distance-based method; **c** the result of octree-based method.

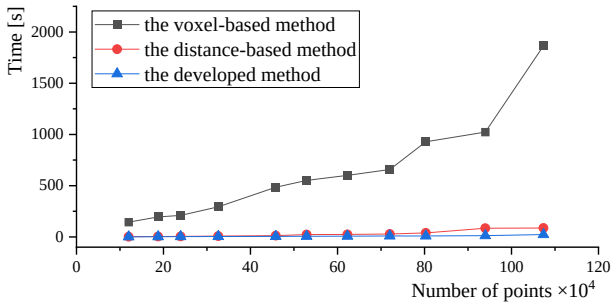


Fig. 20 The comparison of time at processing different point cloud

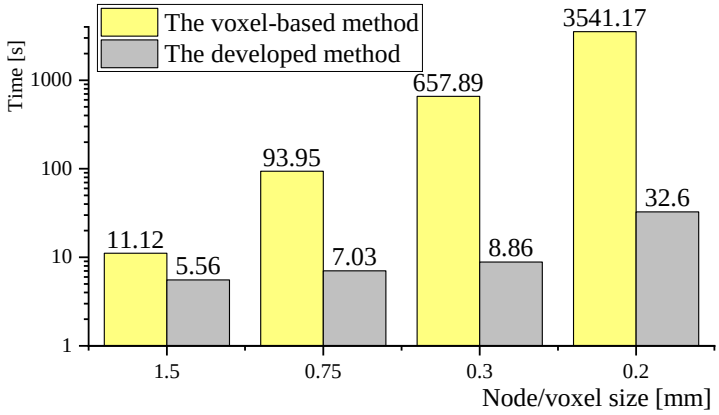


Fig. 21 The comparison of time at different accuracy

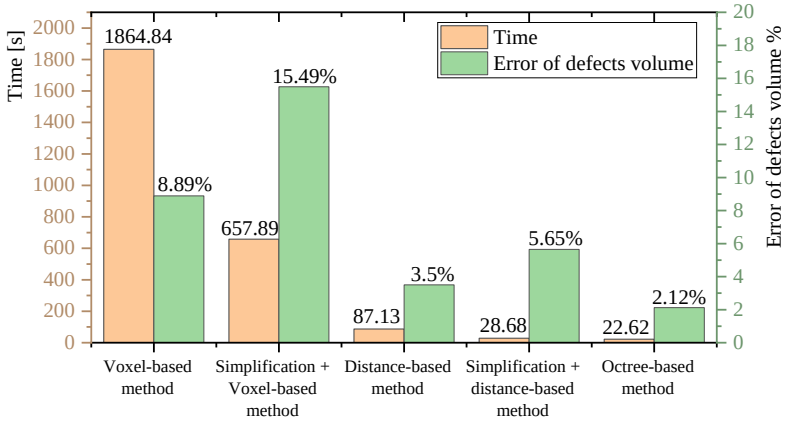


Fig.22 The comparison of time and error of five ways

4.2 The detection experiment of multiple parts

To support the stability and effectiveness of this algorithm for different parts, a variety of different damaged parts were detected. An actual damaged gear in Fig.23a is used to detect defects and the damaged point cloud was obtained through scanning. The nominal model of gear is shown in Fig.23 b. The damaged point cloud and nominal point cloud are aligned and the result of alignment is shown in Fig.23 d. The result of detecting damaged gear is shown in Fig.23 e. An actual turbine rotor is shown in Fig.24 a and the damaged model is shown in Fig.24 b. The damaged model was aligned with the nominal model (Fig.24 c) and the defect was extracted as shown in Fig.24 e. Similarly, the defects detection

procedure of an engine end cover is shown in Fig.25. Octree as a spatial data structure create topology additionally and is not affected by the shape of the point cloud. Therefore, the defects detection method based on octree is suitable for general components. As seen, the developed method not only provides accurate and quick detection but also provides an adaptive method for different damaged components. Thanks to the octree structure, point cloud topology is created additionally as a spatial data structure. The structure can be well adapted to different parts. Therefore, the defects detection method based on octree is suitable for general components.

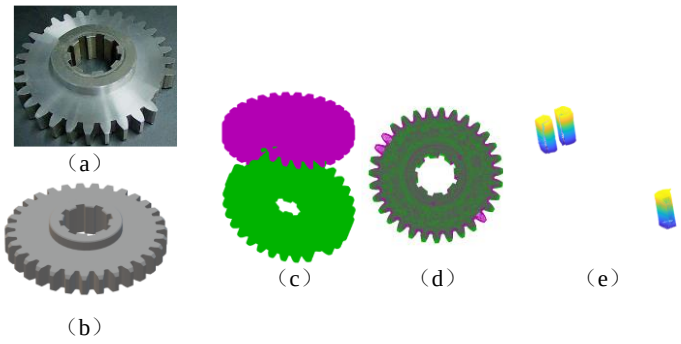


Fig.23 The defects detection of gear. **a** the damaged gear; **b** the nominal model of gear; **c** the point cloud before registration; **d** the result of registration; **e** the result of defect extraction

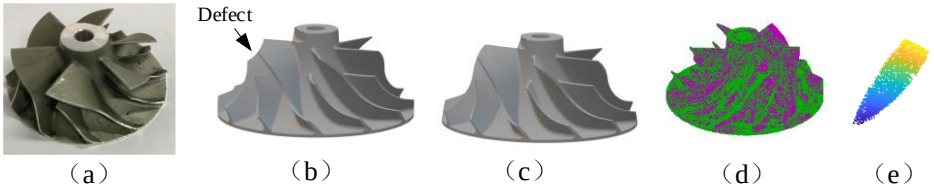


Fig.24 The defects detection of turbine rotor. **a** the turbine rotor; **b** the damaged turbine rotor model; **c** the nominal turbine rotor model; **d** the result of registration; **e** the result of defect extraction

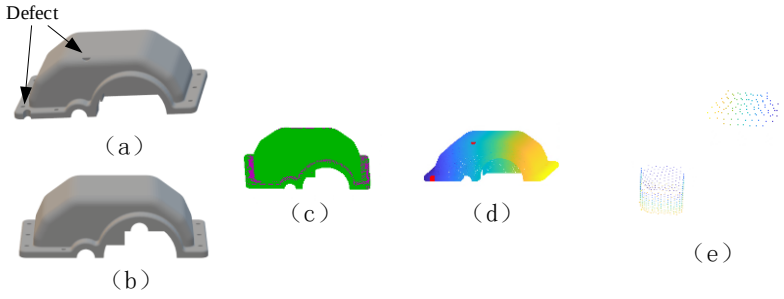


Fig.25 The defects detection of engine end cover. **a** the damaged end cover model; **b** the nominal end cover model; **c** the result of registration; **d** the highlight of defects; **e** the result of defect extraction

5. Conclusions

Accurate and quick detection has a significant bearing on the overall productivity of remanufacture. In this paper, a two-step method based on octree was proposed to detect and extract defects with stable performance as well as high accuracy and efficiency. In the proposed method, octree is introduced to create topology by node-to-node relationship and therefore it can easily process the collected dense point cloud. The two-step method is proposed to reduce the unnecessary time processing the undamaged area. This provides the technique with higher accuracy and effectiveness to be applicable to high-density point clouds and high-precision requirements of remanufacturing. Compared with the state-of-art methods, the

time of the developed method is dramatically reduced by 74.03% and error is reduced by 36.86%, which can accelerate the detection time and thereby improving the remanufacturing productivity. In addition, the extracted defects point cloud can be further used to generate the restoration tool path for remanufacturing directly, establishing the foundation for actual repairing.

Acknowledgements

This study was supported the National Natural Science Foundation of China (Grant No.52175453); the Science Fund for Distinguished Young Scholars of Chongqing (Grant No. cstc2020jcyj-jqX0011); Chongqing General Program of Natural Science Foundation (Grant No. cstc2020jcyj-msxmX0639).

Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. Du, Y. , Cao, H. , Fei, L. , Li, C. , & Xiang, C. . (2012). An integrated method for evaluating the remanufacturability of used machine tool. Journal of Cleaner Production, 20(1), 82-91.
2. Li, Yiting, Huang, Haisong, Xie, & Qingsheng, et al. (2018). Research on a surface defect detection algorithm based on mobilenet-ssd. Applied Sciences.
3. Wilson, J. M. , Piya, C. , Shin, Y. C. , Fu, Z. , & Ramani, K. . (2014). Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis. Journal of Cleaner Production, 80(oct.1), 170-178.
4. Yza, B. , Jin, L. , Hw, C. , Mr, D. , Mza, B. , & Wla, B. , et al. An intelligent and automated 3d surface defect detection system for quantitative 3d estimation and feature classification of material surface defects. Optics and Lasers in Engineering, 144.
5. Zhang, X. , Cui, W. , & Liou, F. . (2021). Voxel-based geometry reconstruction for repairing and remanufacturing of metallic components via additive manufacturing. International Journal of Precision Engineering and Manufacturing-Green Technology.
6. Rannou, G. , Thierry, D. , & Lebozec, N. . (2010). Ultrasonic monitoring of steel corrosion during accelerated.
7. Zolfaghari, A. , Zolfaghari, A. , & Kolahan, F. . (2018). Reliability and sensitivity of magnetic particle nondestructive testing in detecting the surface cracks of welded components. Nondestructive Testing & Evaluation, 1-11.
8. Zolfaghari, A. , & Kolahan, F. . (2017). Reliability and sensitivity of visible liquid penetrant ndt for inspection of welded components. Materialprufung, 59(3), 290-294.
9. Xu, C. , Guo, J. , Zhou, X. , Chao, W. , & Dong, L. . An edge sensitive 3D measurement using two directional laser stripes scanning with a laser projector.
10. Skrodzki, M. , Jansen, J. , & Polthier, K. . (2018). Directional density measure to intrinsically estimate and counteract non-uniformity in point clouds. Computer Aided Geometric Design, 64(AUG.), 73-89.
11. Yilmaz, O. , Gindy, N. , & Jian, G. . (2010). A repair and overhaul methodology for aeroengine components. Robotics & Computer Integrated Manufacturing, 26(2), 190-201.
12. Wilson, J. M. , Piya, C. , Shin, Y. C. , Fu, Z. , & Ramani, K. . (2014). Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis. Journal of Cleaner Production, 80(oct.1), 170-178.

13. He, J. , Li, L. , & Li, J. . (2011). Research of Key-Technique on Automatic Repair System of Plane Blade Welding. IEEE.
14. X Zhang, Cui, W. , & Liou, F. . (2021). Voxel-based geometry reconstruction for repairing and remanufacturing of metallic components via additive manufacturing. International Journal of Precision Engineering and Manufacturing-Green Technology.
15. Zhang, X. , Li, W. , Adkison, K. M. , & F Liou. (2018). Damage reconstruction from tri-dexel data for laser-aided repairing of metallic components. The International Journal of Advanced Manufacturing Technology.
16. Liu, T. , Wang, N. , Fu, Q. , Zhang, Y. , & Wang, M. . (2019). Research on 3D Reconstruction Method Based on Laser Rotation Scanning. 2019 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE.
17. Chao, F. , Jin, L. , Gong, C. , Wenyan, P. , & Liu, S. . (2018). Repair volume extraction method for damaged parts in remanufacturing repair. International Journal of Advanced Manufacturing Technology, 1-14.
18. Shi, B. Q. , Jin, L. , & Liu, Q. . (2011). Adaptive simplification of point cloud using k-means clustering. Computer-Aided Design, 43(8), 910-922.
19. Chien, C. H. , & Aggarwal, J. K. . (1986). Volume/surface octrees for the representation of three-dimensional objects. Computer Vision Graphics and Image Processing, 36(1), 100-113.
20. Piya C , Wilson J M , Murugappan S , et al. Virtual Repair: Geometric Reconstruction for Remanufacturing Gas Turbine Blades[C]// Asme International Design Engineering Technical Conferences & Computers & Information in Engineering Conference. 2011.
21. Bremer C . Automated Repair and Overhaul of Aero-Engine and Industrial Gas Turbine Components[C]// Asme Turbo Expo: Power for Land, Sea, & Air. 2005.
22. Xiao, W. , Liu, G. , & Zhao, G. . (2020). Generating the tool path directly with point cloud for aero-engine blades repair. Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture, 235(2), 095440542097091.
23. Lin, A. C. , & Liu, H. T. . (1998). Automatic generation of nc cutter path from massive data points. Computer-Aided Design, 30(1), 77-90.
24. Zou, Q. , & Zhao, J. . (2018). Iso-parametric tool path planning for point clouds.
25. Biegler, M. , Wang, J. , Kaiser, L. , & Rethmeier, M. . (2020). Automated tool-path generation for rapid manufacturing of additive manufacturing directed energy deposition geometries. steel research international.
26. Elseberg, J. , Borrmann, D. , & Nuechter, A. . (2013). One billion points in the cloud: an octree for efficient processing of 3d laser scans. ISPRS Journal of Photogrammetry and Remote Sensing, 76(FEB.), 76-88.
27. Renwei, Liu, Zhiyuan, Wang, Frank, & Liou. (2018). multifeature-fitting and shape-adaption algorithm for component repair. Journal of manufacturing science and engineering: Transactions of the ASME, 140(2), 1-19.
28. Lowe, D. G. . (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110.
29. Li, P. , Wang, J. , Zhao, Y. , Wang, Y. , & Yao, Y. . (2016). Improved algorithm for point cloud registration based on fast point feature histograms. Journal of Applied Remote Sensing, 10(4), 045024.
30. Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking points with labels in 3D: A review of point cloud semantic segmentation. In: IEEE Geoscience and Remote Sensing Magazine (Vol. 8, no. 4). pp. 38–

59. https://doi.org/10.1109/MGRS.2019.2937630.