Surface Reconstruction by Triangular Mesh

Of models with Uneven Density and with Hole

A thesis
submitted in fulfilment
of the requirements for the degree
of
Master of Engineering
at
The University of Waikato
by
YE CHEN

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2015

Abstract

With the fast development of computer software, 3D computer aided design (CAD) and computer aided manufacturing (CAM) are being improved rapidly and have been commonly used in industry, especially in medical area and moulding field. 3D CAD/ CAM has greatly shortened the mould development time, thus it plays an important role in product development and automatic production.

It has been a serious topic about how to get the product design and also being manufactured in a short time to grasp the market opportunities in the competition of the developing manufacturing techniques, diversification and complexification of product demand. But not all the products have the design draft, for example, some of them were made by the mould of clay which came from craftsman by their hands. In order to conquer those problems, reverse engineering came into being. Reverse engineering could achieve a faster product design and manufacture, it also could be used in the field of vehicle components, home appliances, medical supplies (like human body measurement such as for artificial leg bones, tooth, etc.,), and rapid prototyping (such as precision casting products, industrial design, etc.). The so-called reverse engineering now targets to the existing mould (samples or manual model). It starts with the existing model, sampling surface data by data acquisition equipment (DAE) then acquire an ordered or unordered could of points. After that by the use of discrete geometry processing technology to get the CAD model of solid entity which computer could recognize. The following procedures of the CAD model could be redesigned or improved as required.

Acknowledgements

I want to say thank you for many people during my master studying period. First of all, thank my parents for their unconditional support and encouragement throughout my whole life.

I would like give my deepest and honest thanks to my academic supervisor Dr. Chi Kit Au, for his great patience and kindness for the whole year and all the academic support. Especially for those helpful assistance when I got lost in the studying.

There are some other senior students Lu Huiyang, Lou Jia, Qin Xin, Ren Ran and Navaneeth, I do appreciate their generous of sharing studying experience to me.

And thank administrators Mary Dalbeth and Janine Williams for supporting of office-use materials and dealing with my trivial affairs.

Table of Contents

List of Figures

List of Tables

Chapter 1:

Introduction

Chapter 1:

Introduction

3D (three-dimensional) technology is a broad definition, any processing relevant to visually perceived to depth and distance are all 3D technology. Basically, it is a technology to deal with voxels. With the powerful computing ability of computers, how to “move” the subjects existing in real world into computer to solve the problems is the major breakthrough. In fact, the computer aided technologies are focusing on dealing with those problems.

The existing common geometric modelling techniques used in CAD are [1] :

1) Wireframe model, the usage of basic line elements, like straight lines, circular features, points and free curves to determine the target prism part to form the 3D wireframe draft.

2) Surface modelling, the usage of basic surface elements, like plane surface, quadric surface and free-form surface to describe every surface of the solid and to construct solid model.

3) Solid modelling, by defining basic voxel, such as cuboid, sphere, cylinder, cone and torus, etc. to acquire the entity internal computer to represent the existing model by voxel set operations and elementary transformations.

4) Feature modelling, based on solid modelling, add more geometrical information about shape tolerance, size tolerance, surface roughness and material performance.

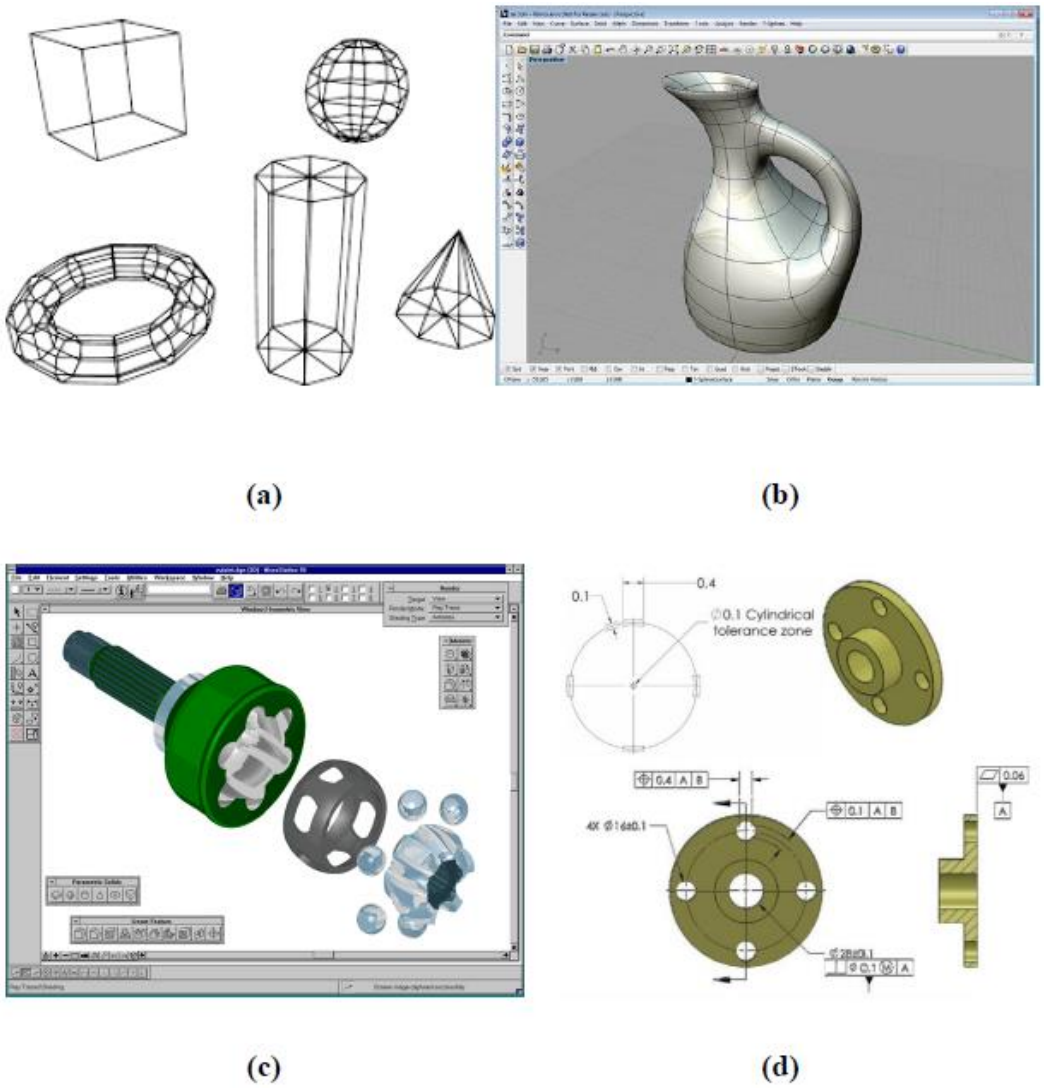


Figure 1. 1 Wireframe model (a) [2]; Surface modelling (b) [3]; Solid modelling (c) [4]; Feature modelling (d) [5]

Those methods share one characteristic in common that there is no solid entity in the beginning. Technicians design the entity step by step, from simple model to complex by basic elements of line, surface or voxel. That is called the forward engineering.

There is another geometric modelling technique in CAD called reverse engineering. In the 1960s, reverse engineering had been a newly-developing independent research area in industry. In 1959, British company Ferranti

developed the first coordinate measuring machine in the world and exhibited at the International Machine Tool exhibition in Paris [6]. In 1963, DEA which is a world famous premier brand now delivered the first stationary gantry coordinate measuring machine and initiated the new frontier of coordinate measuring technology [7]. Since then reverse engineering has developed as a relatively independent research branch. The main technique of reverse engineering to reconstruct subject's surface is by 3D computer-aided modelling. Firstly, acquire contour coordinates by digital measuring instrument, then construct, edit or modify the surface feature, after that transfer data into CAD/CAM system. The mould will be produced by CNC (Computer Numerical Control) machine of which the play path created by CAM or manufactured by rapid prototyping. STL (Stereolithography) is another file format generated by 3D CAD system. It is a swap file showing the surface triangulating of solid surface. In reverse engineering area, STL is one of the main data output file from measuring equipment. Due to STL is simple formatted file, easy to read and show, some applications based on STL are expended, like forming finite element mesh or generating process path directly from STL [8] [9].

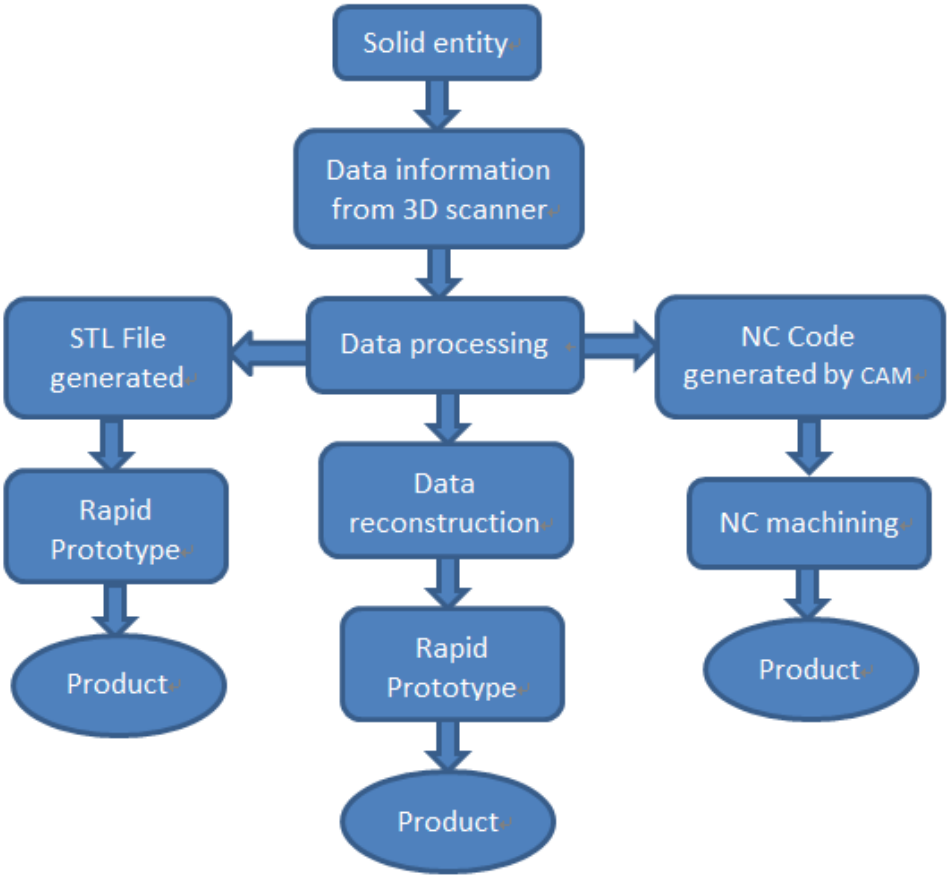


Figure 1. 2 Schematic diagram of reverse engineering

1.1 Motivation

In practical use, 3D computer-aided modelling has the advantages of high-efficiency and economical. For manufacturing most products, it takes a long period from model design to the acceptance test, and normally after that those models still need to be modified. The cost increases with the time spending on modifying moulds, especially for the small-lot production, mould investment occupies an even bigger amount of the total cost. Therefore, the best way to manufacture products of short production period and small-lot production is combing reverse engineering and rapid prototype. This combination could complete complicated mould and components within a short time and the more

complicated those components are, the more superior the combination works.

Technically, 3D computer-aided modelling could reconstruct the 3D model which is fully compatible with the existing CAD/CAM system by the measuring data, which is the ultimate goal of reverse engineering. But in the practical use including in the project and theoretical, this goal could not be perfectly achieved yet. Especially for large scale of point cloud data modelling, it is far away from to be used in CAD system directly. Thus, 3D computer-aided modelling and the existing CAD/CAM system are complementarily to each other. The CAD/CAM system have been developed for decades, it is nearly complete from both theory and practical application. It is impossible to ground-up redesign CAD/CAM just to satisfy the needs of reverse engineering modelling. On the other side, the plenty work of reverse engineering modelling could refer the CAD/CAM system instead of creating a new platform. Based on the analysis above, 3D computer-aided modelling could take the advantage of the existing CAD/CAM to meet its needs which it could not achieve by itself. And now, most of the CAD/CAM systems contain the reverse engineering module. There are also some reverse engineering software in the market, but they still have many shortcomings.

The main shortcomings of the current reverse engineering software:

1. Most systems of the software are developed for specific applications, which mean the data processing is particularly for the certain measurement equipment and subject with poor universalizable.

2. Surface fitting system applies to most quadratic algebraic surfaces, but to free surface, especially for those free curved surface fitting of large amount of disordered points, most of the surface fitting system does not have that faction.

3. The segmentations of data area need interactive operations which lower

the modelling speed and automation.

4. The low performance of system integration. For example, some systems only focus on surface fitting and the others lean on bonding with specific manufacturing.

In many simulation models, especially in medical use models, deformations of the model occurred commonly by environment disturbance. That is the reason we need a system to simulate possible deformation of the original solid entity. 3D mesh of surface of the solid entity is a good way to solve this problem. 3D mesh could connect all points on the surface or internal, build up a correlation between data points. When the solid entity performs deformations from environment disturbance, the correlations between points are settled. We could simulate what would happen in real occasions by that kind of system. But so far, none of the existing software could achieve it perfectly.

1.1.1 Example of 3D computer-aided modelling in artificial tooth

The traditional dental restorations system is using indirect method which makes tooth model and succedaneum waxwork out of patient's mouth and then gaining restoration by dewaxing casting. It costs at least 9 hours for the whole process of making succedaneum and patient have to wear temporary crown for 1-2 weeks to wait for the preparation of succedaneum. Due to the uncomfortable of temporary crown, the pre-treated tooth always gets inflammation and swelling even been in displacement in some cases. The effect of new succedaneum will be affected by those factors. Besides, patient has to see dentist at least twice, the first time for making tooth model and second time for wearing succedaneum. It takes much

longer than the artificial tooth made by 3D modelling. It divided artificial tooth making into three processes, diseased tooth voxel date acquisition, succedaneum design and CNC machining. It could be finished within an hour and the temporary crown is unnecessary. Dentist could finish all treatment one time with the patient. Besides diseased tooth voxel date acquisition, succedaneum design and producing almost have achieved automatic production meanwhile succedaneum design still could be adjusted manually that releases complex and complicated artificial tooth making process and labour force. The 3D modelling artificial tooth is precisely close to diseased tooth, and their marginal adaptation to pre-treated tooth could close to 50 *um.* [10]

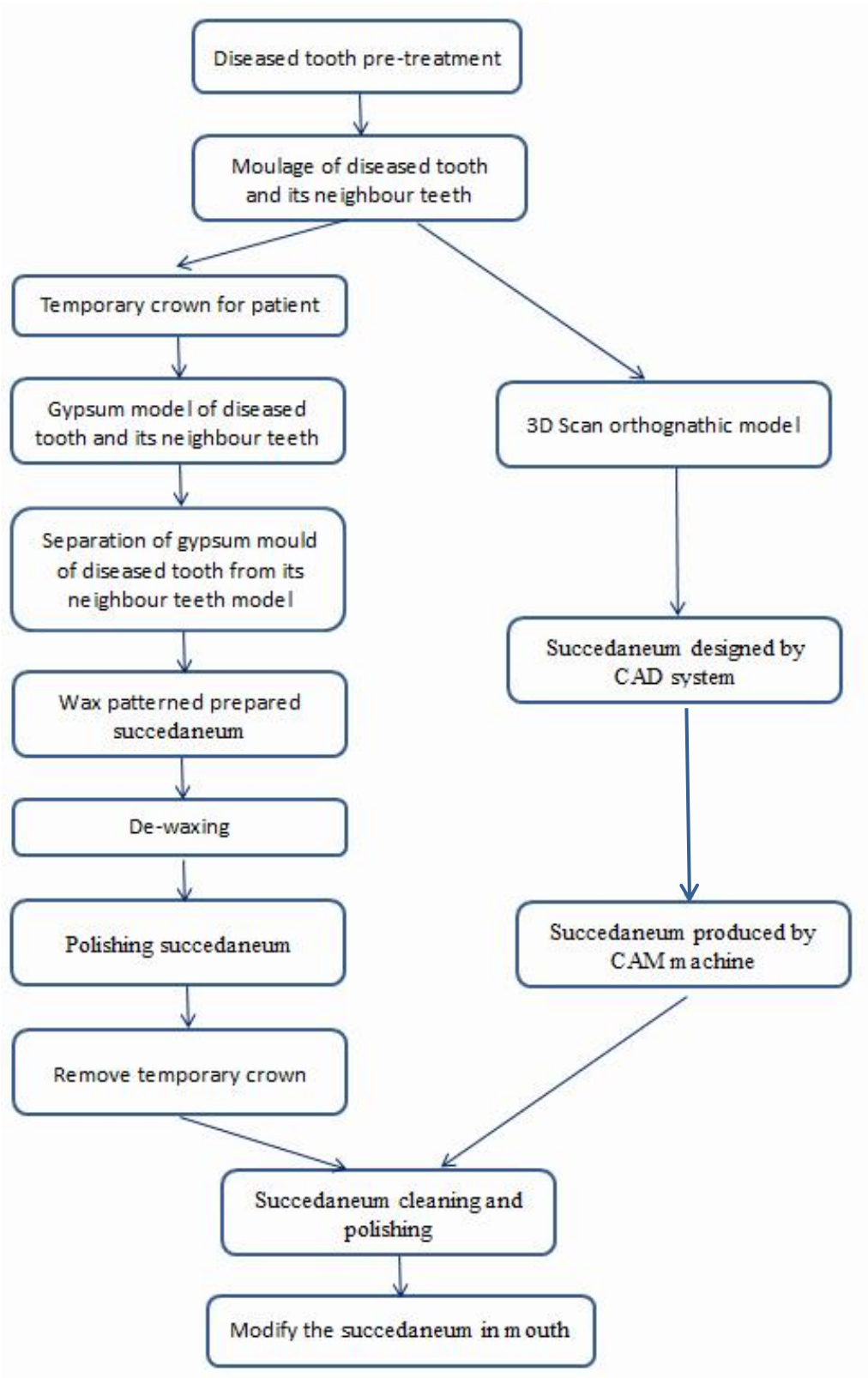


Figure 1. 3 Comparison of two different dental restorations

1.2 Objectives and scopes

This research aims to achieve the reconstruction of the surface of target models by their 3D scanned coordinates in Solidworks API (Application Program Interface) by VBA (Visual Basic for Applications). The most challenging part is how to keep the correct topology after reconstruction and deal with a large number of redundancy data.

The scopes of this research cover the following aspects:

1. Brief introduction of data acquisition equipment
2. Existing algorithm analysis
3. Testing Boundary Extension Algorithm

1.3 Methodology

All the research results are achieved in Solidworks, the programming performs relatively good in both models with flat area and with hole. The method is listed:

- Data acquisition

  The two data acquisition systems are contact system and non-contact system. Contact system is mainly used in measuring the high precise geometric shapes and non-contact system is for measuring complex curved surface. The data sources in this research are from internet by former measurement. [11]

- Point cloud pre-processing

  In this step, a boundary cube will be built to cover all coordinates, and the

point cloud will be divided into same-sized grids. To the point cloud which contains a large number of coordinates will be reduced into a smaller group of coordinates which still keep the main feature of surface but more effective during in program running and manufacture.

- Set seed-triangle

  After being read in Solidworks, the unorganized points could be seen on the screen. Manually choose the seed-triangle in flat area.

- Boundary extension with topological reconstruction

  The initial boundary is the three edges of seed-triangle, the boundary will extend by connecting new points to create new edges followed by certain rules which will be discussed in details in this research. There are two part major part of it, first one is to search for the new point and second one is to correct the topology.

- Analysis the reconstruction efforts by different coefficients and check if there is any major issue in algorithm.

Chapter 2:

Literature Review

Chapter 2:

Literature review

2.1 Data acquisition Introduction

DAE can divide into two types, contact and non-contact equipment[12]. Fig.2.1 is the classification diagram of DAE.

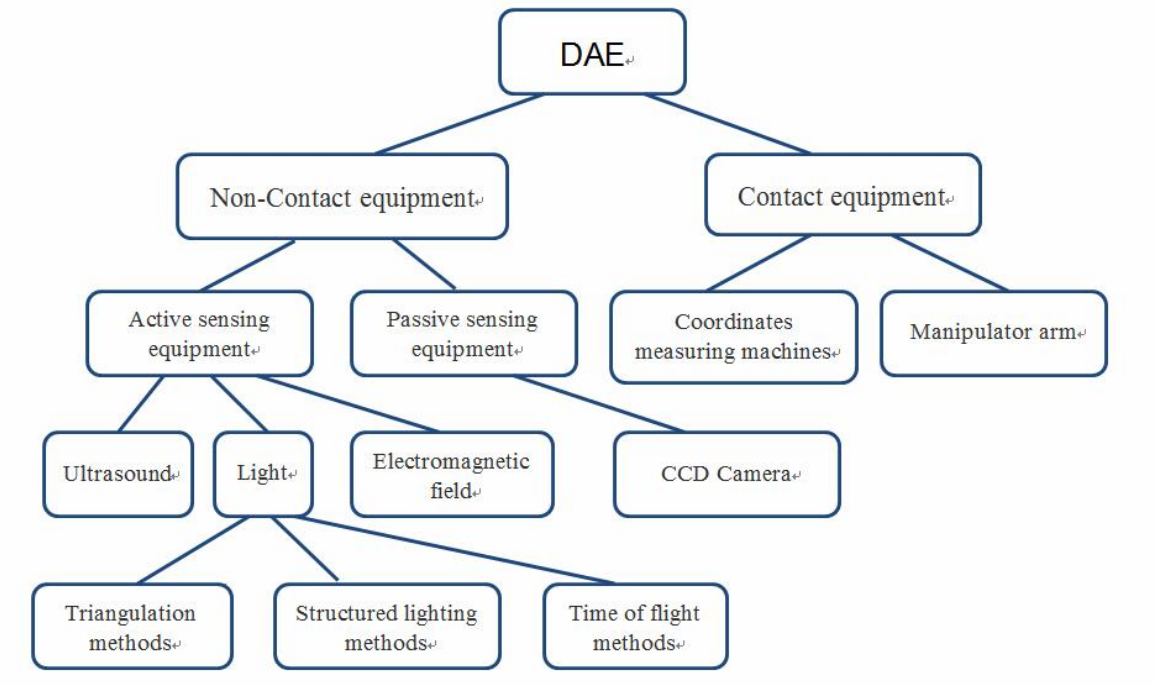


Figure 2. 1 DAE classification diagram

Contact DAE measure data by a detector which was installed in the end of the mechanical arm to contact the surface of objects. Contact DAE has hard probe (also named as mechanical probe), touch trigger probe and analog probe [13]. Hard probe is operated manually on the surface of solid entity. The judgment is based on human eye and feelings, then switch and transmit the coordinates to the processor by foot trigger. Hard probe is still being used because of its low cost. However, it is extremely hard to determine the contact point, even for those experienced technicians, measurement errors are still hard to avoid. Touch trigger probe uses electronic switch system to show the changes of switches when probe contacts the surface of solid entity. Then the coordinates will be locked and sent to the processor. Touch probe is more repeatable and accurate comparing with hard probe because it is controlled by electronic switch which is not easily affected by human factors. Now touch probe is the most commonly used probe in modern 3D coordinate measurement. Analog probe produces side movement when the probe contacts target model. The probe will create a relative voltage change induced by variable coil or grating scale. This analog voltage signal will transmit to digital signal and then sent to the processor. This measurement is called analog measurement. It is a continuously measurement that there is no absolute coordinates so the probe could not leave the surface of solid entity during the measurement. Analog probe can be used more proper on smooth curvature surface but rough curvature surface. Fig 2.2(left) shows the analog probe.

Figure 2. 2 Analog touch probe (left) [14], Bridge coordinate measuring machine(right)[15]

CMM (Coordinates Measuring Machines) is a typical contact DAE shown in Fig 2.2 (right). CMM can move on three orthogonal guide tracks. The probe conveys signal by contacting surface. Displacement measurement system can compute the coordinates on the surface of objects via processor or computer. The mechanical structure and electrical system of contact DAE is quite mature with high accuracy and reliability. With the cooperation of measuring software, contact DAE can measure the object's basic geometrical shape fast and accurately. Basic geometrical shape includes sphere, circle, cylinder, cone etc. But there are also some weaknesses of this equipment including high cost, low speed and easy to damage the surface of objects.

According to energy evaporation, non-contact DAE can be divided into active sensing equipment and passive sensing equipment.[13]

Active sensing equipment would forward energy to the surface of object first, and then calculate the distance between the point on the surface and the reference point by measuring the reflected energy on the surface of object. The sensor of

active sensing equipment could send different kinds of energy, for example, laser, ultrasound, electromagnetic, X-ray and so on. Receiving terminal accepts the reflected energy from the surface of object, and according to physical principles of the energy property, the distance between the points on the surface can be calculated, furthermore, the coordinates could be acquired.

In reverse engineering, people always use laser active sensing equipment. The method mainly includes triangulation methods[16], structured lighting methods[17], time of flight methods [18, 19]and so on.

2.2 Active sensing equipment

2.2.1 Laser measuring methods

- **Triangulation method**

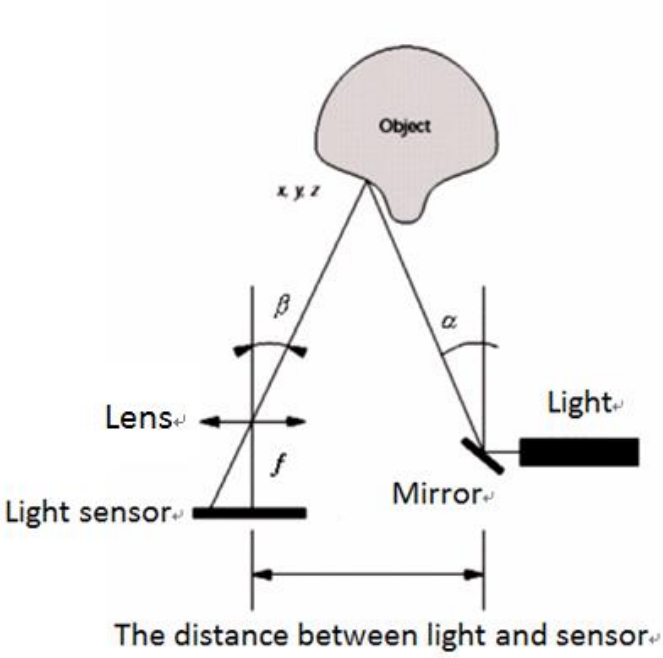


Figure 2. 3 Triangulation method schematic diagram [20]

It deduces the point location on surface by the angle between light and light sensor and their position [21]. Generally there are two different structures, oblique structure and direct light structure. Fig 2.3 is the schematic diagram of oblique triangulation methods. The light from one intense light-source reflected by a mirror incident on the surface of object with an angle α. Reflected light is received by the sensor. Angle β is determined by focal length f and pixel size. Besides, the distance between light source and sensor has already known, point coordinate (x, y, z) could be calculated by triangulation principles. Direct light triangulation method is similar to oblique triangulation method. The only difference is direct light structure vertically incidents light on the surface. DAE triangulation method can collect data in a fast speed, the accuracy of data depends on sensor's resolution and the distance between DAE and the object.

- **Structured light methods**

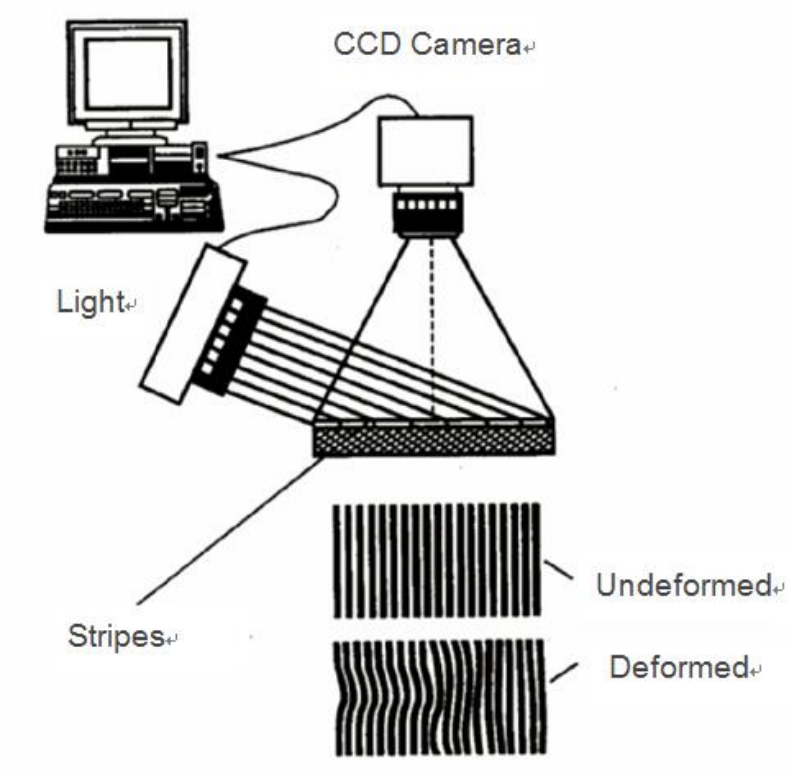


Figure 2. 4 Surface strip-structured light method schematic diagram [13]

Firstly, incident a beam of light, project it on the object and capture the reflected image. Then by analysing the image, the coordinates of subject on surface could be acquired. Structured light method can be divided into two parts, linear structure-light and surface strip-structured light. Linear structure-light method uses stripe light to incident a light plane in the space. The intersecting line of the light plane and the surface is a curve on the plane. The points on the curve are the measurement object. Take one point from the light plane as an original point and build a coordinate system, coordinates could be acquired by the perspective correspondence between light plane and optical receiver equipment. Similarly, surface strip-structured light method uses strip-structured light projector to incident several light planes in space. When these light planes intersect with the objects, and get several curves in a plane. The points on the curves are the

measurement objects. Likewise, the perspective correspondence between light plane and image plane is also a way to get the coordinates. Fig 2.4 shows the schematic diagram of surface strip-structured light method. A large number of point coordinates could be captured in every surface strip-structured light measurement but the process of analysing the reflected image and get coordinates is really complicated.

- **Time travelling methods**

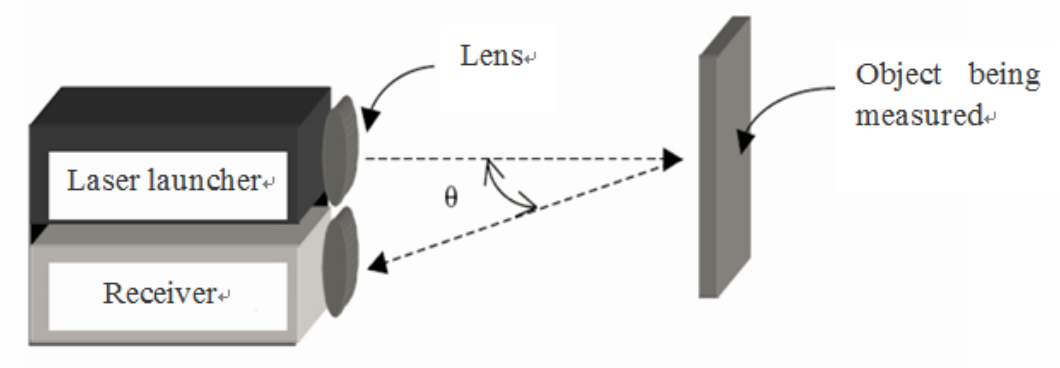


Figure 2. 5 Time travelling method schematic diagram

It calculates the distance between DAE and object by the time cost of light (normally by laser or impulse) travelling to get the coordinates. At beginning DAE based on time travelling method sends electromagnetic radiation, focus electromagnetic radiation to a narrow parallel beam then send to the surface. The reflected beam will be received by photosensitive receiver (such as sensitive diode) in DAE. Meanwhile, clock in measurement will calculate the time from launch to receive of parallel beam in DAE to compute the distance. Fig 2.5 shows the schematic diagram of time travelling method, the angle θ is small enough that it does not affect the distance, it is always negligible. A variation of time travelling method is phase shift method. This method calculates the distance by comparing phase shift of launch beam and receiving beam[19].

2.3 Passive sensing equipment

Opposite to active sensing equipment, passive sensing equipment does not send energy. It only receives specific energies, like radiation. CCD (Charge Coupled Device) camera is the most frequently used DAE in passive sensing equipment. CCD camera converts video signal to analog signal, then signal line sends the signal to image processing card. The card converts analog signal to digital signal and store in the internal memory. At the same time, the image processing card sends analog signal to the monitor. CCD camera collects the information of object's surface, like color and radiation intensity. But for some 3D information, like deepness, surface reflection characteristic and surface direction, are hidden behind the data. Specific computer vision technologies are required to extract the information[22].

Active sensing equipment can acquire high-precision 3D data relatively easily, but the sensor needs to send energy, like laser. It leads to safety issues. Especially when the measuring object is human body, active sensing equipment is not safe enough. Secondly, those equipments are quite expensive because of energy emission and the special operating environment. Besides, active sensing equipment cannot get curve surface reflectance and other visual information for animation photorealistic rendering. On the other hand, passive sensing equipment, like camera, cannot provide 3D information to the object directly. The data should be extracted from the photos by using Computer Vision technique. However, the camera can provide visual information of object surface to make photorealistic rendering come true. Moreover, passive sensing equipment is cheaper and has less safety issues than active sensing equipment.

Figure 2. 6 Bosch Laser Rangfinder outdoor measurements[23]

Laser Scanner (also named as Laser Rangfinder) which is based on triangulation method is widely used in reverse engineering (Fig 2.6). The data from laser scanner of measurement entity is the distance value to planar rectangular grid vertex value (Fig 2.7). The data is called Range Image. The distance value on the surface of entity being measured can be converted to coordinate (x, y, z) in 3D system. So the distance image becomes a point set with neighbouring relation. Laser Scanner needs to launch laser to the surface and receive reflected light, so it can only measure the points which are not blocked. Therefore, in order to measure an object, a laser scanner has to measure many times from different perspectives to get several distance images. Then register all these distance images to a global coordinate system. After all the procedures, the whole surface measure data of [24].

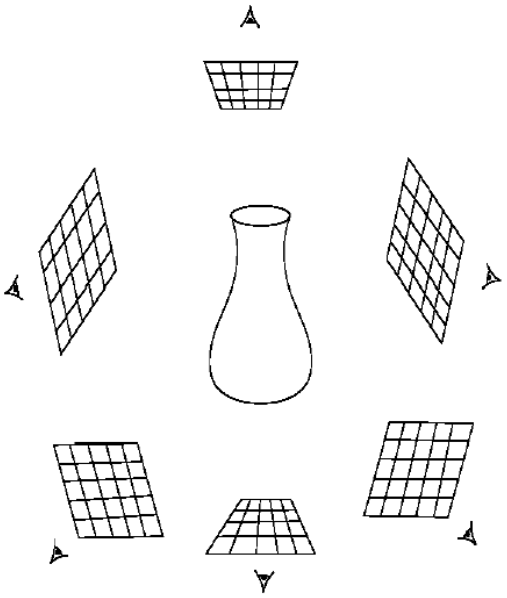


Figure 2. 7 Different positions measuring object

The equipment which is on the basis of triangulation method is called laser triangulation displacement sensor. When it gets installed on the mechanical device, the device becomes a laser scanner. Nowadays, oblique structure triangulation displacement sensor has LD series (Japan KEYENCE ltd.) while direct light structure triangulation displacement sensor has OP2 (RENISHAW ltd.), LC and LB series (KEYENCE ltd.) and other models[13].

Manufacturer	Model	Working distance (mm)	Measurement distance(mm)	Resolution (μm)	Linear (μm)
MEDAR	2101	25	±2.5	2	15
KEYENCE	LC-2220	30	±3	0.2	3
KEYENCE	LB72	40	±10	2	±1%
RENISHAW	OP2	20	±2	1	10
PANASONIC	3ALA75	75	±25	50	±1%

Table 1 technical index for laser triangulation displacement sensor[13, 25]

2.4 General processing

2.4.1 Point cloud

Point cloud, also called as unorganized data, is a set of 3D points and no connections among them. Those points are defined by X, Y, Z and normally represent a streamlined surface and there would be another different value of streamlined surface to approximate them [26].

The main purpose of point cloud reconstruction is searching topology connection among all points to approximate the original surface by triangulation mesh [27]. Point cloud data acquires from 3D scanner. 3D scanner automatically measures points over the subject surface. In the point set, data includes point coordinates and the shape of surface. That is the entity digitalization process and the point set is the point cloud. Point cloud techniques are being used widely but it only performed well after modelling.

The point coordinates are the fundamental data of point cloud, it contains space topology information. The point cloud spatial index is the basic operating and also is the key factor to increase reconstruction efficiency.

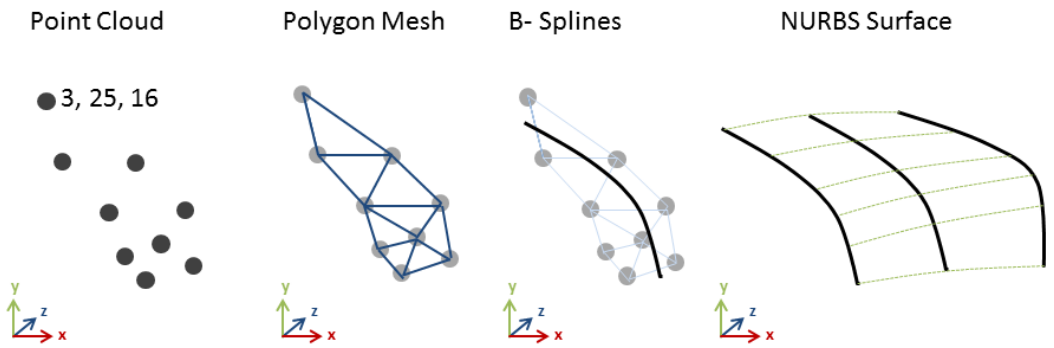


Figure 2. 8 Main reconstruction efforts [28]

2.4.2 Mesh model

Mesh model is defined as unit combinations of discrete problem domain. It is made up by a series of points, edges, surfaces and voxel. Mesh generation refers to the process of polygonal or polyhedral mesh approaching problem domain[29]. Based on the dimensionality of grids, it could be divided into 2D grids and 3D grids. It is a developed theory of 2D grid subdivision and being supported by complete mathematics theory. 3D grids is a topology structure to describe space topological information which is more complicated with uncertain subdivision structures. From the view of constituting mesh topological structure, it could be divided into structured mesh and unstructured mesh. The units in structured mesh are same edge numbered polygons. The subdivision of structured mesh is under conditions. The mesh topology of it is simple and under certain rules. Unstructured mesh was normally showed as triangulation unit which is the triangle mesh. Triangle mesh could show the complicated surface of geometrical feature, and its data scale and data complexity do not much associate with the complexity of geometrical feature. Triangle mesh could be used to describe the surface of complicated model with many detailed features.

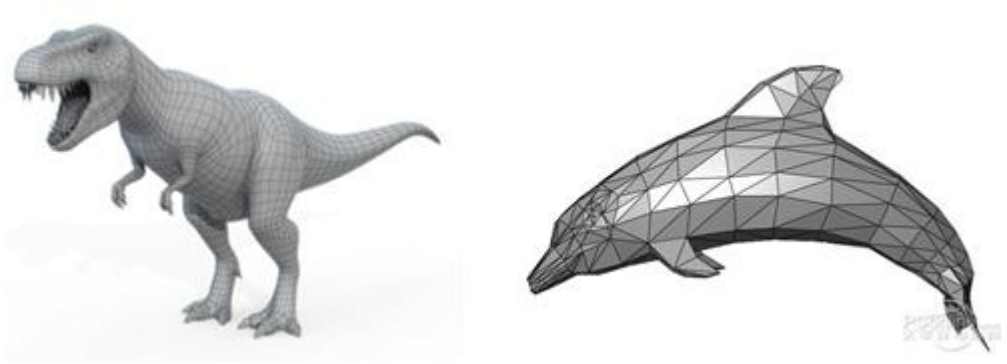


Figure 2. 9 Quadrilateral mesh (left) [30]; Triangle mesh(right) [31]

2.4.3 Reconstruction mesh from point cloud

The problem of reconstructing triangle mesh model from point cloud could be described as: given a set of points, searching a closet approach of original surface by all possible topology connections among points. The process of mesh model reconstruction normally covers topological reconstruction and mesh optimization [32]. For the first step, when reconstructing topology, the topological connection among points has to be found. The reconstruction result will be a polygon mesh and normally was triangle mesh. Triangle mesh optimization means replacing existing vertexes and triangle mesh by a model of less quantity of vertexes and triangle mesh but still keeps the former mesh model's geometric precision and structural information. After the optimization, there would be a smaller amount of mesh with better quality. In this research, triangle optimization did not be taken as an individual chapter but after draft algorithm, it had been a key considering factor during topology reconstruction.

2.5 Existing theory

2.5.1 Algorithm Based on Delaunay triangulation method

This method was the extension of 2D triangulation method in high dimensions by Delaunay triangulation method or Voronoi diagrams to construct point cloud topology data. In the computational geometry, the Delaunay triangulation of a planar point set should meet the requirement that all points could not locate in any triangle circumcircle area after the grid was subdivided. Delaunay triangulation set the principle of maximization the minimum internal angle. It avoided the product of narrow triangles. The triangulation algorithm was proposed by Russian mathematician Delaunay in 1934. Then Silbson [33] proved that Delaunay triangulation algorithm achieved the best quality as a whole. Therefore, Delaunay triangulation algorithm became the most popular research subject in the fields of math, geography and engineering. They also came up various kinds of implementation algorithms. Bowyer [34] and Watson [35] used Delaunay triangulation self-dual diagram, the Voronoi diagrams to accomplish triangulation algorithm of planar point set to high dimensional and non-Euclidean space in theory.

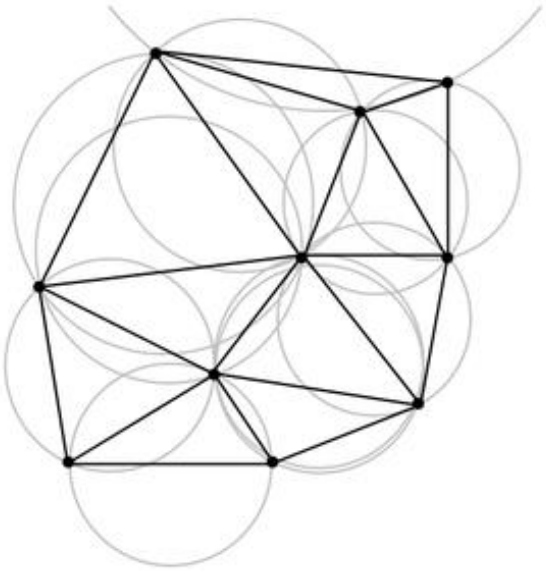


Figure 2. 10 Delaunay triangulation [36]

The original Delaunay triangulation method was for planar point set. But 3D unorganized point cloud was far more complicated than planar point set and they cannot use Delaunay triangulation directly. Thus, many scholars tried to improve Delaunay triangulation from different aspects in order to make it available for spatial data.

- Dimension reduction. This method reduced 3D data to 2D data. Then it could be used the mature technology of Delaunay triangulation to get topology structure. After that, it would be restored from low-dimension to high-dimension based on mapping. Normally planar parameterization was used to reduce the dimensions. The point cloud data would be projected to planar, but three-to-two dimensions cannot be a one-to-one mapping, partitioning projection was required as well as dividing into blocks. Xiong' s method [37] was to project point cloud data to a parametric domain on its own surface and then used Delaunay triangulation method on that parametric domain. Finally to project parameter domain mesh to space surface by its inverse operation of reducing dimension.

- $\alpha - shape$ class method. This method developed from planar Delaunay triangulation. Increased the circle in Delaunay triangulation to a ball in the space and then searched for subdivision points in a ball domain. $\alpha - shape$ method came up by Edelsbrunner and Mucke [38] in 1994. Firstly, given a point set p and a real number referenceα and $\alpha - shape$ of p was an independent polyhedron. Different values of α created different $\alpha - shape$ set. Delaunay method was being used in every independent $\alpha - shape$. The difficulty of it was to find a suitable value of α to keep essential triangulated elements and also deleted unnecessary faces and edges. Moreover, this method did not perform well on point cloud data of uneven or discrete surface.

- Crust class method was also named as Voronoi filtering algorithm. It was raised by Amenta and Bern [39]. This method relied on the concept of pole doublet and axis. Pole doublet was the subset of vertex in Voronoi diagram. In each Voronoi diagram, vertex p_v had at least two poles p_1 andp_2. They were on the different sides of p_v. Angle between vectors of those two poles and p_v were obtuse. Central axis was a basis line that bisected its line segments. Boissonnat. Cazals [40] and Amenta [41] proved that under strict conditions, a dense sampled point cloud and pole set could have an excellent approach to central axis. Vector between vertex and pole fit original surface normal vector. Power crust [42] which was similar to Crust method but based on that, it added weighted graph concept. This method can be used to deal with noise and shape feature surface.

Based on Delaunay triangulation reconstruction algorithm had been proved feasible and accurate in theory. But in real use, the result was not ideal because of complicated computing, especially for those point cloud with noise and feature detailed objects.

2.5.2 Reconstruction method based on implicit surface

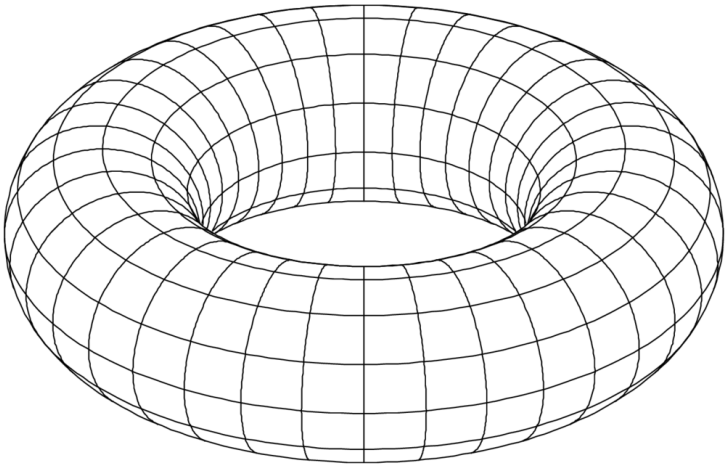


Figure 2. 11 Implicit surface torus (R=40, a=15) [43]

Reconstruction based on implicit surface is a curve fitting method in essence, it uses a set of implicit equations of curve fitting the original point cloud, then generate triangular mesh on zero value level. The implicit functions include the radial function, the linear equation groups or the partial differential equations. This step is mainly based on Marching Cube and Bloomenthal polygonization method. Given a set of distributed point cloud on the surface X = $\{x_1\} \in R^3$, the purpose of implicit surface reconstruction is to find a function y = *f (x)*, the zero level set of *f (x)* approach the original surface of point cloud X, the unique

difference of this method is that the function curve and the distance approximation measures in different ways. Radical basis functions (RBF) and moving least square method are widely used in the construction of implicit surfaces. [27]

Ohtake [44] et al use a compact support radial function interpolation or approximation of the original surface of the point cloud data. Begins with the assumption that the point cloud data scattered on a surface and built point cloud hierarchical Oc-tree structure, and then from coarse to fine, build every layer a compactly supported radial basis function interpolation or approximation of the original surface.

Hoppe's [45] method is divided into two steps, the first step is to define a function $f : D \rightarrow IR$, where D is an approximation point set of point cloud. *IR* is a surface approximation to the manifold surface of point cloud data. *f* is a symbol of the distance measurement error, Z (*f*) is the zero level set of a best approximation of the original surface. The second step of this method is to approximate the zero level set with a contour algorithm.

Lin [46] et al proposed a polar field model of implicit surface reconstruction method, the method is called "field fitting method", the surface reconstruction problem is modelled for the sake of particle distribution, the surface is expressed as discrete pole zero surface. The discrete polar field has produced particle, and the particle distribution is solved by greedy algorithm.

A method based on implicit surface fitting can be used to reconstruct a smooth, continuous and deformable mesh, which is suitable for smooth objects. But it is difficult to find a suitable implicit equation to describe some complex surfaces,

and it is more complex to draw. Periodic surface cannot generate high quality implicit surface by the mesh reconstruction algorithm based on implicit surface. Three dimensional periodic surfaces is the parameter space of the corresponding two dimensional domains, and the two dimensional parameter spaces are not closed. For such periodic parameters surface, the 3D reconstruction outcome is not ideal.

2.5.3 Boundary extension method

Boundary extension method is also known as the advancing front method (AFT), the basic idea is: first creates a seed triangle meeting initial conditions, then use the edges of the seed triangle as a forward border, gradually adding point cloud point to form a new triangle, loop the procedures and extend to cover all points. This method is derived from the greedy algorithm, that is, every time to find a best candidate point from the rest of the grids, so that every time the point joins the grid makes the best topology, the whole grid is optimal when it ends. This approach appears to be not as complete as Delaunay's one which has mature theoretical support. In fact, every time the best point of the judgment is integrated into the theory of computational geometry. Therefore, the key of this kind of algorithm is the determination of initial triangle and the edges of the new formed triangles. Many scholars put judgment criteria for this method.

Boissonnat's [47] is the method of choosing the free point which could form a new triangle with forward border and provides the biggest field angle.

Bemardini [48] rolling ball method, also known as BPA algorithm (Ball Pivoting Algorithm) which is based on the theory of $\alpha-$ shape, construct grids by using

the radius *r* of the ball rolling in the point cloud data, the basic principle of BAP algorithm is : a specified radius of the spherical surface made by three points constitutes a triangle in the grid, and there does not contain any other points inside the ball. The method starts with a seed triangle, and the ball rolls along the edge of the seed triangle until the ball touches the new point to form a new triangle. If the radius of the ball rolling on all the leading edge cannot find a suitable new triangle, then increases the radius of the triangle, until finding a new point to form a new triangle. Repeat the steps above until all points are added to the grid.

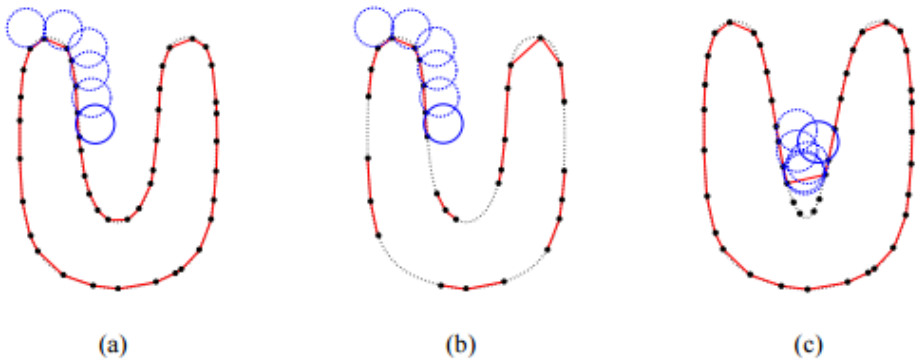


Figure 2. 12 BPA in 2D. (a) A certain radius circle pivots among points, and edges are being connected. (b) In those uneven density area, points in low density area would not be connected as edges. (c) The feature will miss when the circle pivots to narrow or sharp area.[48]

Lin [49] method is to start from the seed triangle, according to the point cloud data own property to choose the new point to join in the grid, this property is called the UDS (Uniformity Degree Sampling). UDS of a point represents for the density of point cloud close to this point, it is the intrinsic property. The greater the value is, the less uniform point density close to that point. UDS is the value of $L_{max}/L_{min}$of the edges connecting to this point. The grid is a minimum weight triangle approximation of the two-dimensional manifold of the point cloud data.

Thus, the topological mesh is an optimal approximation of the original sample surface.

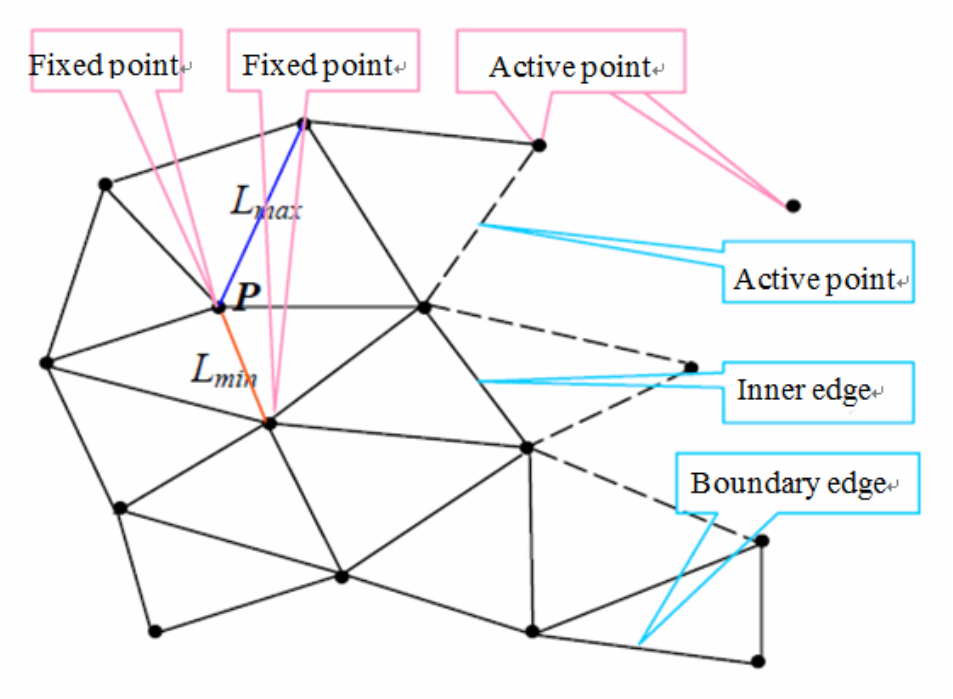


Figure 2. 13 The Uniformity Degree Sampling of P is L_{max}/L_{min} [25]

This method can compute large-scale point cloud data, and get closed or non-closed surfaces directly out of the point cloud, the basic idea of this algorithm is simple, but the programming requires a lot of skills. The implementation of the algorithm involves the storage of data, the method of determining the seed triangle, the searching method of the frontier, the intersection test between the frontiers, which directly affect the operating efficiency of the Boundary extension method.

2.5.4 Based on learning method

Based on learning method combines statistical learning and machine learning to mesh reconstruction.

Kohonen [50] et al found that in a simple self-organizing neural network, which accepts the basic event as a signal input, it is automatically mapped into the output of the information received by the neural network. In this mapping mode, the output of the neural network and the basic event are the topological isomorphism. This principle makes the topology reconstruction simpler.

Ivrissimtzis [51] uses the growing cell neural network to reconstruct the topological structure of point cloud data. They believe that all of the neural networks that used for point cloud reconstruction are a neural network that can be modified to be called a growth element structure. The algorithm adjusts the point cloud data to the neural network structure, which consists of neural network connections and neurons.

Jenke [52] thinks the point cloud data and the original surface are modeled as a probability distribution problem. The Bayesian rule could be used to estimate the probability of the reconstructed surface and original surface superposition. The key of this method is to define the measurement data and the reconstructed surface as a point cloud, which is used to describe the primitives in the finite dimensional space in the way of statistical hypothesis. This make surface reconstruction problem been discretized into a numerical optimization problem. Finally, the problem is transformed into a triangulation.

Neural networks can be with any precision approximation of any continuous function and its derivative, the characteristic of neural network can be used in 3D reconstruction of point cloud data, by using neural network the network

connection preservation of full value and threshold surface mapping relationship to improve the model of fault tolerance and association ability.

The disadvantages of this method are that it is difficult to converge on the network, the calculation cost is high, and just can be used for the topological reconstruction of simple surfaces. It is unable to divide the mesh of arbitrary topology and arbitrary complex shape.

2.6 Mainly existing problems of methods above

The mainly considering aspects of 3D point cloud reconstruction are: running speed, precision, simulating capability, the memory usage and robustness. After analysing the method, the key problems are listed below:

2.6.1 Theory based on Delaunay triangulation method

The weakness of this method was it took too much computing time and to those point cloud with noise, the consequence of reconstruction was not very well, and the robustness was not good enough.

2.6.2 Theory based on implicit surface

It was difficult to find a suitable fitting function that caused the low precision of algorithm.

2.6.3 Theory based on boundary extension algorithm

The key factor of it was to find a suitable data structure for growth judgments and managing point cloud.

2.6.4 Theory based on learning reconstruction method

The being most commonly was by neural network. This method needed to solve a great deal of equation set that lead an inefficient computing capacity. Meanwhile, the robustness was not good enough that could not resist noise.

Chapter 3:

Research Methods

Chapter 3:

Research Methods

3.1 Overview

The purpose of this research was to find out an algorithm running in Solidworks that suits models with or without hole. The density uniformity and point cloud scale affect the results directly. There is no doubt that the more coordinates involved in programming the more precise the surface would be, meanwhile it accounts for more running time and storage. In this research, the following steps had been done to rebuild the surface of object and balance the precise-efficient problem.

- Open file in Solidworks
- Point cloud pre-processing
- Seed triangle and candidate point preparation
- Boundary extension with topological reconstruction

3.2 Import .txt file in Solidworks

All research results were based on Lenovo T440 64-bit operating system and Solidworks2013 64-edition.

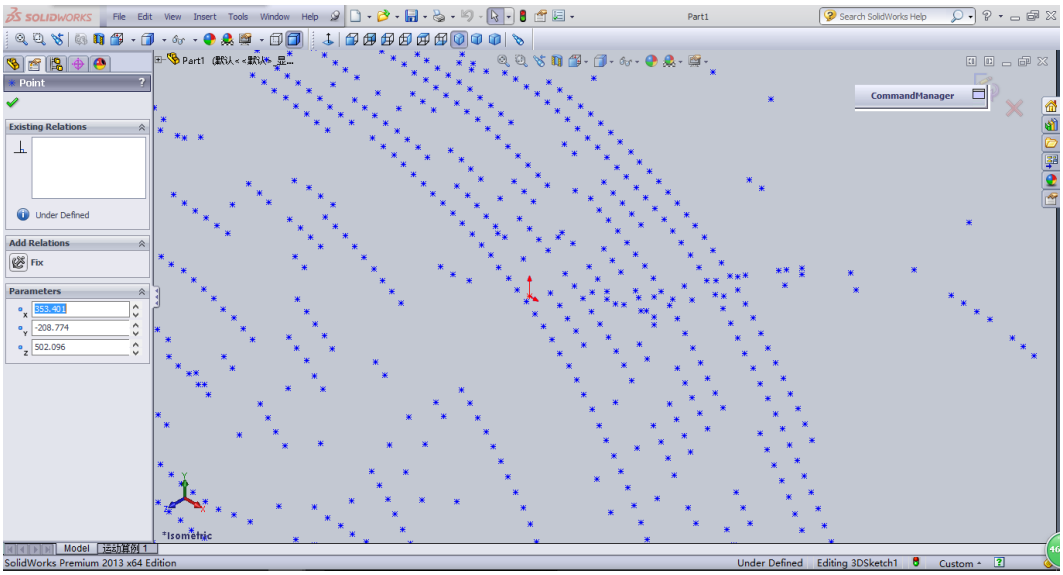


Figure 3. 1 “Bunny” coordinates were being imported into SolidWorks

3.3 Data pre-processing

3.3.1 Unified point cloud’s order of magnitude

As shown in the pictures below, “Container” [11] and “Bunny” [53] coordinates are in different order of magnitude. Coefficient (1000 for “Container” and 10 for “Bunny”) had been set to make sure different point cloud models running under same order of magnitude. The coefficient was mainly up to the decimal scale of coordinates. It does not show the difference after .txt file being read in Solidworks because of the proportional scale function, but after that, the point cloud scale was going to be reduced and being divided into smaller grids. A coefficient could unify different orders of magnitude to same magnitude. In this research, coordinates’ dimensions are unified in millimeters (mm).

X	Y	Z
-100.000	-20.000	100.000
-100.000	-280.000	100.000
-100.000	-46.000	100.000
-100.000	-72.000	100.000
-100.000	-98.000	100.000
-100.000	-124.000	100.000
-100.000	-150.000	100.000
-100.000	-176.000	100.000
-100.000	-202.000	100.000
-100.000	-228.000	100.000
-100.000	-254.000	100.000
100.000	-280.000	100.000
75.000	-280.000	100.000
50.000	-280.000	100.000
25.000	-280.000	100.000
0.000	-280.000	100.000
-25.000	-280.000	100.000
-50.000	-280.000	100.000

X	Y	Z
-2.09893	1.77858	0.601162
-2.7939	1.87328	0.344192
-5.11691	4.10898	3.87322
1.4553	1.99958	2.4757
-0.5765	1.65208	0.869282
-0.82674	1.57668	0.777921
-2.02805	1.72948	0.333255
5.00534	0.253779	2.92231
5.48829	-0.0399202	1.77059
-0.86679	0.241379	3.82137
-0.76902	0.24818	3.88839
4.42435	1.14058	2.27577
-4.60557	4.82648	-1.60501
5.69217	-0.59522	2.37054
6.19936	-1.69574	1.26973
-1.56561	6.40768	-0.085304
-6.36183	2.56728	5.15689
-5.5654	1.58678	5.44389

Figure 3. 2 "Container" coordinates (left) [11] and "Bunny" coordinates (right) [53]

3.3.2 Determine boundary value

After looping through all the coordinates in .txt file, set an order of the coordinates from small to large and save them in an array *OriginalPointsXYZ()*.

- If $X_1 < X_2$ Then $\begin{cases} P_1 = (X_1, Y_1, Z_1) \\ P_2 = (X_2, Y_2, Z_2) \end{cases}$

- If $X_1 = X_2$

Judge $Y_1 < Y_2$ $\xrightarrow{No}$ $Y_1 = Y_2$ $\xrightarrow{No}$ $\begin{cases} P_1 = (X_2, Y_2, Z_2) \\ P_2 = (X_1, Y_1, Z_1) \end{cases}$

Yes ↓ Yes ↓

$\begin{cases} P_1 = (X_1, Y_1, Z_1) \\ P_2 = (X_2, Y_2, Z_2) \end{cases}$ Judge $Z_1 < Z_2$ $\xrightarrow{No}$ $\begin{cases} P_1 = (X_2, Y_2, Z_2) \\ P_2 = (X_1, Y_1, Z_1) \end{cases}$

Yes ↓

$\begin{cases} P_1 = (X_1, Y_1, Z_1) \\ P_2 = (X_2, Y_2, Z_2) \end{cases}$

- If $X_1 > X_2$ Then $\begin{cases} P_1 = (X_2, Y_2, Z_2) \\ P_2 = (X_1, Y_1, Z_1) \end{cases}$

When this step had been done, the maximum and minimum value of coordinates could be found and all coordinates had been save in array *OriginalPointsXYZ()*. This step was to determine the size of boundary cube for the following step to reduce the scale of point cloud. As shown in screen capture (Fig 3.3), the coordinates could also been read from the parameters section, in this "Container" model, the boundary coordinates were quite obvious.

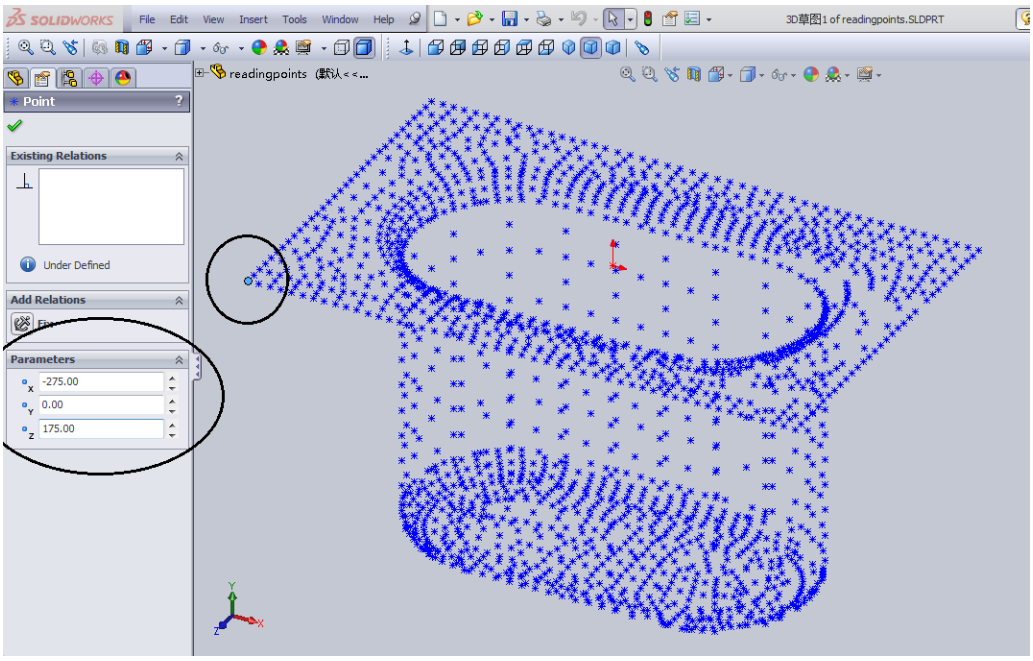
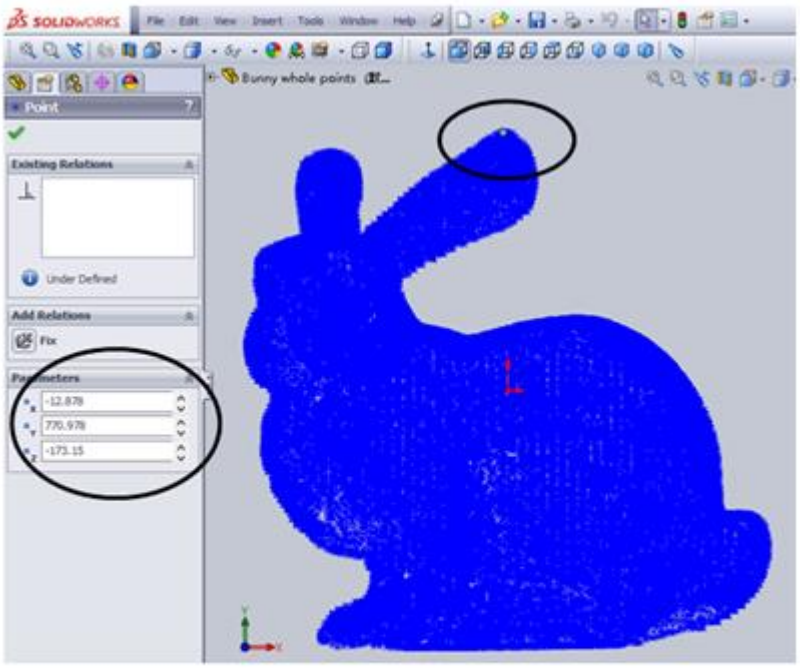
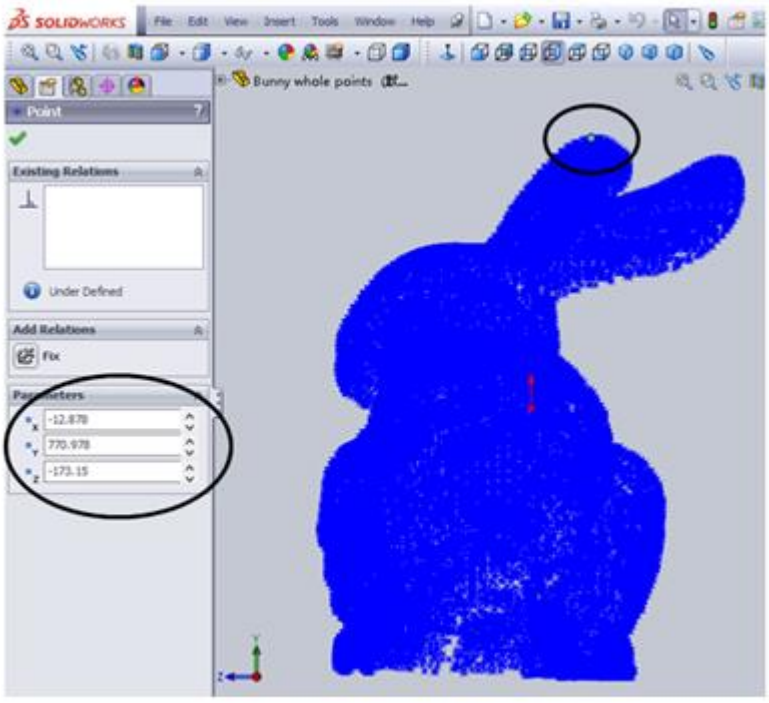


Figure 3. 3 "Container" point could in SolidWorks

But in the following models, "Bunny" [53] and "Human head" [54], the boundary coordinates would not be accurate by reading from parameters section.

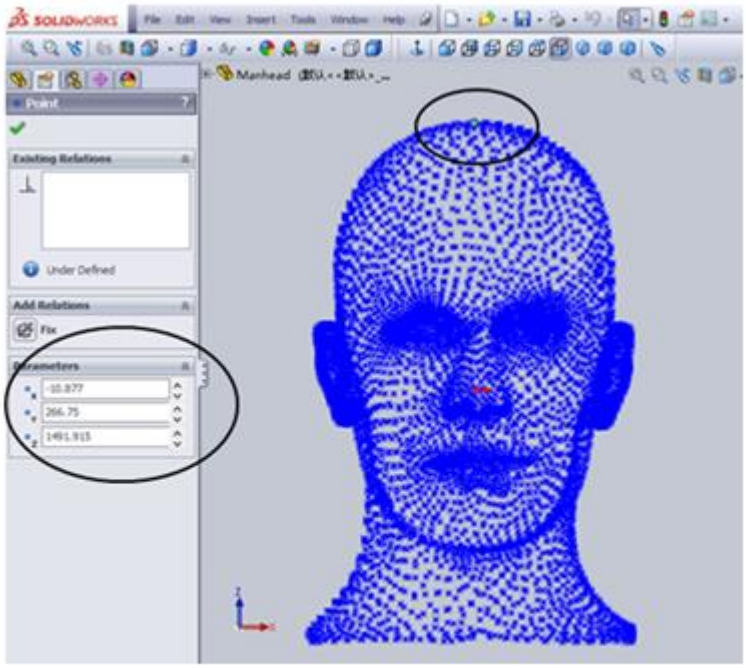


a

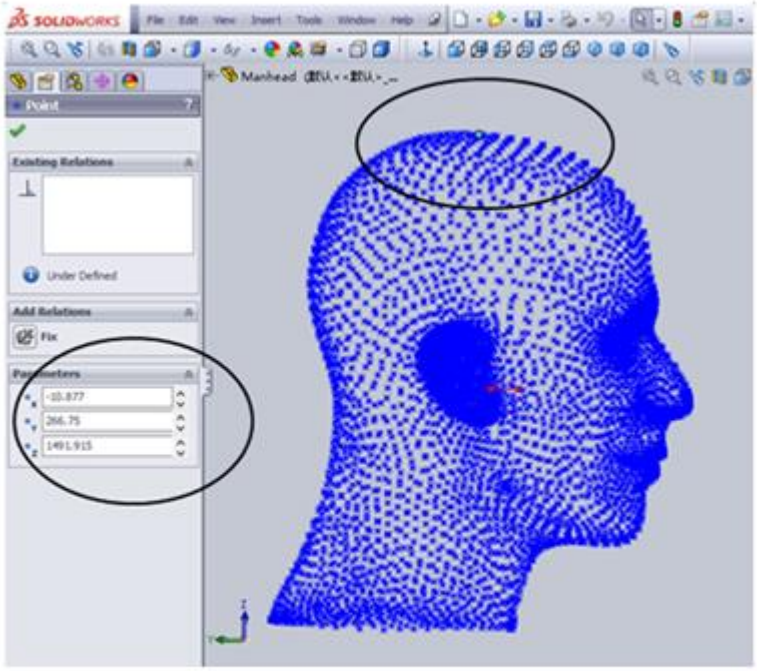


b

Figure 3. 4 a- front side of "Bunny" point cloud,
b- right side of "Bunny" point cloud



a



b

Figure 3. 5 a- front side of "Human head",

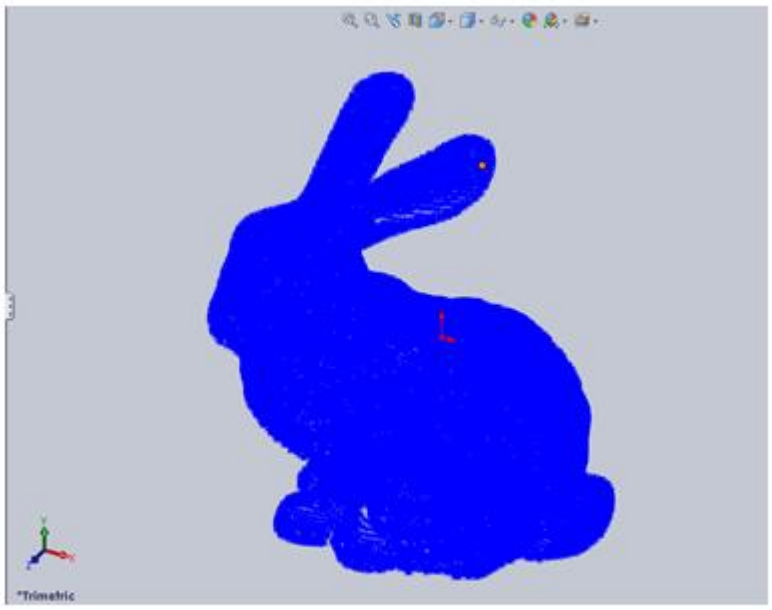
b- left side of "Human head"

As shown above, although all coordinate values could be found in parameters section, it was still misleading if the selected point was the maximum or the minimum valued one. The main function of the parameter section in this research was to be a reference. After found maximum and minimum value of point cloud achieved by programming, the accuracy of setting boundary cube could be improved.

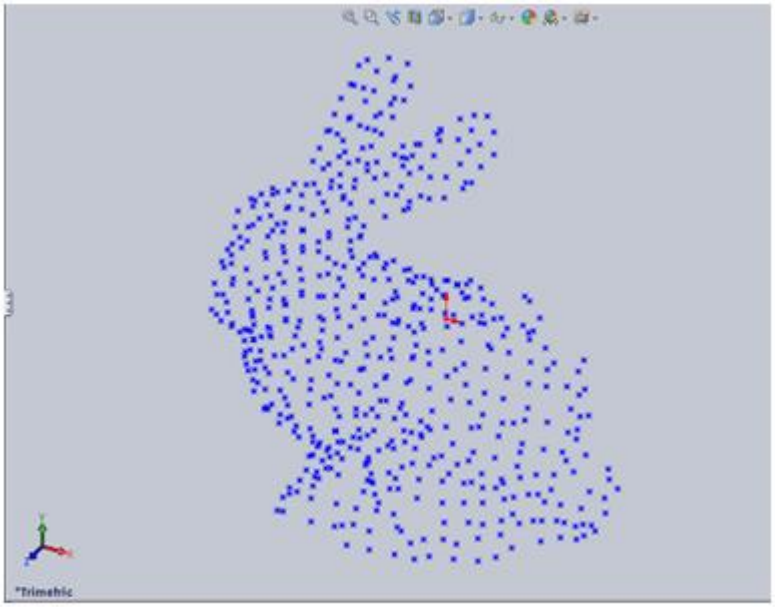
3.3.3 Point cloud data simplification

The main purpose of this step was to set a boundary cube to cover all coordinates and remove some of them. In order to cover them all, the minimum boundary cube should be slightly larger than original model contour or exactly cover the boundary coordinates. Another coefficient (reductionCofficient) had been introduced into programming in this step. Due to the different sizes and density of point cloud, the coefficient could only be set manually in this research instead of being calculated by equations.

As the maximum and minimum value had already known, the minimum boundary cube could be calculated. The algorithm started from the minimum value of coordinates and all coordinates would be divided into a smaller cube. The point which was the closet to the center of block would be saved in a new list meanwhile the other points in this block would be removed.



a



b

Figure 3. 6 “Bunny” with whole points (a), “Bunny” after reduction (b)

- Boundary cube and reduction cubes

The boundary cube size was up to the maximum and minimum value of coordinates. In the Bunny point cloud, the length between bunny head and tail

was 15842.85mm, and depth between bunny ear and feet was 15960.50mm, the widest part accounted for 12783.54mm. It would not affect the result if the boundary cube been taken slightly larger for more clear and easier to calculate. Therefore, boundary cube for “Bunny” was taken as 16000mm*16000mm*16000mm and it had been divided into 10mm*10mm*10mm, the reduction cubes, along with each axis and each axis held 16 cubes.

The “reductionCoefficient” could be changed manually in this programming to meet the needs of different density and precision of point cloud and multiple uses in industrial.

- Coordinates simplification

In order to increase the working efficiency, not all of the points are essential to surface reconstruction. Meanwhile, the more points there are, the easier programming error occurs. The main purpose of simplification was to reduce the number of points and density uniformed. The uniform density was really important in this research, as shown in Fig 3.7, besides the incomplete testing algorithm itself, it worked well in both high density area (top of the “container”) and low density area (“container” cylinder), but the errors in connected domain was obvious.

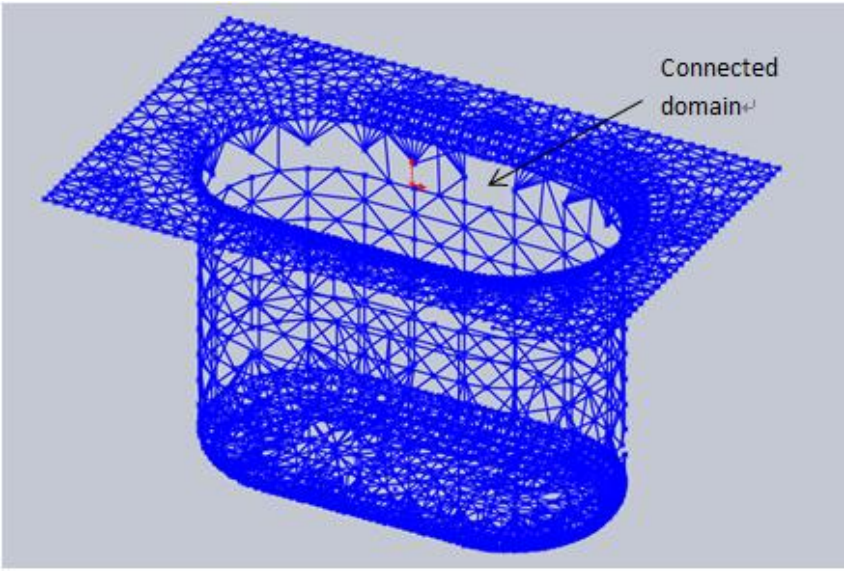


Figure 3. 7 "Container" testing result 1

The method of this step was based on the reduction cubes which had been set by last step. To simplify the model from 3D to 2D, a draft was showing in Fig 3.8, dashed line represented for reduction cubes which zoned all points into different areas. The only points been reserved was the one closest to the center of dashed square, the others in same dashed square had been removed. All those reserved points would be saved in a new array *PointsXYZ()*. And all following algorithm would run with *PointsXYZ()*.

The first dashed square started from minimum value of X and Y, and length of dashed square had already known (10mm for "Bunny"). The center point of first dashed square was easy to find, *1* (x+5, y+5). The following center points were *2* (x+5, y+10), *3* (x+5, y+15), *4* (x+10, y+5) .etc. The closest point searching method was:

If there was no point in this area, output null.

If there were points in this area, searching the one had longest distance to center point and then compared the distances between this point to center point and other points in same dashed square to center point. If the distance was shorter than the

longest distance point, removed the point which had the longer distance. Kept comparing until other longer distance points than the existing one had been removed, then the existing one point was the closet point to center. Save all the points in *PointsXYZ()* [55] and read it in Solidworks.

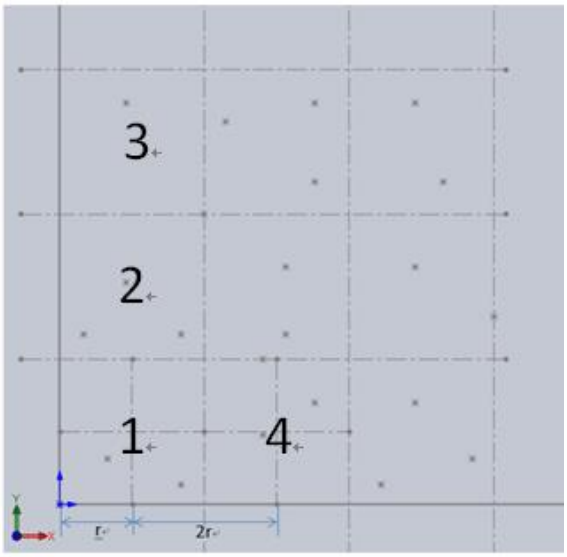


Figure 3. 8 Coordinates in 2D plane

3.4 Seed triangle and candidate point preparation

Theoretically, seed triangle could be any minimized triangle which did not include any other points. In this research, vertexes of seed triangle were set on the flat area.

After setting seed triangle, all points in *PointXYZ()* would be divided into different grids. That was because when the seed triangle started to search other points to form a new triangle, only the points in certain area needed to be judged whether it was the best candidate point. When the grids being set, seed triangle only need to search the points in neighbor grids, otherwise it had to search whole

point cloud until the end of *PointXYZ()* [56]. Hu, Ma and Wu (2008) provided an improved way to divide point cloud into constraint space [55, 57, 58] .

$$\begin{cases} m = [1 + 3.32\lg(X_{max} - x_{min})] + 1 \\ n = [1 + 3.32\lg(Y_{max} - y_{min})] + 1 \\ p = [1 + 3.32\lg(Z_{max} - z_{min})] + 1 \end{cases} \quad (3\text{-}1)$$

Where *m*- number of grids along with X axis

n- number of grids along with Y axis

p- number of grids along with Z axis

After the calculation, m_b, n_b, p_b for "Bunny" and m_c, n_c, p_c "Container" respectively were:

$$\begin{cases} m_b = 16 \\ n_b = 16 \\ p_b = 16 \end{cases} \qquad \begin{cases} m_c = 10 \\ n_c = 10 \\ p_c = 10 \end{cases}$$

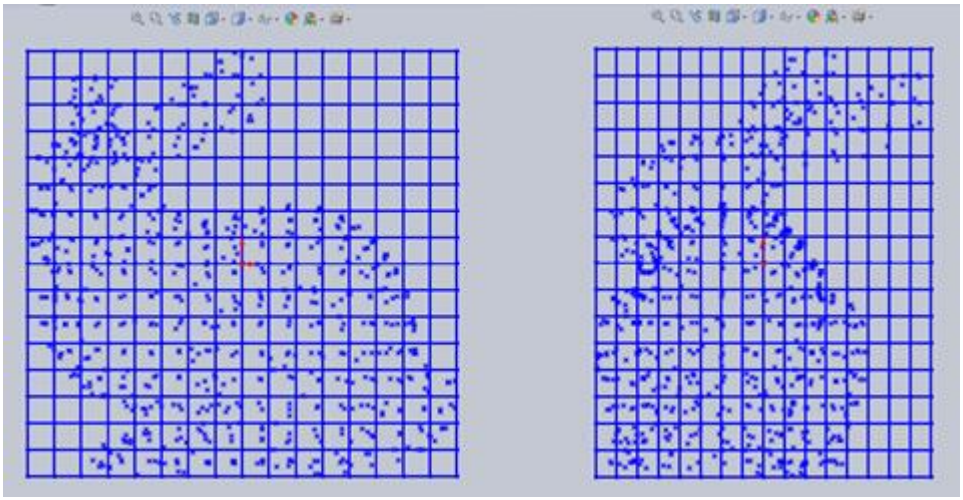


Figure 3. 9 "Bunny" in grids front side (left) and right side (right)

The gird length along axis is:

$$\begin{cases} l_x = \dfrac{X_{max} - x_{min}}{m} \\ l_y = \dfrac{Y_{max} - y_{min}}{n} \\ l_z = \dfrac{Z_{max} - z_{min}}{p} \end{cases} \tag{3-2}$$

For P (P_x, P_y, P_z) in point cloud [59]:

$$\begin{cases} i_x = \left\lceil \dfrac{P_x - x_{min}}{l_x} \right\rceil \\ j_y = \left\lceil \dfrac{P_y - y_{min}}{l_y} \right\rceil \\ k_z = \left\lceil \dfrac{P_z - z_{min}}{l_z} \right\rceil \end{cases} \tag{3-3}$$

Set $\omega_{i,\ j,\ k}$ as grid address, $i \in [1, m]$, $j \in [1, n]$, $k \in [1, p]$. Coordinates in *PointXYZ()* would be re-organized as:

Grid ID	$\omega_{1,1,1}$	$\omega_{1,1,2}$	...	$\omega_{1,1,k}$	$\omega_{1,2,k}$	...	$\omega_{1,j,k}$	$\omega_{2,j,k}$	...	$\omega_{i,j,k}$
Point ID	P_1	P_5	…	…	…		…	…	…	P_n
	P_2	P_6		…			…			
	P_3	P_7					…			
	P_4									

Table 2 Schematic table of point re-organization

3.5 Boundary extension with topological reconstruction

The general idea of boundary extension was to search the closet points which had to meet all requirements to seed triangle edges or to *ActiveEdge*. And there were also some term definitions to be mentioned:

- *ActiveEdge*: Lines had been marked as an edge of a triangle.
- *FixedEdge*: Lines had been shared as an edge by two triangles.
- *ActivePoint*: Points in *PointXYZ().*
- *DeadPoint*: A point had been shared by or above eight lines and the points being marked during processing. Once a point got marked as *DeadPoint*, it would not be involved in the following operation.
- *BanList*: List for a set of points and certain edges that could not form a triangle. Points in *BanList* could still be connected to other edges besides the certain marked edge.
- *GirdID*: Location of points being divided into small cubes.
- *GridLineID*: The middle point of lines had been taken as a main feature in grids and saved in *GridLineID*.
- *MarkLine*: A record for points and lines.
- *minValue*: The shortest distance from candidate point to target edge.

Before searching new points, the only data were three *ActiveEdges* (the original seed triangle edges) and all points in its and neighbour grids. That simplified problem to search the closest point to that edge. It would take a lot of work to calculate the distance between each point to target edge. Instead of doing that, the middle point of *ActiveEdge* was most easily to get, then problem turned out to calculate the shortest distance between two points in a certain area. Plus, searching by the middle point of *ActiveEdge* could provide a better candidate point because it would prior take the point with greater internal angle and that

would make a better quality triangulation mesh.

There were two main loops, the first one was going to search for the best candidate point and second one set some restricted conditions to check the generated triangles. The programming stopped when there was no new edges could be connected and all points connected by lines. Before looping, the property of lines would be checked. The last step was to draw triangles.

3.5.1 Loop 1 for all points in *PointXYZ()*

1. Searching the points in the neighbour grids of the middle point of seed triangle edges.

2. Judging if there were already lines connected between the point and vertexes of the searching edge. If there were, judged the property of the lines. If one of the lines was not *FixedEdge*, then the point was not *DeadPoint*.

3. Judging if each internal angle was greater than 20 ° after being set as a new triangle.

4. Judging if the angle between the former triangle and new triangle was greater than 90°.

5. Comparing the distance between candidate points which met all the requirements to the edge. Took the closest one as the vertex of new triangle.

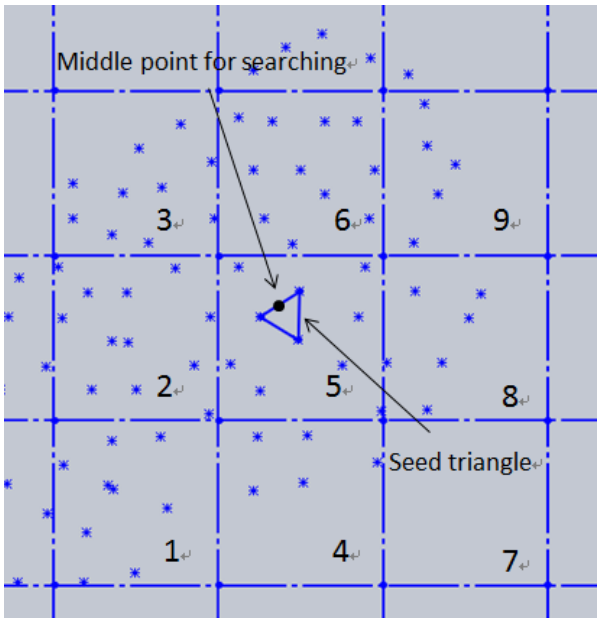
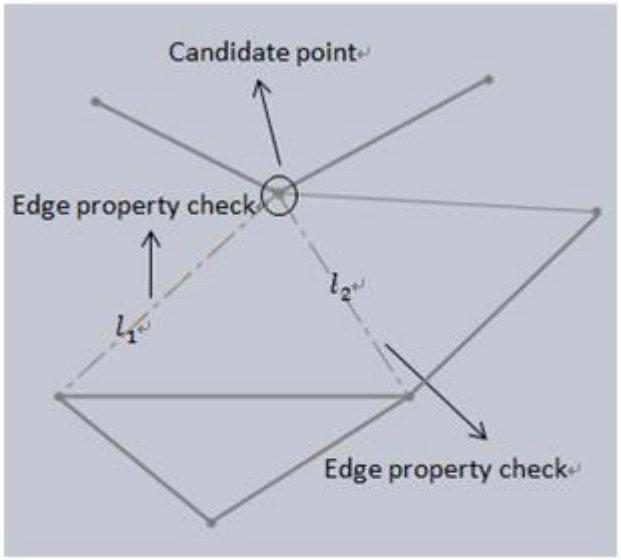
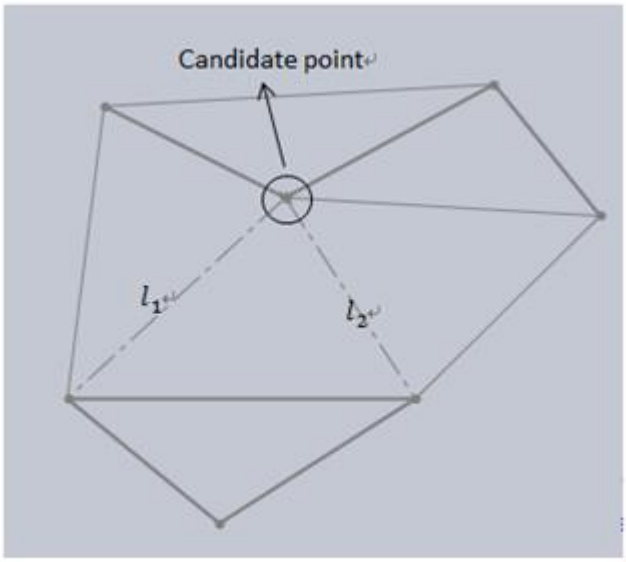


Figure 3. 10 ActiveEdge searching area

For first loop, the searching edge for closest points was the edges of seed triangle. After formed a new triangle, the edge of seed triangle would be shared by two triangles and the new formed edges still belonged to one triangle. Marked the edge shared by two triangles as *FixedEdge* and checked if the two vertexes of *FixedEdge* were *DeadPoint*. Marked the new formed edges as *ActiveEdge*, and kept searching for new point by its middle point. As in Fig 3.10 showed, the searching would start from the point close to the *ActiveEdge*, if the points in its own grid did meet the requirement, it would search the numbered grids. In 3D model, the neighbour grids were the 3*3*3=27, including itself.



a



b

Figure 3. 11 Searching new points (a) and point proper change (b)

After finding a candidate point, judge if there were already being connected by edges. As in Fig 3.11 (a), checked if there were existing l_1 and l_2. If there were, then did not connect them. If there were not, connected them and got l_1 and l_2. Checked all the lines connected to candidate point in Fig 3.11 (b), l_2 had been shared by two triangles then marked it as *FixedEdge*, l_1 was the edge of one

triangle, then marked it as *ActiveEdge*. The candidate point still could be searched by other edges. But in Fig 3.11 (b), all edges connected to candidate point were all shared by two triangles which were the *FixedEdge*, then marked the point as *DeadPoint* which would not be searched by other lines anymore.

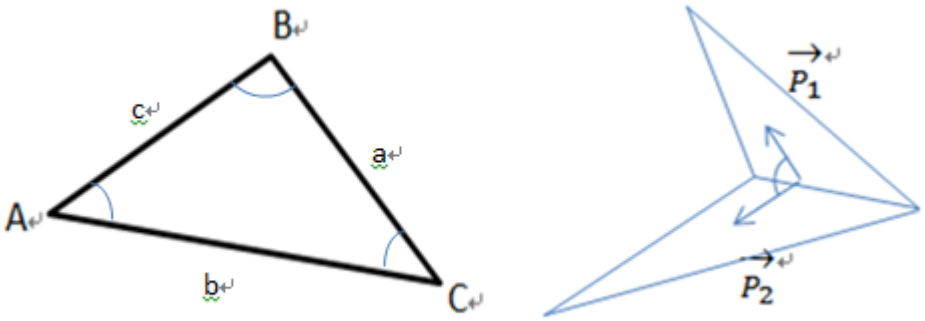


Figure 3. 12 A triangle (left), Angle between perpendiculars (right) schematic

Since all points data had been saved in *PointXYZ()*, coordinates could be called. Each internal angle would be checked in new formed triangle. If any of those angles was smaller than20°, the candidate point would be marked as *DeadPoint*. Another point would be found and calculated.

$$\begin{cases} a = l_{BC} = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2 + (z_B - z_C)^2} \\ b = l_{AC} = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2 + (z_A - z_C)^2} \\ c = l_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} \end{cases} \tag{3-4}$$

$$\cos^{-1} A = \frac{(c^2 + b^2 - a^2)}{2 \times b \times c} \tag{3-5}$$

For checking the angle between neighbour triangles, make a perpendicular from both of the triangle's vertex to the shared *FixedEdge*. The dihedral angle could be calculated as:

$$\sin^{-1}\theta = \frac{|\overrightarrow{p_1} \times \overrightarrow{p_2}|}{|\overrightarrow{p_1}||\overrightarrow{p_2}|} \quad (3\text{-}6)$$

θ would be checked, if it was in the range of (90°,180°). That was to make sure the surface of mesh would not be too sharp. If θ was not in that range. The vertex (candidate point) of new triangle would be marked as *DeadPoint*, the *ActiveEdge* would search for a new point and check repeatedly.

The main reason to set internal angle greater than 20° was to fit original model's curvature as close as possible and provided a smooth surface, meanwhile the set of angle between neighbour triangles greater than 90° was to make sure the out-ward spreading boundary extension and avoided sharp features. [59, 60]

Normally there would be more than one candidate point met all the requirements. The distance between points and *ActiveEdge* would be calculated. The closest one would be chosen as the best extension point and form a new triangle with *AcitveEdge.*

End loop

After searched a new point, a virtual triangle had been created, the three edges were: an *ActiveEdge* and two target edges. Calculated the grid ID of target edges and saved in *GirdLineID.* Saved the distance from the candidate point to *ActiveEdge* in *minValue*

3.5.2 Loop 2 for all points and edges

The main purpose of loop 2 was noise removal and to correct the topological connections.

1. Judging if the point was in the neighbour grids of target edge, if it was the existing vertex of the target edge, if it was the candidate point, if it was outside the former triangle.

2. Judging if the distance of the point to target edge was smaller than $^1/_2$ *minValue.* If they did, marked the point and edge to *BanList.*

3. If the distance of point to target edge was greater than $^1/_2$ *minValue.* Judging if the point was internal of the new triangle, if the distance of point and target edge was smaller than *minValue.*

End loop.

For every point and line, *MarkLine* saved the information about edge property (*ActiveEdge* or *FixedEdge*), edge location (the grid ID of the line middle point) and number of lines the each vertex had connected (no more than eight). Then it could be checked if there were connections before. To check if the point was internal or outside the triangle, just need to check the angle between perpendiculars by equation (3-6). If the angle was greater than 90°, it was outside the triangle, if it was or smaller than 90°, it was internal the triangle and would be marked as *DeadPoint.*

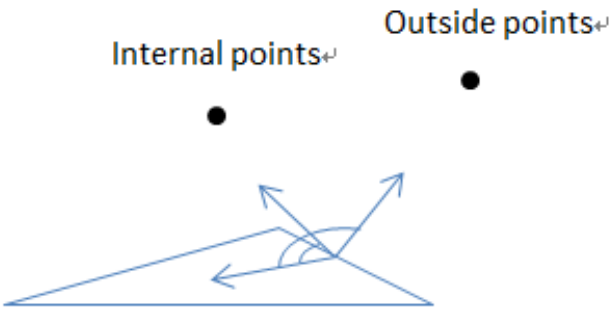


Figure 3. 13 Point location

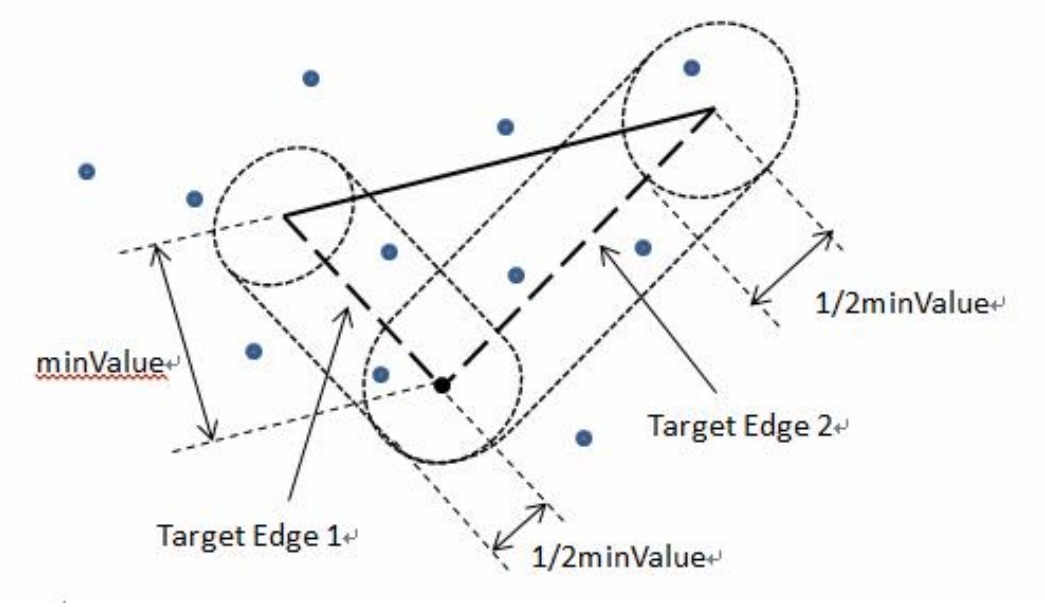


Figure 3. 14 Noise removal area

The reason to set *minValue* as the searching radius was because of the different density and size of point cloud. It would be better to set a self-adaptively variate to adjust it. [61]

For those had been searched by two cylinders, the distances between them to target edge was smaller than $^1/_2$ *minValue*. Since *minValue* was the array saving the shortest candidate point to *ActiveEdge*, those points closer to *ActiveEdge* did

not meet the entire requirements in Loop 1 to form a triangle with the *ActiveEdge*. Then those points and the edge would be saved as a set in *BanList.* If there was no line connected to this point, marked it as *DeadPoint.* For those point in the region ($^1/_2$ *minValue, minValue*) and internal the triangle, if there were lines connected to this point, marked it to *BanList* meanwhile stopped it creating new triangle.

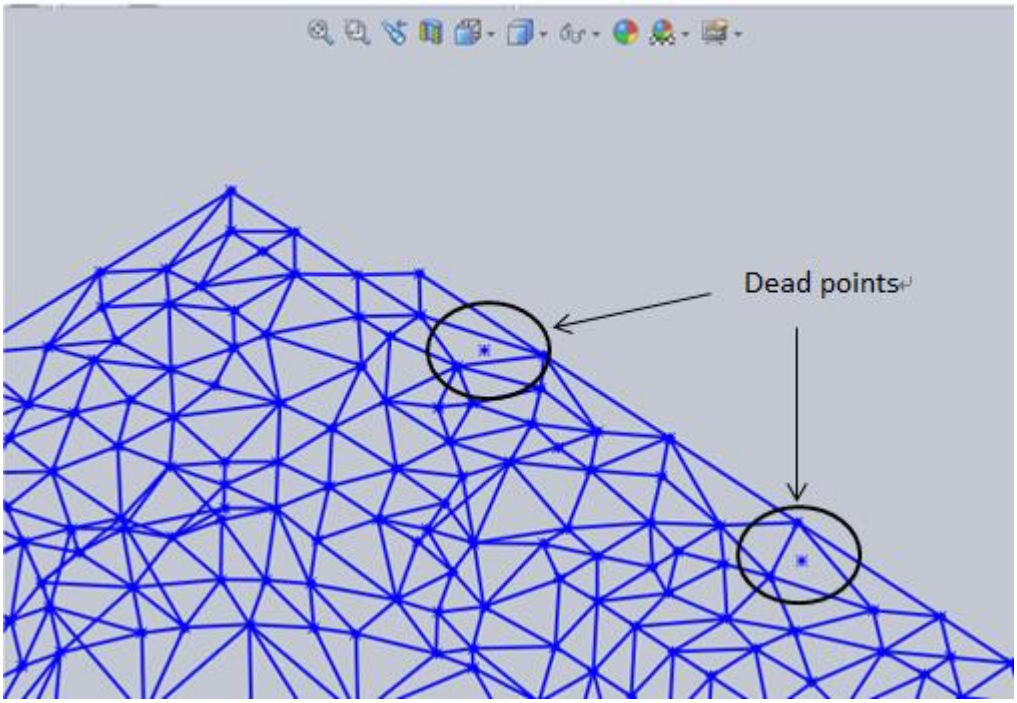


Figure 3. 15 Deadpoint on top surface of "Container"

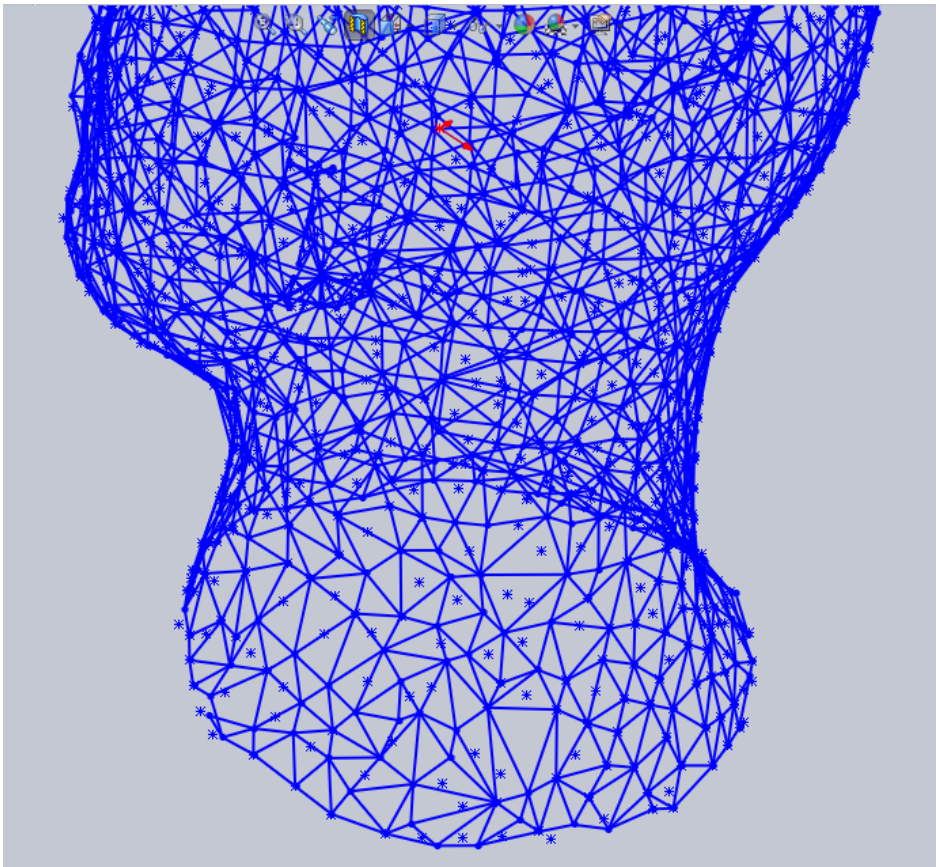


Figure 3. 16 Deadpoints on neck area of “Human head”

3.6 Algorithm conclusion

The idea of reconstruct the mesh surface by point cloud was:

1. Got the point cloud
2. Found the maximum and minimum value of it to create a boundary cube
3. Set a redutcion Coefficient to reduce and uniformized point cloud
4. Divided the after reduced point cloud into grids
5. Set seed triangle
6. Loop 1 for searching candidate point
7. Loop 2 for correcting the connection of points and edges

Chapter 4:

Results and Discussion

Chapter 4:

Results and Discussion

4.1 Algorithm Analysis

Algorithm in Chapter 3 had already been optimized by point cloud simplification and checking loop. The main frame of algorithm draft was about:

- Point cloud pre-processing

Read .txt file in Solidworks;

Compare the maximum and minimum value of coordinates;

Set boundary box to cover all coordinates;

Calculate the number of grids by equation set (3-1) [55, 57, 58];

Set seed triangle in flat area;

- Boundary extension judgment conditions

Search candidate point by the middle point of *ActiveEdge* in its neighbour grids;

Check if there was already a line between *ActiveEdge* and candidate point, if there was, marked the point as *DeadPoint*;

Check if each internal angle was greater than 20°;

Check if the dihedral angle was greater than 90°;

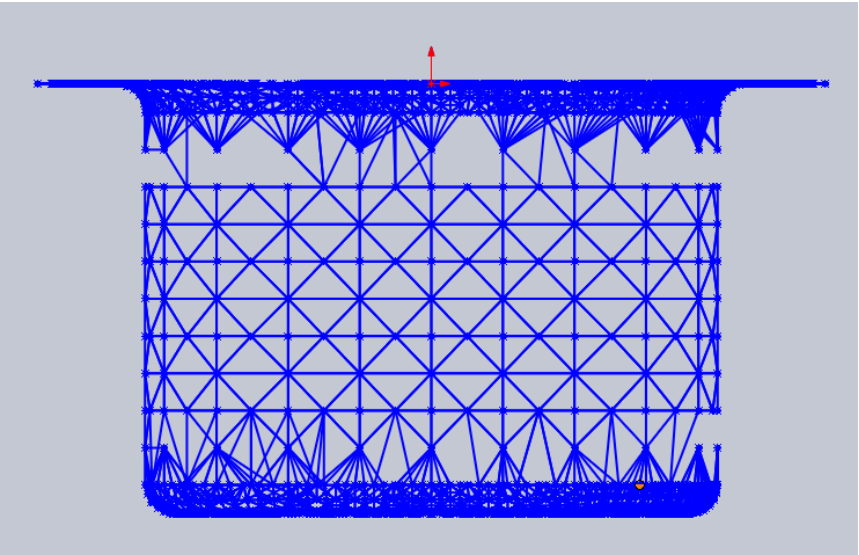
Chose the point met all requirements with a shortest distance to *ActiveEdge*.

Loop ended when there was no new edge could be made (ActiveEdgesCount = 0) and all points had been connected (For nlines = 1 To iline).

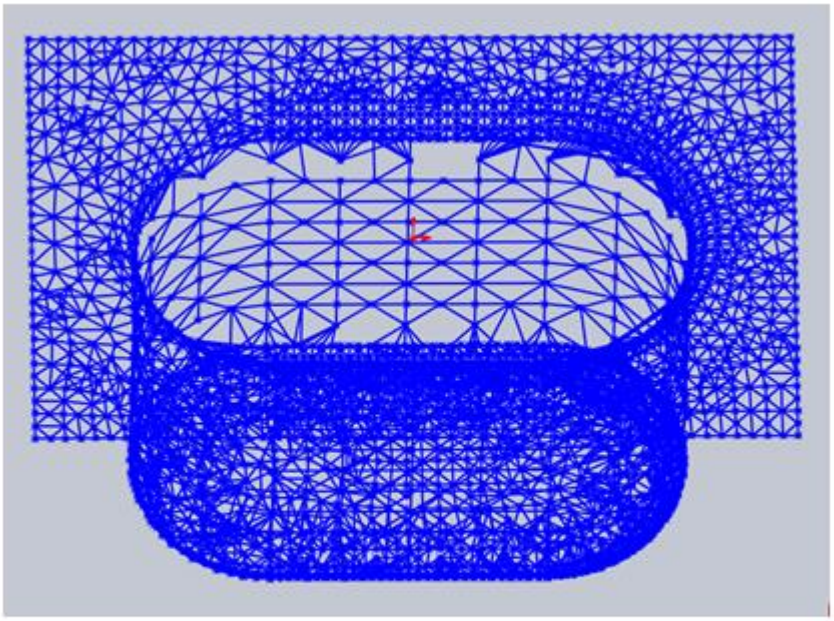
Generally, there were two loops during boundary extension. First loop stopped once there was a new *ActiveEdge* been created. Mark that line and save it in array *MarkLine*. In the first loop, it traversed through all points.

For the second loop, the *ActiveEdge* was being compared to its neighbour *ActiveEdge* to see which one was the best potential boundary edge of seed triangle extension. It would be checked if all connected lines of the *ActiveEdge* had former connection to target *ActiveEdge* and if there were any connections between the vertexes of *ActiveEdge* and candidate point.

There were conditions to run the second loop:
The target edge must be *ActiveEdge*;
The *ActiveEdges* being compared must be in neighbour grid
The vertexes of target edge must not be *DeadPoint*



(a)



(b)

Figure 4. 1 Front side of "Container" in first test (a); top side of "Container" (b) in first test

After the loop, the triangle mesh of "Container" was showed above. Due to comparison radius in the second loop, it did not perform well in the uneven density connected domain. But it generated a good quality mesh surface on top and bottom flat area and on cylinder region.

The main improvement of algorithm in Chapter 3 was it introduced point cloud simplification that provided even point cloud density to following operations. It had been tested in "Bunny" and "Human Head" as well, the results were relatively well. But there were still some shortcomings for apparent boundary models as show in Fig 4.2, the boundary of "Container" lost some features due to the point reduction, besides that, the main body of reconstruction was much better than the one without simplification.

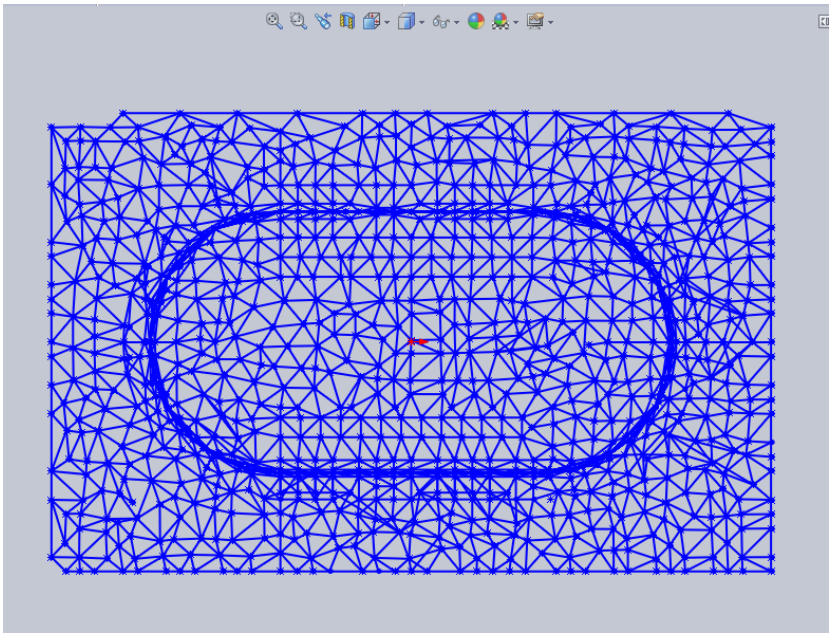


Figure 4. 2 Top side of “Container” mesh model

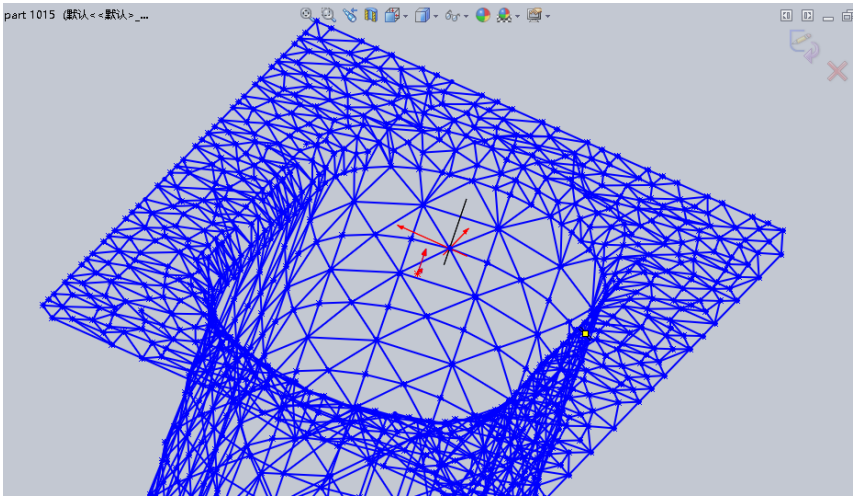


Figure 4. 3 Connected domain of “Container”

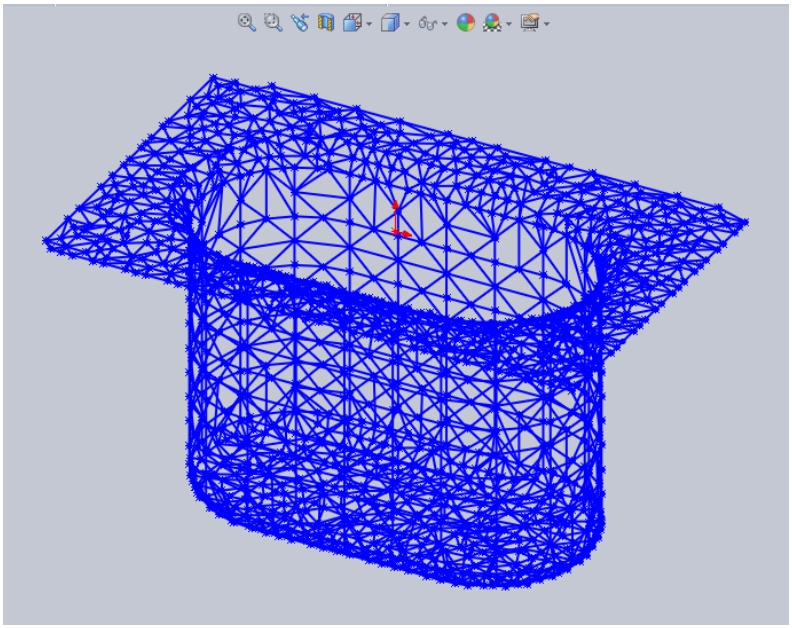


Figure 4. 4 Trimeric view of "Container"

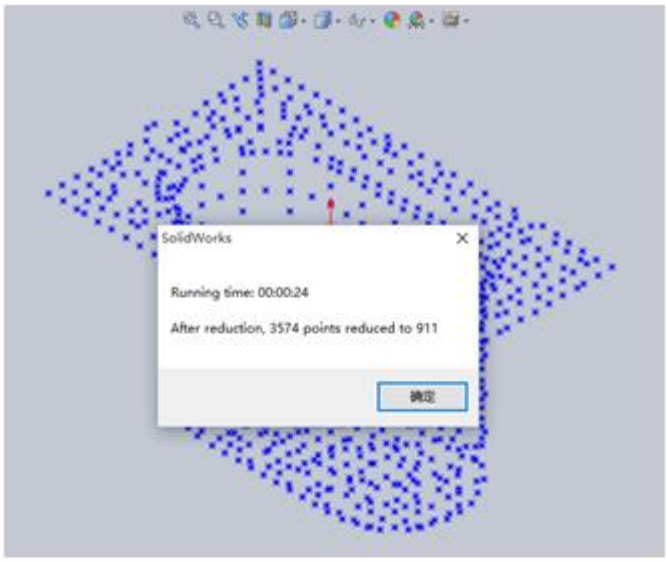
4.2 Reduction comparison

There were two main reasons to reduce the point number in point cloud. To those relatively even point cloud like "Bunny" and "Human Head", it was to increase the work efficiency and shorten running time. To those uneven density point clouds, like "Container", it was mainly for uniformizing density to avoid long and narrow triangle mesh and when the low quality mesh generated, it basically could not avoid wrong connections of edges, like line overlap, reverse connections.

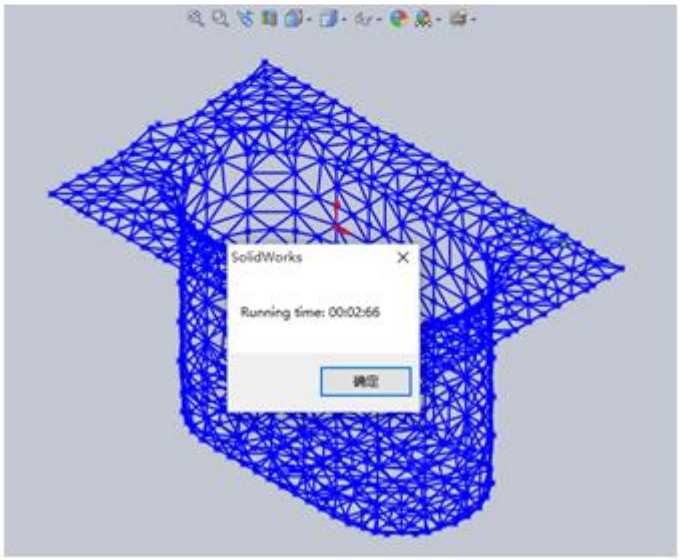
Since "reconductionCoefficient" was set manually, the precision could be controlled by different needs.

4.2.1 Reduction cube working efficiency test at different length

“Container” reduction cube length at 20mm



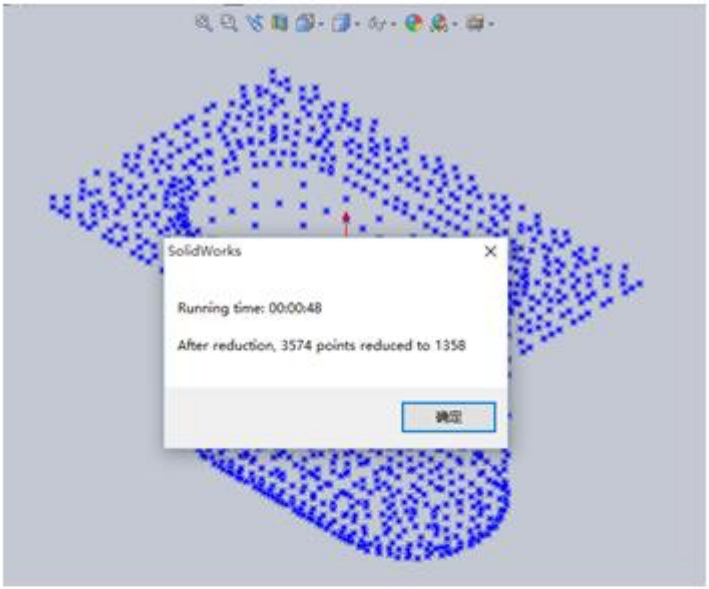
a



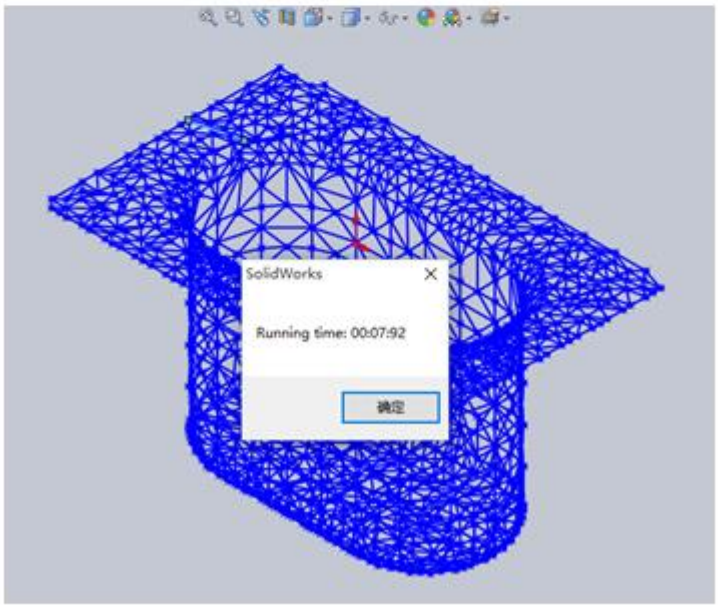
b

Figure 4. 5 **“Container” reduction test at cube length 20mm, reading time (a), reconstruction time (b)**

“Container” at 15mm



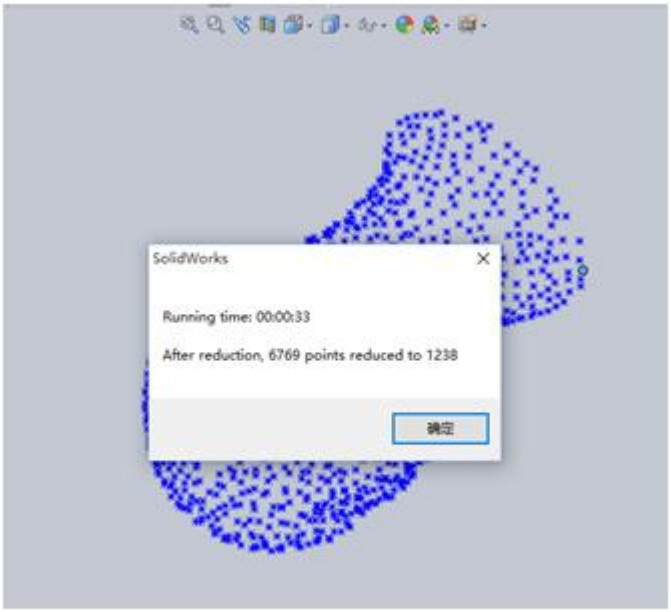
a



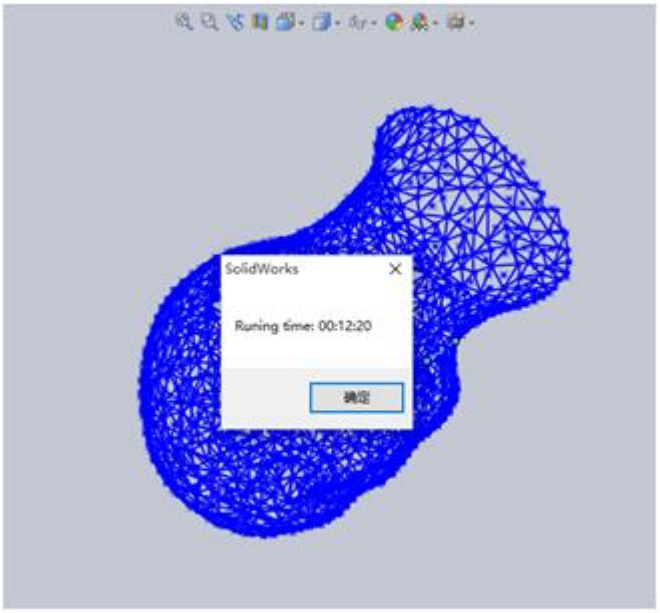
b

Figure 4. 6 “Container” reduction test at cube length 15mm, reading time (a), reconstruction time (b)

“Human Head” at reduction cube length 35mm



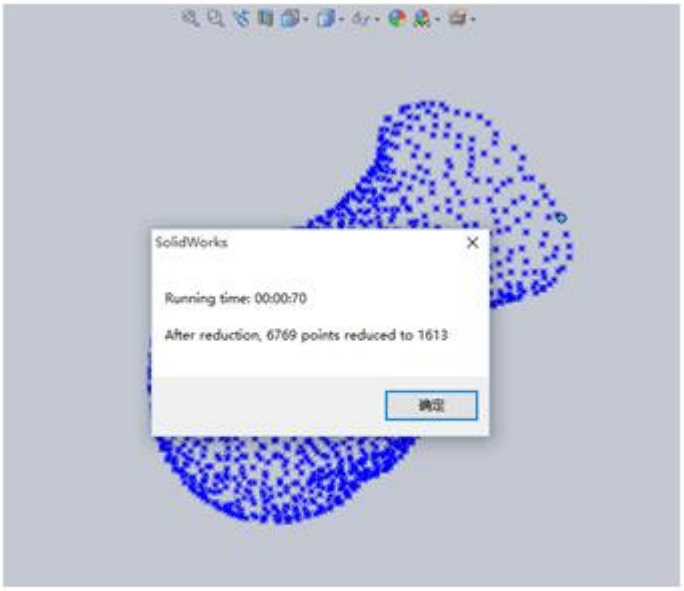
a



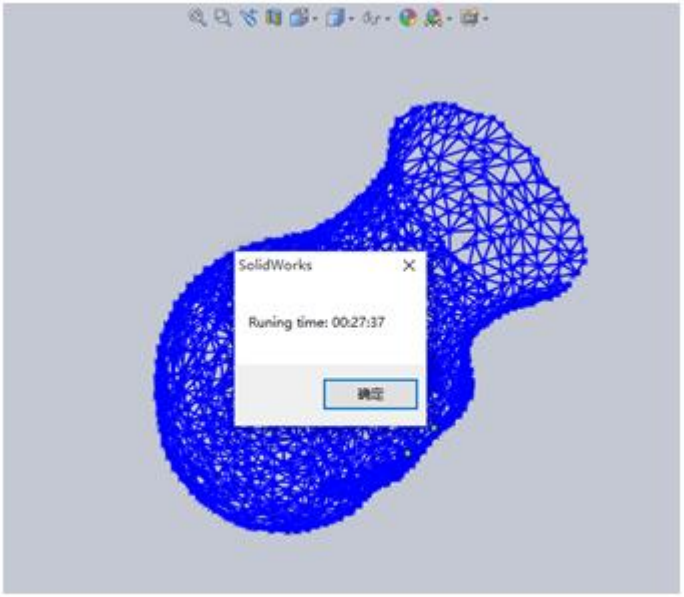
b

Figure 4. 7 “Human head” reduction test at cube length 35mm, reading time (a), reconstruction time (b)

“Human Head” at reduction cube length 30mm



a



b

Figure 4. 8 “Human head” reduction test at cube length 30mm, reading time (a), reconstruction time (b)

As shown in the figures above, the reduction of point could affect the quality and working efficiency directly. For “Container” model, the original number of points was 3574, after reduced to 911 by reduction cube length 20mm, it took only 24s to be imported into Solidworks, and the original time was 2:05s. For the original 6769 points of “Human head”, it took 6:51s to be imported into Solidworks. For 31607 points of “Bunny”, after reduction at 5mm reduction cube length it took 4:08s to open file in Solidworks.

4.2.2 Reduction cube working efficiency test by mesh quality

20mm reduction cube length

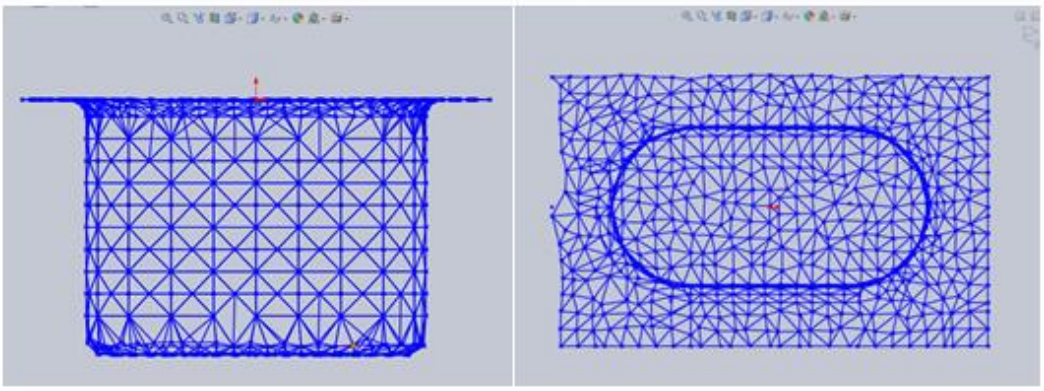


Figure 4. 9 “Container” reduction test at cube length 20mm, reading time (left), reconstruction time (right)

15mm reduction cube length of “Container”

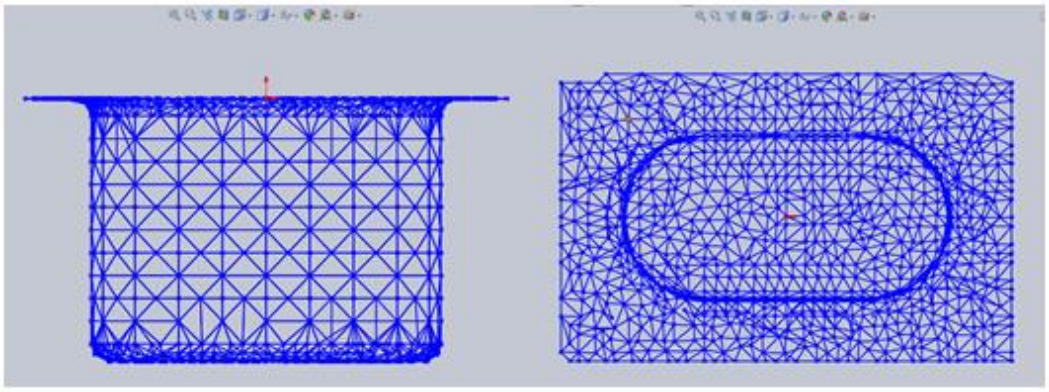


Figure 4. 10 “Container” reduction test at cube length 15mm, front side (left), right side (right)

For the “Container”, the best testing result was 15mm reduction cube length. Although in the main body part, the main feature had been restored, the boundary information had lost at 20mm reduction cube length. 10mm had been taken to the test as well, the connected domain issue started to appear, then it lost the main purpose of uniformizing density.

No wonder, the smaller reduction cube length provided a more accurate triangle mesh. Besides that, the main difference between 35mm (Fig 4.8) and 30mm (Fig 4.9) “Human head” reduction was, when the cube length was 35mm, there more *Deadpoint* on the surface by this algorithm, because those point were not in neighbouring grids but in the noise removal area. It was a waste of resource, because they participated every step of computing, but did not contribute to the precision and efficiency. When the reduction cube length adjusted to 30mm, the use ratio of point increased sharply. In “Human head” model, the better “reconditionCoefficient” was 30mm.

35mm reduction cube length of “Human head”

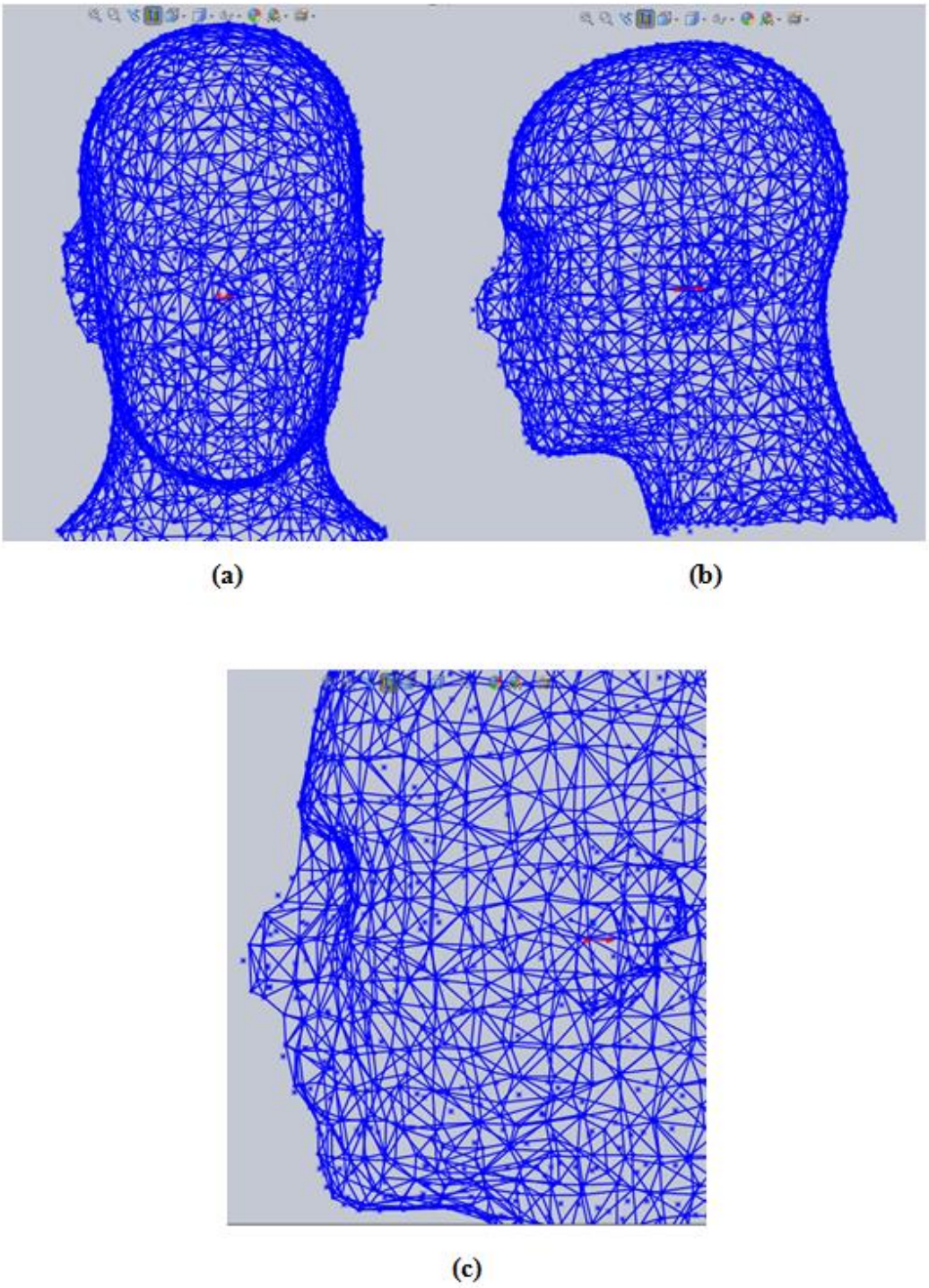
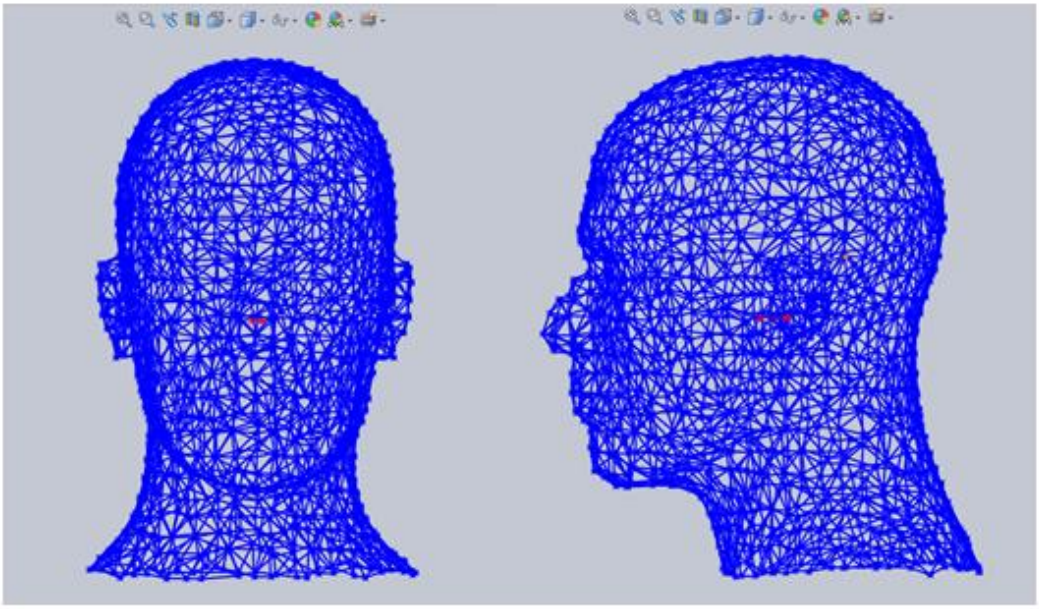
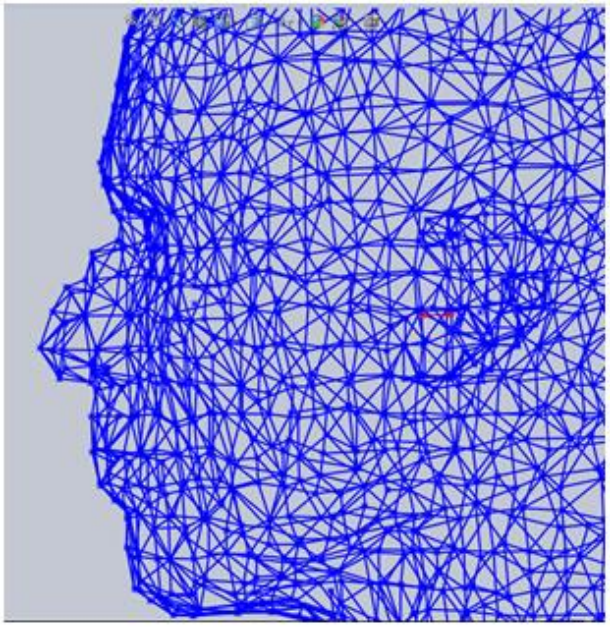


Figure 4. 11 “Human head” reduction test at cube length 35mm, front side (a), right side (b), side face contour (c)

30mm reduction cube length of “Human head”



(a) (b)



(c)

Figure 4. 12 “Human head” reduction test at cube length 30mm, front side (a), right side (b), side face contour (c)

10mm reduction cube length of “Bunny”

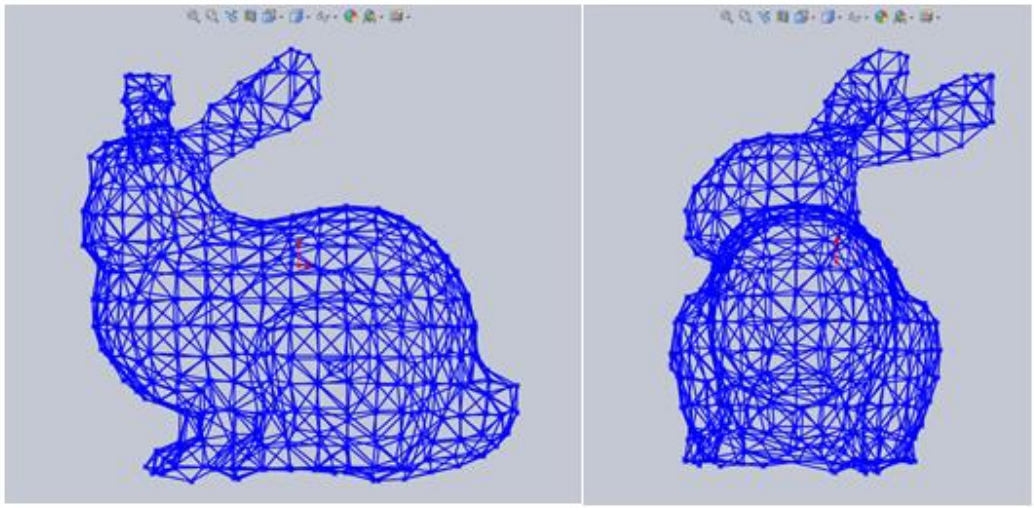


Figure 4. 13 “Bunny” reduction test at cube length 10mm, front side (left), right side (right)

7mm reduction cube length of “Bunny”

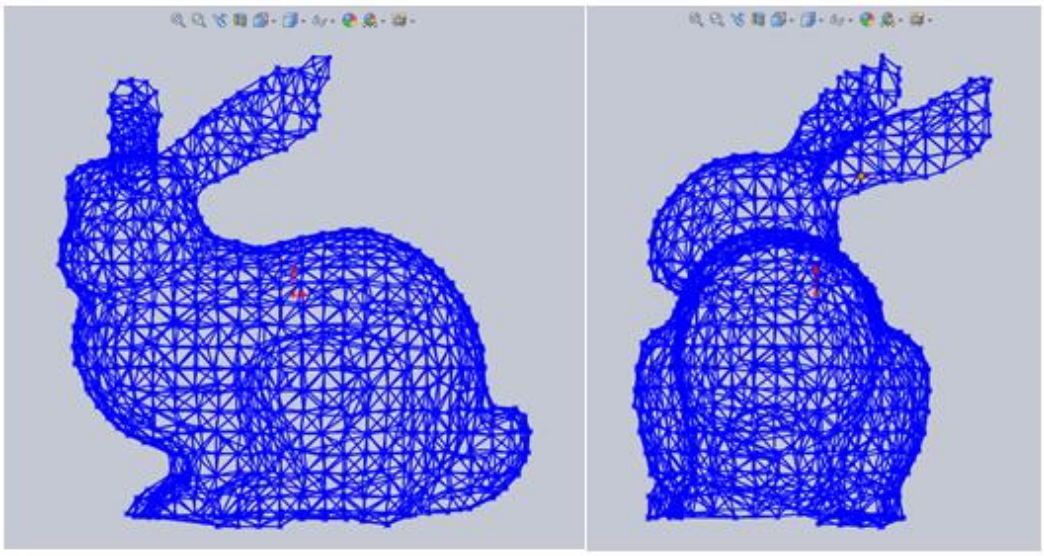


Figure 4. 14 “Bunny” reduction test at cube length 7mm, front side (left), right side (right)

The point number of “Bunny” was 31607 before reducing, when the reduction cube length were 10mm and 7mm, the point number reduced to 706 and 1413

respectively. Despite the ear areas, it generally did not contain other shape areas. Although a smaller triangle mesh which provided a smoother surface, 10mm cube length still kept the main feature of “Bunny” and shorten at least half of running time.

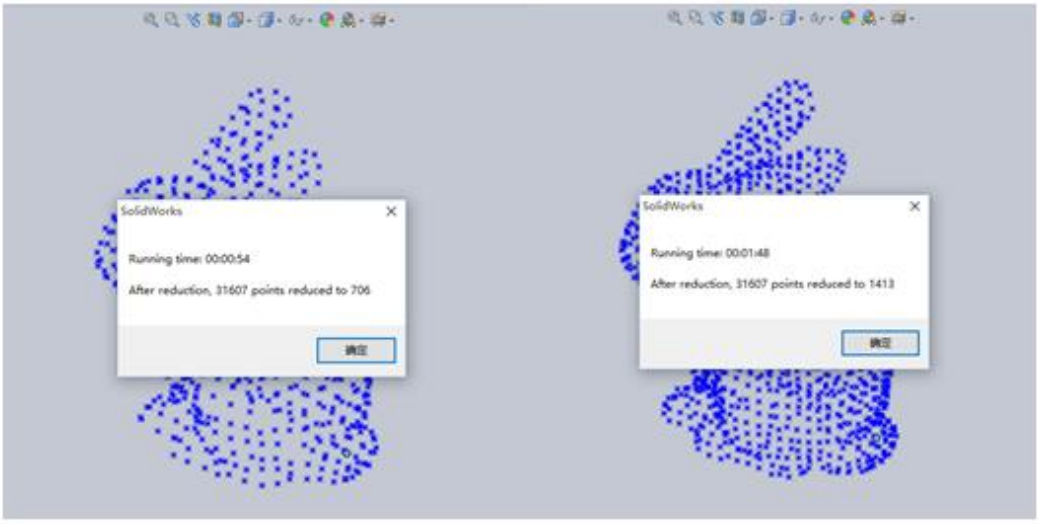


Figure 4. 15 “Bunny” point number at cube length 10mm (left), “Bunny” point number at cube length 7mm (right)

Chapter 5:

Conclusions and Recommendations

Chapter 5:

Conclusions and Recommendations

5.1 Conclusions

Unorganized point cloud triangulation is a major step of reverse engineering and the foundation of further surface processing. The estimated valued showed that the application of CAD/CAM had reduced the cost of engineering and technology by 13%- 30%, the time spent on design to manufacture by 30%-60%, the cost of labour force by 5%- 20%, meanwhile it had increased product quality by 5-15 times, the depth and breadth of analysing problems by 3-35 times, utilization rate of instrument and equipment by 30%-60%. Sale of CAD/CAM system increases 15% every year. [62]

In this research, a boundary cube-conduction method based on boundary extension had been achieved. It could rebuild the surface of point cloud effectively and efficiently. And the triangulation mesh was smooth and even. The main reason of that was because of dealing with the uniformization of point cloud in the first step and in the following algorithm, a portion of *DeadPoint* had been removed which avoided a mass of calculations and lines overlap. Since the point cloud was processed by cube-conduction and grid, two coefficients had been introduced into algorithm. Those coefficients could be adjusted to meet different needs of industrial. This algorithm is relatively neat and stable on point cloud with or without hole.

5.2 Recommendations

Reverse engineering has raised an increasing concern and made a great achievement in data processing, surface fitting, feature recognizing, commercial software improvement and coordinate measuring machine's development. There are still lots of man-machine interaction to be done in the practical use of reverse engineering, thus the experience and operating skill of a technician directly affect the quality of product, that why it could not be guaranteed to have a smooth new surface.

The following key technologies shall be the main development tendency:

(1) Data acquisition device: Develop the machine specific for reverse engineering to achieve a fast and high precision way of getting 3D digitizing data of the target product meanwhile providing a automatically measurement with path planning;

(2) Data processing: According to different kinds of data, keep developing the data processing software to improve the existing methods;

(3) Surface fitting: Control the fairness (smoothness and fluency) of curved surface and splice them together smoothly;

(4) Integration technology: Develop technology of reverse engineering field including data measurement technology, model reconstruction techniques, web-based collaborative design and digital manufacturing technology etc.

References

References

[1] Y. CAI, Q. XUE, and H. XU, *CAD/CAM Theory and Practice* vol. 15: China Machine Press, 1998.

[2] R. LLC. *WIREFRAME D MESH PRIMITIVES CLIP ART*. Available: http://www.clker.com/clipart-240621.html

[3] B. STACKPOLE. (2012). *Autodesk Expands Surface Modeling Credo*. Available: http://www.designnews.com/document.asp?doc_id=237728

[4] I. Bentley Systems. (2015). *Definition Of: Solid Modeling*. Available: http://www.pcmag.com/encyclopedia/term/51724/solid-modeling

[5] I. TEC-EASE. (2013). Available: http://www.tec-ease.com/gdt-tips-view.php?q=90)

[6] C. Metrology. (2012-2014). *Coordinate Measuring Machine History – Fifty Years of CMM History leading up to a Measuring Revolution*. Available: http://www.coord3-cmm.com/50-years-of-coordinate-measuring-machine-industry-developments-and-history/

[7] C. M. M. Inc. (2015). *DEA*. Available: http://www.cmmxyz.com/dea.html

[8] E. Béchet, J.-C. Cuilliere, and F. Trochu, "Generation of a finite element MESH from stereolithography (STL) files," *Computer-Aided Design,* vol. 34, pp. 1-17, 2002.

[9] Z. X.-h. L. Jing-hua and Y. Guang-rong, "Tool Path Generation Based on STL Data Model [J]," *Journal of Engineering Graphics,* vol. 1, p. 001, 2002.

[10] X. ZHANG, "CAD Key Technology In Basic Research And Application Of Dental Restoration And Occlusal Adjustment ", Nanjing University of Aeronautics and Astronautics, 2006.

[11] A. Kashef. (2015). *Tutorial-how to import points to solidworks from a text file!* Available:

https://grabcad.com/questions/tutorial-how-to-import-points-to-solidworks-from-a-text-file

[12] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models," in *An Introduction, ComputerAided Design*, 1997.

[13] Z. XU and C. SUN, *3D Reverse Engineering Techniques*: China Measure Publishing House, 2002.

[14] D. Industry. (2015). *Analog touch probe*. Available: http://www.directindustry.com/prod/solartron-metrology/product-4818-632558.html

[15] D. Industry. (2015). *Bridge coordinate measuring machine*. Available: http://www.directindustry.com/prod/carl-zeiss-industrielle-messtechnik-gmbh/product-5693-590699.html?utm_source=ProductDetail&utm_medium=Web&utm_content=SimilarProduct&utm_campaign=CA

[16] B. Journet and G. Bazin, "Laser range-finding techniques for industrial applications," *28th Int. Smp. Automotive Technol. Automation,* pp. 369-376, 1995.

[17] J. Aggarwal and Y. Wang, "Inference of object surface structure from structured lighting-an overview," *Machine Vision: algorithms, architectures and systems,* pp. 193-220, 1988.

[18] T. N. Chou and C. Wykes, "An integrated vision/ultrasonic sensor for 3D target recognition and measurement," 1997.

[19] E. Angelopoulou and J. R. Wright Jr, "Laser scanner technology," 1999.

[20] M. Petrov, A. Talapov, T. Robertson, A. Lebedev, A. Zhilyaev, and L. Polonskiy, "Optical 3D digitizers: Bringing life to the virtual world," *IEEE Computer Graphics and Applications,* pp. 28-37, 1998.

[21] T. Varady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-Aided Design,* vol. 29, pp. 255-268, 1997.

[22] J. Gu, *3D reconstruction of sculptured objects*, 1998.

[23] TOOLGUYD. (2010). *Bosch's New Long Distance Measuring Laser Range Finders*. Available: http://toolguyd.com/boschs-new-long-distance-measuring-laser-range-finders/

[24] B. L. Curless, "New methods for surface reconstruction from range images," Stanford University, 1997.

[25] H. LIN. (2003). *Discrete Geometric Information Process - From Points To Surface*.

[26] Q. WANG and R. WANG, "A Fast Progressive Surface Reconstruction Algorithm for Unorganized Points," *Journal of Software,* vol. 11, pp. 1221-1227, 2000.

[27] G. QIAN, "Study of unorganized point cloud grid reconstruction and optimization," Zhejiang University, 2008.

[28] CHRSTARK. (2013). *Scanning – Designing – Printing – All In 3d (Bringing The Schooner Zodiac Into The Digital Age For Its 90th Birthday)*. Available: https://fajexblog.wordpress.com/2013/12/26/turn-tall-ship-3d-scan-data-into-a-cad-model/

[29] Y. LIANG, "A Study on Adaptive Surface Mesh Generation," *Zhejiang University Press,* 2009.

[30] RAVENS. (2010). *3D Base mesh Tyranosaurus Rex dinosaur*. Available: http://www.10ravens.com/10ravensproducts/3d-base-mesh-tyranosaurus-rex-dinosaur-2.html

[31] 222chengzhicheng. (2011). *Detailed Annotation Of Dual-Core Processor Tegra2*. Available: http://222chengzhicheng.blog.163.com/blog/static/42082927201141006611295

[32] L. LI, "Research on the Theory, Method and Application of Surface Reconstruction from Scattered Points," *Mechanical Design and Theory,* 2001.

[33] R. Sibson, "Locally equiangular triangulations," *The computer journal,* vol. 21, pp. 243-245, 1978.

[34] A. Bowyer, "Computing dirichlet tessellations," *The Computer Journal,* vol. 24, pp. 162-166, 1981.

[35] D. F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The computer journal,* vol. 24, pp. 167-172, 1981.

[36] P. KUMAR. (2006). *Triangle++*. Available: http://compgeom.com/~piyush/scripts/triangle/

[37] Y. XIONG and Y. HU, "Surface triangle mesh algorithm based on mapping and Delaunay triangulation method," *Journal of Computer-aided Design and Computer Graphics,* vol. 14, pp. 56-60, 2002.

[38] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics (TOG),* vol. 13, pp. 43-72, 1994.

[39] N. Amenta and M. Bern, "Surface reconstruction by Voronoi filtering," *Discrete & Computational Geometry,* vol. 22, pp. 481-504, 1999.

[40] J.-D. Boissonnat and F. Cazals, "Smooth surface reconstruction via natural neighbour interpolation of distance functions," in *Proceedings of the sixteenth annual symposium on Computational geometry*, 2000, pp. 223-232.

[41] N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 415-421.

[42] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp. 249-266.

[43] Ag2gaeh. (2015). *Implicit surface torus (R=40, a=15)*. Available: https://en.wikipedia.org/wiki/Implicit_surface#/media/File:Torus-40-15.png

[44] Y. Ohtake, A. Belyaev, H.-P. Seidel, N. A. Dodgson, M. S. Floater, and M. A. Sabin, "Multi-scale and adaptive CS-RBFs for shape reconstruction from cloud of points," in *Advances in Multiresolution for Geometric Modelling*, ed: Springer, 2005, pp. 143-154.

[45] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Surface reconstruction from unorganized points* vol. 26: ACM, 1992.

[46] Y. LIN, C. CHEN, M. SONG, and J. BU, "Polar Field Based Implicit Surface Reconstruction," *JOURNAl OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS,* pp. 1035-1041, 2009.

[47] J.-D. Boissonnat, "Geometric structures of three-dimensional shape reconstruction," *ACM Trans. Graphics,* vol. 3, pp. 266-286, 1984.

[48] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 5, pp. 349-359, 1999.

[49] H.-W. Lin, C.-L. Tai, and G.-J. Wang, "A mesh reconstruction algorithm driven by an intrinsic property of a point cloud," *Computer-Aided Design,* vol. 36, pp. 1-9, 2004.

[50] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics,* vol. 43, pp. 59-69, 1982.

[51] I. P. Ivrissimtzis, W.-K. Jeong, and H.-P. Seidel, "Using growing cell structures for surface reconstruction," in *Shape Modeling International, 2003*, 2003, pp. 78-86.

[52] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, "Bayesian point cloud reconstruction," in *Computer Graphics Forum*, 2006, pp. 379-388.

[53] Jinger167. (2014). *bunny.txt - Stanford bunny*. Available: http://www.codeforge.cn/read/244921/bunny.txt__html

[54] Fearless. (2014). *3d_data_md.txt* Available: http://www.codeforge.cn/read/246506/3d_data_md.txt__html

[55] H.-p. XU, Z. MA, and W. De-feng, "Novel Data Processing Scheme For 3d Scattered Point Data In Reverse Engineering," *Journal of Engineering Design,* 2008.

[56] W. CHEN and X.-l. LIU, "A Fast Triangulation Algorithm for Unorganized 3-D Points [J]," *Computer Simulation,* vol. 9, p. 091, 2009.

[57] H. A. Sturges, "The choice of a class interval," *Journal of the American Statistical Association,* vol. 21, pp. 65-66, 1926.

[58] Y. KE and Q. WANG, "Research on Point Cloud Slicing Technique in Reverse Engineering," *JOURNAl OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS,* 2005.

[59] J. NIE, Z. MA, Y. HU, and X. CHEN, "Rapid Surface Reconstrcution Algorithm From Dense Point Cloud," *JOURNAl OF COMPUTER-AIDED DESIGN & COMPUTER GRAPHICS,* 2012.

[60] H. CHEN and S. CHEN, "Research on Triangulation Method of Object Surface with Holes in RE," *Design and Research,* 2006.

[61] D. LIU, W. LIAO, N. DAI, and X. CHENG, "Research And Implementation For Denoising Noisy Scattered Point Data," *Journal of southest university (Natural Science Edition),* 2007.

[62] Bbddbai. (2012). *The Application, Shortcomings And Improvement Of Reverse Engineering*. Available: http://wenku.baidu.com/view/aaeece210722192e4436f605.html

Appendix

Appendix

Imageware Products

Imageware products map

Usage \ Imageware Products	Surfacing	Inspection	Evaluation	Point Processing	Polygon Modeling	View
Real-time rendering			✔			
Modeling from sketch & drawing	✔		✔			
■ Class A surfacing	✔	✔(*)	✔	✔		
■ Rapid surfacing	✔	✔(*)	✔(*)	✔		
■ Reverse engineering	✔	✔(*)	✔(*)	✔		
■ ■ Point processing		✔(*)		✔		
■ Comparison of physical model and CAD model		✔				✔
Polygon modeling				✔	✔	
Rapid prototyping				✔	✔	
Data sharing (3D viewer)						✔

■ Surfacing
■ 3D inspection

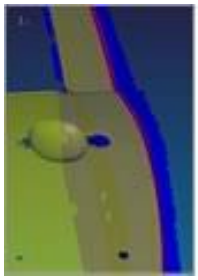
(*)...Additional modules

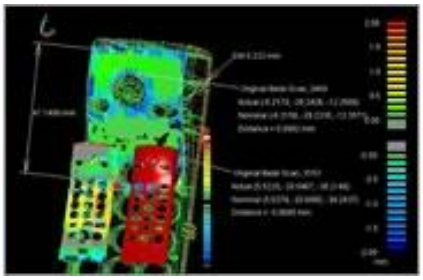
Imageware Surfacing

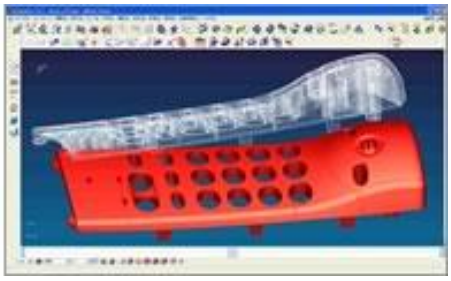
Imageware Surfacing generates free-form surfaces referring to point cloud data captured from a 3D measurement system and a design sketch. This product has the modeling functionality meeting variety types of requirements in the development process, from class A surfacing that is needed for styling through rapid surfacing that needs to be completed for a short time. Combination with the Point Processing feature realizes a high-end reverse engineering tool.

Imageware Inspection

Imageware Inspection has the functionality to compare point cloud data of a die or product captured from a 3D measurement system with CAD data, and to inspect the various aspects.
Imageware can be used for various inspections including measurement of faces, sections and dimensions, gap and flushness between two neighbor parts or assemblies by using measured data and CAD data. Imageware accept data from most of the 3D measurement systems regardless of touch probe or laser scanners, optical (camera), x-ray scanners type. The alignment functionality supports correction to the CAD coordinate system that cannot be aligned in a laser or optical type of measurement system.
* Inspection product includes all of the Point Processing functions.

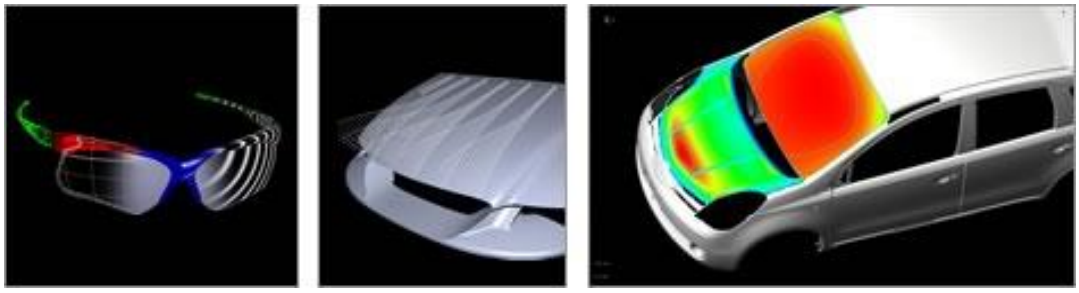






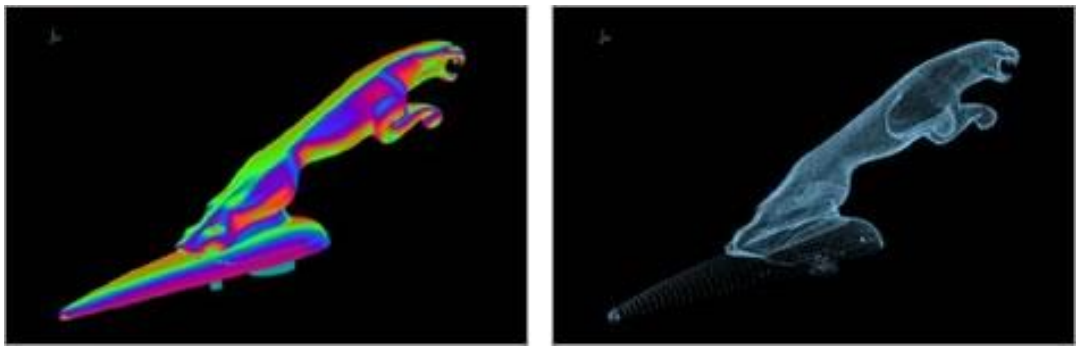
Imageware Evaluation

Imageware Evaluation has the data evaluation functionality to ensure a high-quality surfaced model. You can observe the quality of a curved surface by using the functionalities to display the curves, curvatures or inflection points generated referring to the measured point cloud, evaluate continuity, display elements in a high-lighted color, and apply zebra mapping.
Being combined with Point Processing or Surfacing, this Evaluation product can be used as a high-end styling tool.



Imageware Point Processing

Imageware Point Processing has the functionality to edit point cloud data captured from a 3D shape measurement system.
By importing point cloud data from variety types of measurement systems, removes noise and reduces point cloud data, merging and smoothing, extracting sections, making alignment, polygonizing, and modifying the shape, such data can be used as reference data for reverse engineering or as a model for measurement.
By using with Polygon Modeling, you can use point cloud data as a 3D model for RP system or as an analysis model.



Imageware Polygon Modeling

Imageware Polygon Modeling provides creation of a polygon mesh model from CAD data and measured point cloud data, verification and correction of errors in polygon meshing, filling, smoothing, thinning and division of polygon meshes. Modified polygon meshes can be used as a simplified 3D model for RP system, analysis and manufacturing.

Imageware View

Imageware View is the viewer to browse Imageware files.

You can observe color-mapped error with 3D data after inspection.