

Research Article

Generating Free-Form Grid Truss Structures from 3D Scanned Point Clouds

Hui Ding,^{1,2} Xian Xu,³ and Yaozhi Luo¹

¹Department of Civil Engineering, Zhejiang University, A-818 Anzhong Building, 866 Yuhangtang Road, Hangzhou, Zhejiang 310058, China

²Housing and Construction Bureau of Futian District, Shenzhen, Guangdong 518048, China

³Department of Civil Engineering, Zhejiang University, A-823 Anzhong Building, 866 Yuhangtang Road, Hangzhou, Zhejiang 310058, China

Correspondence should be addressed to Xian Xu; xian_xu@zju.edu.cn

Received 8 December 2016; Revised 5 August 2017; Accepted 13 August 2017; Published 20 September 2017

Academic Editor: Muhammad N. Akram

Copyright © 2017 Hui Ding et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reconstruction, according to physical shape, is a novel way to generate free-form grid truss structures. 3D scanning is an effective means of acquiring physical form information and it generates dense point clouds on surfaces of objects. However, generating grid truss structures from point clouds is still a challenge. Based on the advancing front technique (AFT) which is widely used in Finite Element Method (FEM), a scheme for generating grid truss structures from 3D scanned point clouds is proposed in this paper. Based on the characteristics of point cloud data, the search box is adopted to reduce the search space in grid generating. A front advancing procedure suit for point clouds is established. Delaunay method and Laplacian method are used to improve the quality of the generated grids, and an adjustment strategy that locates grid nodes at appointed places is proposed. Several examples of generating grid truss structures from 3D scanned point clouds of seashells are carried out to verify the proposed scheme. Physical models of the grid truss structures generated in the examples are manufactured by 3D print, which solidifies the feasibility of the scheme.

1. Introduction

Grid truss structures are widely used in large span roofs and shelters. Traditional grid truss structures usually have regular geometrical configurations, such as plates, spherical shells, and cylindrical shells. In recent years, with the increase of diversity in architectural appearances and functions, the demand of free-form grid truss structures has risen, especially in some public buildings. Generally speaking, the geometric prototype of free-form grid truss structures can be divided into two categories. The first category is wholly artificial forms, such as curved surfaces sketched by architects or free shapes created by CAD software. The other category is based on physical objects. Isler [1] has done a plenty of researches on the physical experiments and has proposed 39 different kinds of thin shell models. The initial shape of the single-layer glass grid shell over the courtyard of the renovated “Palacio de Comunicaciones” in Madrid was derived from an inflated balloon [2]. In the design of the Guggenheim

Museum in Bilbao, a physical model was handmade based on the architect’s draft and digitally scanned to supply the digitalized geometrical information for the design of the building [3]. The design method based on physical objects has also been applied to bionic structures [4].

To create free-form grid truss structures based on physical objects, the shape information of physical objects should be obtained firstly. A mass of point data (i.e., the point cloud) on the surface of an object are obtained through 3D scanning or 3D measurement technology. Besides, in structural/architectural design, 3D scanning has been widely studied and used in reconstructions of cities [5], buildings [6], pipelines [7], environments [8], and biomedical prostheses [9]. After the shape information is obtained, interpolation and fitting are carried out to generate a curved surface from the point cloud, and then the grid design is conducted on the surface [4]. The basic steps of the above scheme are shown in Figure 1. However, physical models generally have irregular shapes and the step from point clouds to surfaces

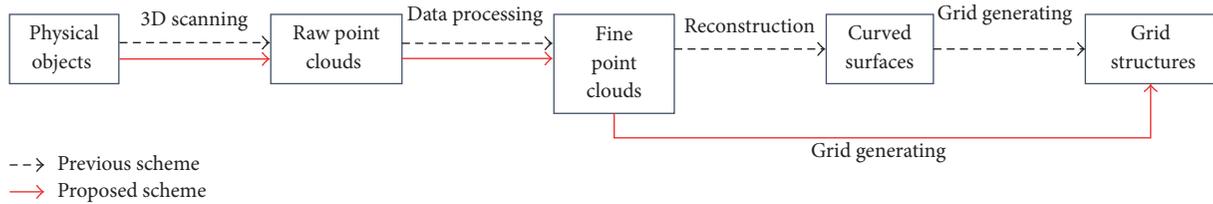


FIGURE 1: Flowchart of generating grid truss structures from physical objects.

is thus very complicated. Also, the step from point clouds to the surface is usually laborious and time-consuming. As the final geometrical model of a grid structure is an assembly of bars, the continuous surface information would not be required. Meanwhile, current 3D scanning equipment can collect the point cloud in a great density. It is thus sufficient to continuously describe the physical appearance in approximation. In this work, a scheme that skips the step of surface generating and directly generates grids from 3D point clouds is proposed. The basic steps of it are also shown in Figure 1.

For the problem of grid generating on point clouds, most previous researches were carried out in the area of computer graphics. The grid generating methods established in this field can be divided into three categories: sculpting-based approaches, contour-tracing approaches, and region-growing approaches [10]. In these approaches, grid or mesh is just used as a means for representation of surfaces and it is not the only means to achieve the purpose. Recently, there is also a trend of using point primitives for the processing and representation of surfaces [11, 12]. However, grid is the basic and necessary element for grid truss structures where each line of the grid represents a truss member and each node of the grid represents a joint. Moreover, computation efficiency is usually given a high priority by the approaches from the community of computer graphics, because their target applications include dynamic and large-scale surface reconstruction and visualization, while in the structural design of grid truss structures, the scale of the grid generating problem is limited and the requirement on the visualization is basic compared to those considered by the community of computer graphics. Hence, computation efficiency is usually not a problem faced by the designers of grid truss structures. In fact, the first concern of the designers is the grid quality, namely, the size uniformity and shape regularity of the grid. The grid quality is critical to a grid truss structure, because it will affect the mechanical performances and manufacturing cost of the structural system. The Finite Element Method (FEM) also has a high requirement on grid quality. There are numerous grid generating methods in FEM [13]. Among these methods, the mapping method [14], the advancing front technique (AFT) [15], and the paving method [16] are the most widely used. These methods also have been employed in grid generation for grid truss structures. For example, the grids of the single-layer glass grid shell over the courtyard of the renovated “Palacio de Comunicaciones” in Madrid and the grids of the “Sunshine Valley” of World Expo Shanghai 2010 were generated by a mapping procedure [2, 17]. However,

these methods can only be applied on surfaces and cannot be directly applied on point clouds.

In this paper, a novel scheme for generating grid truss structures from physical objects is proposed. The 3D scan technology is used to obtain point clouds from the surface of physical objects. A modified AFT is developed to directly generate grids from point clouds. Optimization and adjustment techniques are proposed to improve the quality and applicability of the grids. Examples are carried out to verify the proposed scheme. Physical models of the grid truss structures generated in the examples are manufactured by 3D print. In the previous works, surfaces must be reconstructed from the point cloud and then grid generating is conducted on the surfaces, while in this paper the grid generating is directly conducted on the point cloud, which significantly simplifies the geometrical modeling process and is the main novelty of the work.

2. 3D Scan and Point Clouds

2.1. 3D Scan. 3D scan is mainly used in reverse engineering. A large number of data points are obtained on the object surface using specific measuring equipment and methods. The mass data points are known as point clouds and stored in the given formation. 3D scan technology can be categorized into contact type and noncontact type, which depends on whether it comes into contact with the object to be scanned [18]. The noncontact type is dominating nowadays. There are noncontact scanners based on structural light, laser, and modulated light [19]. In this work, a 3D scanner based on structural light, Shining 3D four eyes scanner, is used to collect the point cloud. The scan range of the scanner is 400 mm × 300 mm to 200 mm × 150 mm. For an object beyond the range, it can be divided into a number of blocks within the scan range. The blocks are scanned separately and then spliced together to form a full point cloud of the object.

2.2. Pretreatment of Point Clouds. The process of scanning will inevitably be influenced by some factors such as ambient light and bearing devices. This may result in miscellaneous points and noise points which need to be removed. Sometimes, in order to reduce the workload of the data process, the point cloud needs to be streamlined. For multiple-view scans, splicing and other treatments are also required. All these processes belong to the pretreatment of point clouds. Many commercial software packages for reverse engineering can conduct it. In this paper, the software matched with a Shining 3D scanner is used.

3. Grid Generating

3.1. AFT. In the AFT, the boundary lines of a surface are discretized into segments around a given length. The discretized boundaries are known as the initial front. The segments in the front are also called front segments. The joints of the front segments are called nodes. Sequentially, a node that can compose a triangular grid with a front segment is generated by a given algorithm or selected from the existing nodes. The front is updated after a new grid is generated, namely, advancing towards the unmeshed area. The AFT has two characteristics. First, it is able to simultaneously generate nodes and grids [20]. The location of a node can be controlled as it is generated. Thus, the shape and size of a grid can be controlled to meet some given quality requirements [21, 22]. Second, it undertakes a large amount of intersection detection, inclusion detection, and distance detection during generating new grids [23]. The intersection detection occurs between line segments. The inclusion detection ascertains whether the grid contains a front node. The distance detection determines the distance between line segments and the distance between line segments and front nodes. The aforementioned detection types cost approximately 80% of the computer time during the implementation of the entire AF procedure. So, when the AFT is implemented, the data structure should be designed elaborately to reduce the amount of detection and improve the efficiency of the method.

3.2. Data Structure of Search Box. Much detection of intersection, inclusion, and distance is required in grid generating with the AFT. During the implementation of the AFT, these detection types cost much computer time. On the other hand, there is no explicit geometric topology relationship between the points in a point cloud. As a result, global searching should be carried out at every time of determining a proper node for a new grid. Since there are tremendous points in a point cloud, global searching will be very time-consuming and will heavily reduce the efficiency of the method. To reduce the search space and enhance computational efficiency, geometric topology relations between the points must be established. This issue is addressed in our work by using a search box.

The use of a search box is essentially a space partition strategy. Firstly, according to the distribution of the points, a rectangle bounding box containing all the points is formed in parallel with the whole coordinate axes. The bounding box is then divided into a number of $s \times m \times n$ cubes, namely, search boxes. Finally, every point is evaluated to determine which search box it is in. In specific, the searching box used in this paper is defined as follows [24].

The lower and upper bounds of the x , y , and z coordinates for a given point cloud were extracted. Bounding boxes were created in parallel with the x , y , and z axes according to the lower and upper bounds. The bounding boxes were then divided into a grid of cubic search boxes. An approximation of the size of a search box d_B is given by

$$d_{p_{\max}} < d_B < d_{H_{\min}}, \quad (1)$$

where $d_{p_{\max}}$ is the maximum distance between the points of the point cloud and $d_{H_{\min}}$ is the size of the minimum hole (if any) in the physical prototype of the point cloud. The number of search boxes in each coordinate direction is determined by

$$\begin{aligned} N_x &= \text{int} \left(\frac{x_{\max} - x_{\min}}{d_B} \right) + 1 \\ N_y &= \text{int} \left(\frac{y_{\max} - y_{\min}}{d_B} \right) + 1 \\ N_z &= \text{int} \left(\frac{z_{\max} - z_{\min}}{d_B} \right) + 1. \end{aligned} \quad (2)$$

The final size of the search box in each coordinate direction is obtained:

$$\begin{aligned} D_x &= \frac{x_{\max} - x_{\min}}{N_x - 1} \\ D_y &= \frac{y_{\max} - y_{\min}}{N_y - 1} \\ D_z &= \frac{z_{\max} - z_{\min}}{N_z - 1}. \end{aligned} \quad (3)$$

Each search box can be indexed by a set of numbers including three components. Each component represents the sequence number in a coordinate direction. Each point is put into search boxes according to the coordinates of it. The index of the search box that a point belongs to is determined by

$$\begin{aligned} N_{xi} &= \text{int} \left(\frac{x_i - x_{\min}}{D_x} \right) + 1 \\ N_{yi} &= \text{int} \left(\frac{y_i - y_{\min}}{D_y} \right) + 1 \\ N_{zi} &= \text{int} \left(\frac{z_i - z_{\min}}{D_z} \right) + 1, \end{aligned} \quad (4)$$

where x_i , y_i , and z_i are the coordinates of the point in x , y , and z directions, respectively, and N_{xi} , N_{yi} , and N_{zi} are the index of the search box in x , y , and z direction, respectively.

The location of a front segment is determined by its end nodes. Hence, a front segment belongs to the search boxes which its end nodes locate in. After all the points and front segments were embedded into the search boxes, the points and front segments in the neighborhood of a given point or front segment can be quickly determined. When searching for a point or carrying various detection instances for a front segment, the search boxes containing the target are selected as the search space.

3.3. Boundary of Point Cloud. The AFT generates grids on a surface from the boundary to the inside. For a surface, the boundary is clearly defined and can be extracted from the surface function. However, for a point cloud, there is no clear boundary. Before grid generating, it is thus essential that the boundary of the point cloud needs to be determined. An algorithm proposed by Qian et al. [25] is adopted and the

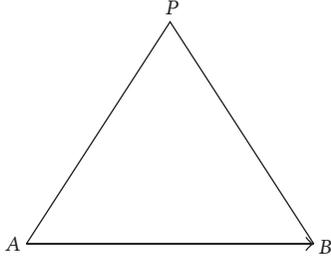


FIGURE 2: A candidate active front segment and a new node.

boundary points of a point cloud can be identified. Then, given a desired length for grid edges, some characteristic points were selected from the boundary points and connected by segments approximate to the desired length. These segments served as the initial active front. The front segments that can be advanced to generate new grids are called active front segments. Specifically, in the inner area, a front segment that is not sheared by two triangles is called an active front segment, and in the boundary a front segment that does not belong to any triangle is called an active front segment.

3.4. Choosing New Node. When AFT is used to generate grids, new nodes are needed to be selected from the points to form triangles with the active front. In theory, the ideal point for a new node should be a point that can form a regular triangle with the candidate active front segment. However, the ideal point generally does not exist. Hence, the point closest to the ideal point which is usually called the optimal point is selected as the new node. To determine the optimal point, a standard for selection should be defined first. The quality of a triangle is dependent on the shape and size of it. Therefore, a triangle quality factor is given as follows:

$$\delta = \alpha_1 \varphi + \alpha_2 \rho, \quad (5)$$

where φ is the shape quality factor [26]; ρ is the size quality factor [27]; α_1 and α_2 are weight factors. The size quality factor is given by

$$\begin{aligned} \rho_1 &= \min \left(\frac{l_1}{l_0}, \frac{l_0}{l_1} \right) \\ \rho_2 &= \min \left(\frac{l_2}{l_0}, \frac{l_0}{l_2} \right) \\ \rho &= \rho_1 \rho_2, \end{aligned} \quad (6)$$

where l_0 is the length of the candidate active front segment (i.e., AB in Figure 2) and l_1 and l_2 are the lengths of the segments connecting the new node and the ends of the candidate active front segment (i.e., AP and BP in Figure 2). Generally, the weight factors are chosen in according to the designer's requirement on the shape and size of the grids. In this paper, $\alpha_1 = \alpha_2 = 1$ is used, with which acceptable grids can be generated.

A new node must be located in the unmeshed area. To guarantee that, a new node P , as shown in Figure 2, should satisfy the following equation:

$$(\overrightarrow{AP} \times \overrightarrow{AB}) \cdot \mathbf{n} > 0, \quad (7)$$

where \mathbf{n} is the inner normal of the candidate active front segment AB . It could be approximately estimated by the inner normal of the triangle at the other side of AB .

In AFT, the intersection detection is the most frequent computation. A new generated grid should not intersect with existing ones. The intersection detection of triangular grids can be converted into the intersection detection of their edges. As proposed by Lo [28], there are four means for judging whether planar lines intersect. One of the means is that if two straight lines AB and CD satisfy (8), an intersection occurs.

$$(\overrightarrow{CA} \times \overrightarrow{CB}) \cdot (\overrightarrow{DA} \times \overrightarrow{DB}) < 0. \quad (8)$$

Equation (8) is only applicable in a planar space, while the case considered in this paper is in a 3D space. Considering that the lines potentially intersecting are close to each other, they are projected to a tangent plane with the least squares method and then the above means can be employed. Moreover, (8) is established for lines with infinite length. To avoid misjudgment when the extensions of two disintersected line segments are intersecting, an additional criterion is formulated as

$$(\overrightarrow{AC} \times \overrightarrow{AD}) \cdot (\overrightarrow{BC} \times \overrightarrow{BD}) < 0. \quad (9)$$

Intersection of line segments AB and CD is detected, if and only if both (8) and (9) are met.

The overall process of generating a new grid includes three steps as follows:

- (1) The optimal point of the candidate active front segment is identified from the point cloud excluding the points already chosen as nodes. As shown in Figure 3, the vertical split plane Γ of the candidate active front segment AB is determined [29], and all data points P that meet the following conditions are then searched:

- (1) $d_1 \leq b$, d_1 is the distance from P to Γ and b is a given constant.
- (2) $r - h \leq d_2 \leq r + h$, d_2 is the distance from P to the candidate active front segment, h is a given constant, and r is given by

$$r = \frac{\sqrt{3}}{2} (w_1 l_e + w_2 l_0), \quad (10)$$

where l_e is the desired length for grid edges, l_0 is the length of the candidate active front segment, w_1 and w_2 are weight factors, and $w_1 + w_2 = 1$.

- (3) P constitutes a triangle with the candidate active front segment AB , and the triangle meets (7).

TABLE 1: Six cases in advancing a front segment.

Number	Before advancing (thick = candidate; thin = neighbors)	After advancing (solid = candidate active front segments; dash = inactive front segments)
(1)	Triangle	
(2)	quadrangle	
(3)	$\alpha < 90^\circ$ $\beta < 90^\circ$	
(4)	$\alpha \geq 90^\circ$ $\beta \geq 90^\circ$	
(5)	$\alpha \geq 90^\circ$ $\beta < 90^\circ$	
(6)	$\alpha < 90^\circ$ $\beta \geq 90^\circ$	

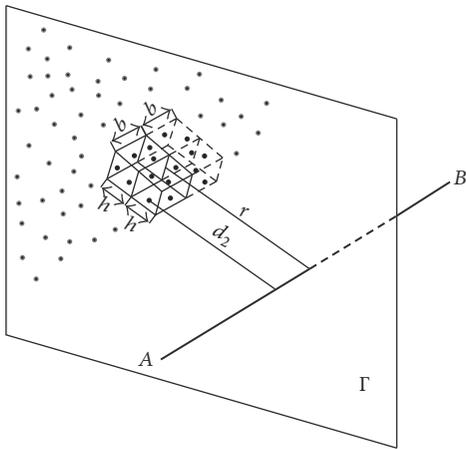


FIGURE 3: Definition of searching space for optimal point.

The first two conditions determine the range of the searching space and the third condition determines the direction of the searching. The constants b and h are determined according to the point distances of the point cloud and the desired size of the generated grids. Generally, when the values of b and h are 3 times the maximum point distance, acceptable results are guaranteed, supposing that the desired grid size is greater than 3 times the maximum point distance. The point that forms the best quality triangle with the candidate active front segment is selected from the

search space. The point is known as the optimal point P_o .

- (2) The distances from P_o to the surrounding front segments are calculated. If all the distances are not less than $l_e/2$, P_o is turned to a new node and the triangle is composed of P_o and the candidate active front segment is kept as a new grid. Otherwise, turn to Step (3).
- (3) All the surrounding front segments with a distance of less than $l_e/2$ to P_o are extracted. Each end node of the front segments can constitute a triangle with the candidate active front segment. The quality factors of the potential triangles are evaluated. The triangle having the greatest quality factor is chosen as a new grid.

3.5. Overall Scheme. The overall scheme of the proposed grid generating method is shown in Figure 4.

Front advancing is regarded as the most critical step in the scheme. In this paper, the main principle of front advancing is based on the method proposed by Foucault et al. [30], but some modifications were conducted to take the characteristics of grid truss structures into account. In each trial of front advancing, there are six possible configurations before advancing for the candidate active front segment, and correspondingly there are six possible results after advancing, as shown in Table 1. After all the candidate active front segments were advanced, a check is imposed on the new front to determine whether there were new candidate active front

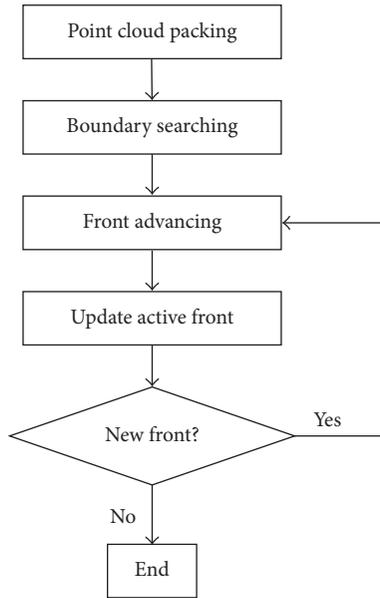


FIGURE 4: Overall scheme of grid generating.

segments. If there were new candidate active front segments, the database of new candidate active front segments is updated and the front advancement is carried out again. If there is no new candidate active front segment, the scheme is finished. For cases (3)–(5) in Table 1, Foucault et al. used 80° as the criteria [30]. But it is found that long and narrow grids might be generated when 80° is used. To avoid this, 90° instead of 80° is used in this paper. Moreover, for case (3), Foucault et al. connected the free nodes of the neighboring edges to construct a quadrangle, and then it is divided into two triangles by adding a diagonal [30]. With that strategy, there is a great possibility to generate grids with small angles which would cause contact of members and is not desired by grid truss structures. To avoid generating grids with small angles, a new strategy where the free nodes of the neighboring edges merged into the point closest to them is adopted, as shown in Table 1. For case (4), a new node needs to be chosen from the points in the point cloud. The procedure presented in the last subsection (i.e., in Choosing New Node) is tailored for this case.

3.6. Grid Generating for Multiple Boundaries. There may be some internal areas without data points in a point cloud. The areas without data points which look like holes may be caused by real holes in the physical object or by purposely removing data points from the point cloud for some special applications. The boundary of such a hole is called an internal boundary. The determination and connection of the internal boundary are similar to those of the external boundary.

There are two ways of front advancing for a point cloud with internal boundaries. One is advancing from both external and internal boundaries. The other is only advancing from the external boundary and treats the internal boundaries as inactive front segments. The first way can guarantee a good quality for the grids on the boundaries. However, as shown in

Figure 5(a), there is a considerable area of front interference in the middle of the configuration, which causes poor quality and irregular distribution of grids in the middle area. The second way can effectively maintain regularity of the grids in most areas except for the grids near the internal boundary, as shown by Figure 5(b). In this work, the second way is adopted after balancing the appearance and quality of the grids.

4. Grid Optimization and Adjustment

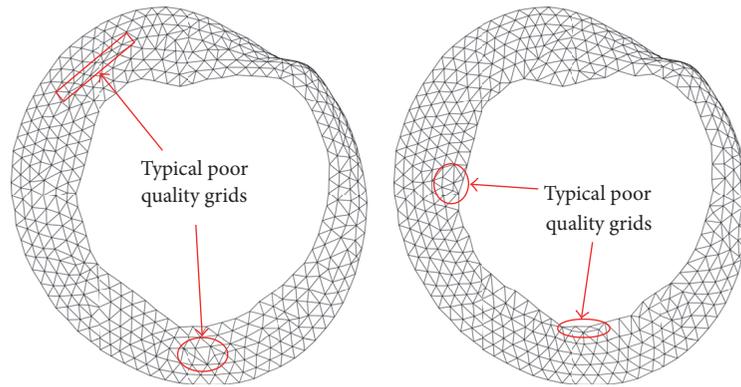
The regularity and uniformity of the grids affect the structural performances and manufacturing cost of the grid structure. There is space for further optimization of the grids generated by the AFT. Meanwhile, in practical applications, the locations of supports are usually assigned in advance. Therefore, the grids are needed to be adjusted to ensure that there are nodes for supporting in the given locations.

4.1. Optimization Using Delaunay and Laplacian Methods. To improve the quality of the new generated grids, an optimization scheme consisting of two steps is developed. Firstly, Delaunay optimization is performed to improve the topology of the grids. Then, Laplacian optimization is used to improve the regularity and smoothness of the grids.

In planar space, the Delaunay triangulation requires the circumcircle of each triangle to not contain in its interior any other vertex which is visible from the vertices of the triangle [31], which is also called the empty circle principle. More specifically, for a quadrangle consisting of two triangles with a common edge, if the circumcircle of any of the triangles contains other nodes, the diagonal of the quadrangle is switched. Nakahashi and Sharov have applied this principle to optimize the planar grids generated by AFT [32]. But it cannot be directly applied to grids in the 3D space, because the edges of the quadrangle may be in different planes. It has been proved that the principle of empty circle is equivalent to maximizing the minimum interior angle of the triangles [31]. This means that the diagonal should locate in a way that the minimum angle of the triangles is maximized. This alternative strategy is applicable to grids in the 3D space. Hence, this alternative strategy is adopted here.

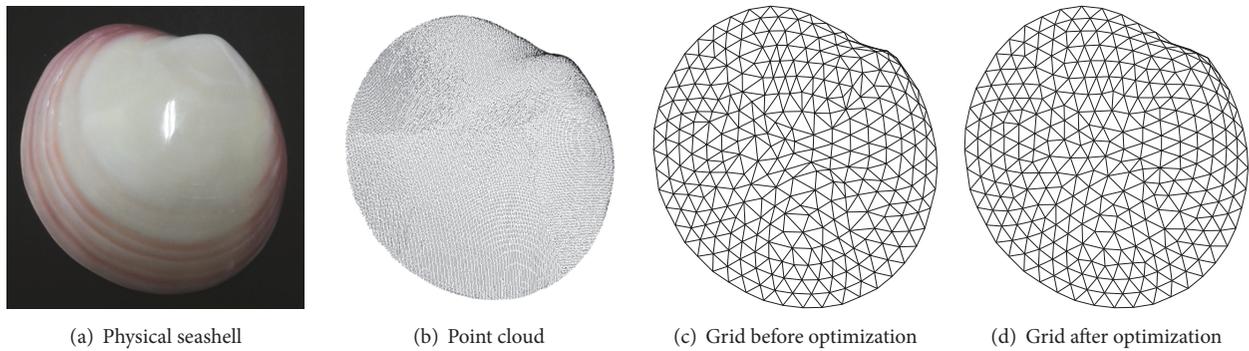
The grids generated by AFT cannot completely satisfy the requirement of regularity for grid truss structures, especially in the area where the front interference occurs. Smoothness optimization is usually used to increase the regularity of the grids. The Laplacian algorithm is the most widely used method for smoothness optimization, due to the simplicity of its principle and its ease for implementation. Nevertheless, the Laplacian algorithm has a considerable shortcoming that it will lead to the shrinking of the grids. The degree of shrinkage increases as the number of iterations increases. A number of works have been carried out to address the shrinkage of the Laplacian algorithm [33–35]. Based on the work of Taubin [34], a strategy where reverse optimization is conducted after every positive optimization to counteract the shrinkage is employed.

In previous work [34], it is revealed that a larger number of iterations would lead to a smoother grid. But it did not give any convergence condition for the iterations. In this paper,



(a) Advancing from both external and internal boundaries (b) Advancing from external boundary

FIGURE 5: Two ways of front advancing for a point cloud with an internal boundary.



(a) Physical seashell (b) Point cloud (c) Grid before optimization (d) Grid after optimization

FIGURE 6: Example 1.

three convergence conditions are proposed according to the requirement of grid truss structures. The iterations stopped as soon as any one of the following conditions is satisfied:

- (a) The shape quality factor of the grid is no longer improved.
- (b) The standard deviation of the member lengths is no longer reduced.
- (c) The average deviation of the nodal coordinates is larger than given criteria.

The first condition is used to prevent the shape quality of the grids from becoming poor. The second condition is used to prevent the size of the grids from becoming uneven. A large number of studies have established that the shape quality of the grids will continuously improve along with iterations, but the standard deviation of the member lengths will first decrease and then increase. The third condition is used to avoid distortion in the geometrical configuration. Although the reverse optimization effectively reduces the shrinkage, it also brings about some small distortions. As the number of iterations increases, the distortions may accumulate to considerable magnitude. The criteria for the average deviation of nodal coordinates are usually appointed by designers according to their requirement on the shape accuracy of the grid structure.

4.2. Handling of Fixed Nodes. Due to the practical needs of supports or equipment installation, it is usually required that there are nodes located in some appointed positions. These nodes are called fixed nodes.

The fixed nodes can be treated as predetermined nodes in the frame of AFT. This treatment is fairly straightforward, but the actual operation is difficult due to the uncertainty of the locations of the fixed nodes. It will increase the complexity of the process of front advancing and additional judgment conditions are required. In this paper, a simple technique different from the above one is adopted. The fixed nodes are not taken into account in the process of grid generating. After grid generating and before Laplacian optimization, the nodes nearest to the appointed positions are identified and then moved to the appointed positions to become the fixed nodes. Finally, Laplacian optimization is imposed on the grids with the fixed nodes. The above operations did not require any additional computer cost in AFT, while the quality of the final grids is maintained to some extent.

5. Examples

The proposed approach is coded into a Visual C++ program. Four examples that use three seashells (Figures 6(a), 7(a), and 8(a)) and a plaster model (Figure 9(a)) as the physical

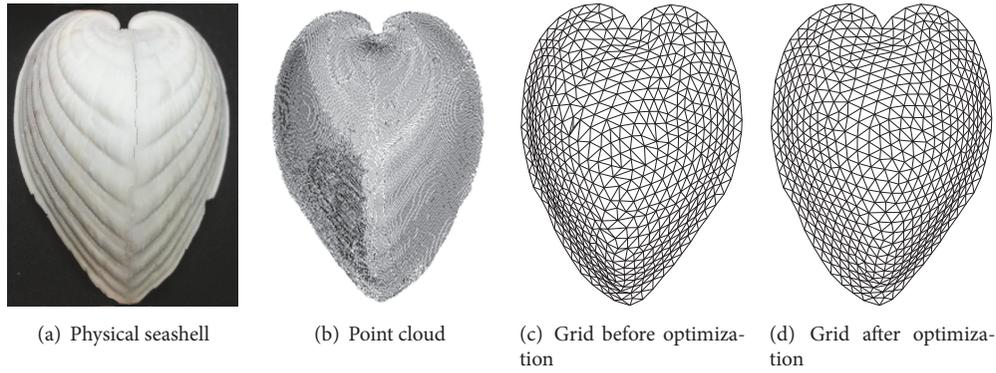


FIGURE 7: Example 2.

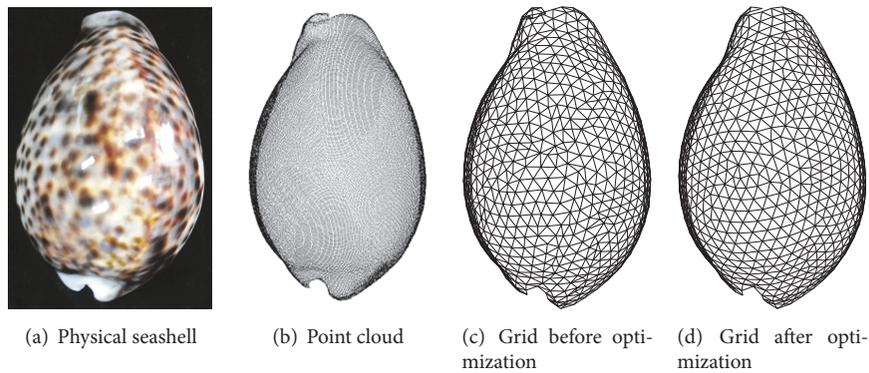


FIGURE 8: Example 3.

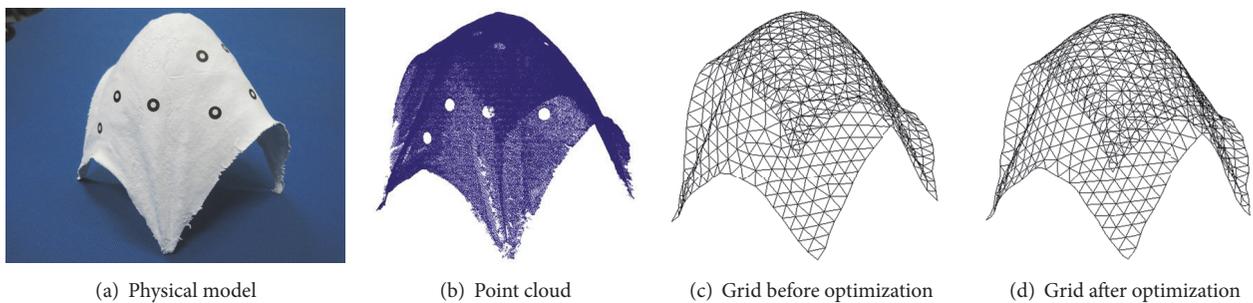


FIGURE 9: Example 4.

prototypes are carried out on the program ran by a computer using a 1.60 GHz Intel(R) Core(TM) CPU with 4.00 GB RAM. Note that the plaster model has a shape resembling a shell generated by suspending a rectangle textile at the four corners. Point clouds (Figures 6(b), 7(b), 8(b), and 9(b)) are obtained by 3D scanning with the scanner mentioned above. Triangular grids (Figures 6(c), 7(c), 8(c), and 9(c)) are generated from the point clouds using the proposed AFT. Then, the Delaunay and Laplacian algorithms are applied to optimize the grids. Apparently, the grids after optimization (Figures 6(d), 7(d), 8(d), and 9(d)) look more regular and uniform than those before optimization. Table 2 gives the shape quality coefficients and member length coefficients of the four examples. The average values of the shape quality

coefficients showed consistent improvement, indicating that the shapes of grids got much closer to regular triangles. Both the standard deviations of the shape quality coefficients and the standard deviations of the member lengths are significantly decreased, which revealed that the shape and size of the grids become more uniform. The average length of the members decreased after optimization. This is caused by the fact that, for triangles with various shapes but the same area, the regular triangle is the one having the shortest perimeter. The computation times for the four examples are given in Table 3.

For comparison of computation efficiency, the B-spline representation [4] is used to reconstruct surfaces from the point clouds given in the above examples and then grids

TABLE 2: Comparison of grid quality parameters.

Examples	Mean of shape quality coefficient \bar{G}	Standard deviation of shape quality coefficient δ_G	Mean of member lengths \bar{L}	Standard deviation of member lengths δ_L
Example 1				
Before optimization	0.932	0.075	4.773	0.740
After optimization	0.962	0.041	4.701	0.566
Example 2				
Before optimization	0.940	0.073	1.915	0.285
After optimization	0.970	0.035	1.907	0.208
Example 3				
Before optimization	0.954	0.068	3.907	0.510
After optimization	0.981	0.028	3.886	0.369
Example 4				
Before optimization	0.944	0.074	2.760	0.478
After optimization	0.974	0.038	2.745	0.327

TABLE 3: Comparison of computation times.

Examples	Example 1	Example 2	Example 3	Example 4
Times by the proposed approach (s)	34	42	39	27
Times by comparison computations (s)	398	705	553	414

are generated on the surfaces by the conventional AFT applied on surfaces. The same coding language and computer are used to conduct the comparison computations. The grids obtained by the comparison computations are similar to those obtained by the proposed approach and are not shown here for the sake of simplicity. The times used in these comparison computations are also given in Table 3. It can be observed that, for each example, the computation time needed by the proposed approach is significantly less than that needed by the comparison computation which includes a surface reconstruction procedure. It should be pointed out that the B-spline representation used in the comparison computations is not guaranteed to be the most efficient method for surface reconstruction since there are various surface reconstruction methods [10, 12]. In fact, in civil/structural engineering, the scale of the grid generating problem is limited and the requirement on the visualization is basic compared to those considered by the community of computer science. As a result, structural designers are more concerned with the convenience of the overall design scheme than with the pure computation efficiency. In the design practice of a free-form truss structure, multicomputer-aided design packages are usually needed and the interactive operations with the computer-aided design packages and the design data transmission between different packages usually cost much labor and time. Skipping a main step (i.e., surface reconstruction) definitely will reduce the manual operations and data transmission and make the whole design scheme easier for the structural designers.

Three additional examples 5–7 are conducted to demonstrate the capability of the proposed approach in dealing with point clouds with multiple boundaries and fixed nodes. In Example 5, some points are removed from the point cloud

of Example 1 and a new point cloud with two internal holes is obtained (Figure 10(a)). The result of boundary searching is given in Figure 10(b) and the generated grids after optimization are shown in Figure 10(c). In Example 6, a new point cloud with an elliptic internal hole is formed by removing some points from the point cloud of Example 3 (Figure 11(a)). This configuration is specially suitable for stadiums with open roofs. The results of boundary searching and grid generating are shown in Figures 11(b) and 11(c). In Example 7, three fixed nodes a , b , and c are appointed on the basis of Example 1. The proposed technique is used to handle the fixed nodes. The final grid obtained is shown in Figure 12. It is clearly shown that there are nodes located in the appointed positions. It is worth noting that there is a tiny distortion in the neighboring area of node a , because node a is so close to the boundary that a node on the nearby boundary has merged with it.

6. Physical Models

3D print technology is used to manufacture physical models for the grid truss structures obtained in Examples 1–3. The 3D printed models had successfully copied the geometrical shapes of the physical prototypes but with latticed structures, as shown in Figures 13–15.

7. Conclusions

In this paper, a scheme for generating grid truss structures from 3D scanned point clouds is proposed. A modified AFT is established and used to generate grids directly from the point cloud. The Delaunay and Laplacian algorithms are employed to optimize the grid generated by the AFT. Some special and practical techniques are tailored to address the

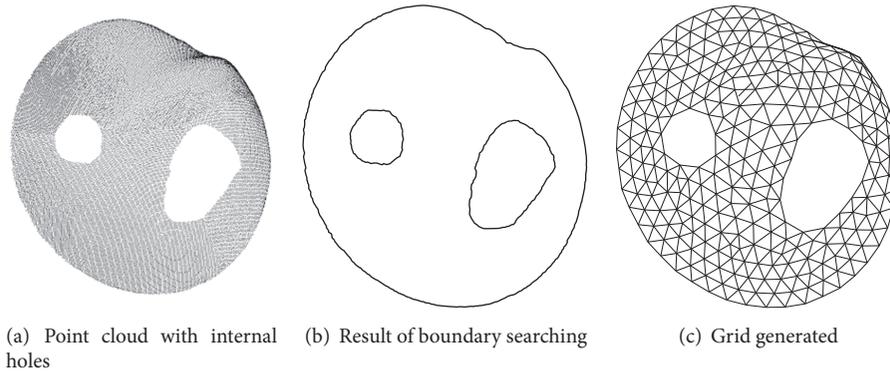


FIGURE 10: Example 5.

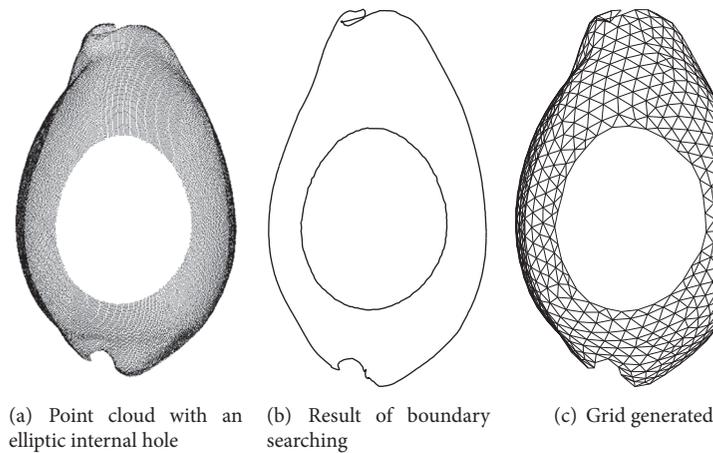


FIGURE 11: Example 6.

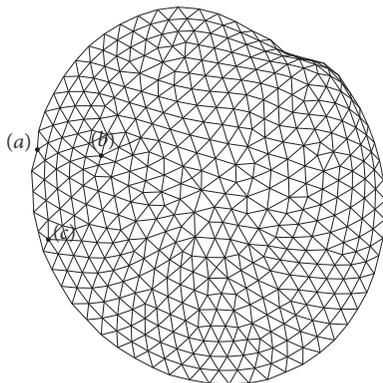


FIGURE 12: Example 7.

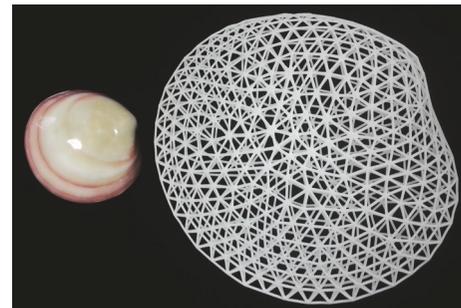


FIGURE 13: 3D printed model for Example 1.

challenges arising in some critical procedures of the scheme. The scheme is verified by typical examples. Physical models of the grid truss structures obtained from the examples are manufactured by 3D print, which finished a closed loop of physical → digital → physical.

It should be emphasized that the proposed scheme focuses on the grid generating problem in the design of a

free-from grid truss structure whose overall shape mimics a given physical model. Considering that the grid generated here is not for visualization or any other computer graphic purpose and will be used as structural members, this is a typical problem of civil engineering and is not common in other engineering areas. Correspondingly, the proposed scheme should not be supposed to be general-purpose, especially being compared to those proposed in the area of computer graphics. Hence, the applicability of the proposed

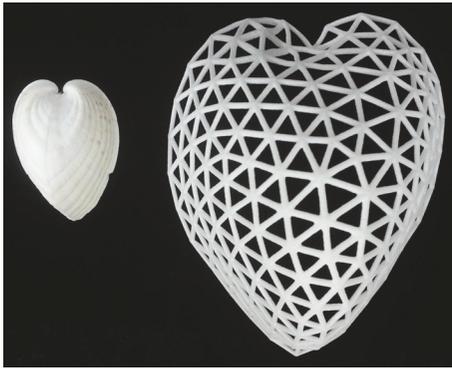


FIGURE 14: 3D printed model for Example 2.

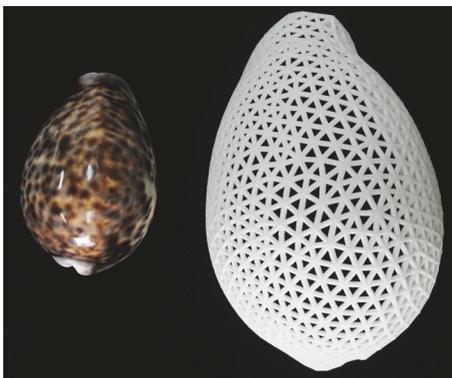


FIGURE 15: 3D printed model for Example 3.

scheme in other grid generating problems is not guaranteed and this could be deemed as a limitation of the work. Another limitation of it is that the grid generated is single-layer. It is well known that the applications of single-layer grid truss structures are limited to small to moderate span buildings due to their flexibility. However, the single-layer grid can be used as the upper layer or lower layer of a double-layer grid truss structure, and the other layer of the structure can be generated from the given layer through a simple geometric operation. In addition, quadrangular grids are also widely used in grid truss structures. Generating multilayer grid truss structures and quadrangular grid truss structures from point clouds will be the issues of interest in our future studies.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

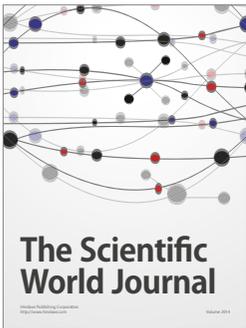
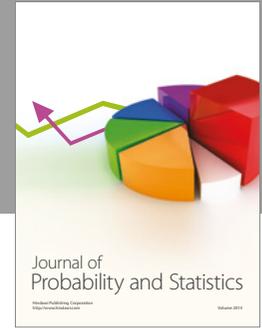
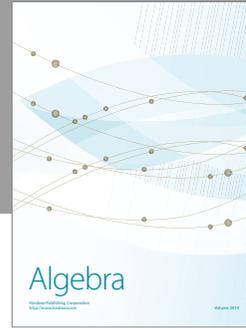
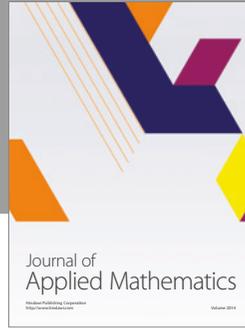
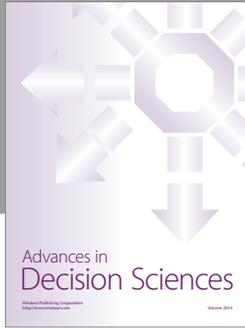
Acknowledgments

This work is supported by the National Key Research and Development Program of China (Grant no. 2016YFC0800200) and the National Natural Science Foundation of China (Grants nos. 51378458 and 51578494).

References

- [1] H. Isler, "Concrete shells derived from experimental shapes," *Structural Engineering International*, vol. 4, no. 3, pp. 142–147, 1994.
- [2] M. Schlaich, U. Burkhardt, L. Irisarri, and J. Goñi, "Palacio de comunicaciones-a single layer glass grid shell over the courtyard of the future town hall of Madrid," in *Proceedings of the Symposium of the International Association for Shell and Spatial Structures (IASS '09)*, Valencia, Spain, September 2009.
- [3] P. Szalabaj, *Contemporary Architecture and the Digital Design Process*, Butterworth Heinemann, New York, NY, USA, 2005.
- [4] N. Li and Y. Luo, "New modeling technique for bionic space grid structures," *International Journal of Advanced Steel Construction*, vol. 5, no. 1, pp. 1–13, 2009.
- [5] P. Ren, M. Zhou, G. Du, W. Shui, and P. Zhou, "3D scanning modeling method application in ancient city reconstruction," in *Proceedings of the 3rd International Conference on Optical and Photonic Engineering (icOPEN '15)*, SPIE, Singapore, April 2015.
- [6] P. Dorninger and N. Pfeifer, "A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds," *Sensors*, vol. 8, no. 11, pp. 7323–7343, 2008.
- [7] J. Lee, H. Son, C. Kim, and C. Kim, "Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data," *Automation in Construction*, vol. 35, pp. 199–207, 2013.
- [8] L. Zhu and J. Hyyppa, "The use of airborne and mobile laser scanning for modeling railway environments in 3D," *Remote Sensing*, vol. 6, no. 4, pp. 3075–3100, 2014.
- [9] A. Comsa, I. Maniu, and E.-C. Lovasz, "Triangulation 3D scanning method in biomedical reconstruction," *Solid State Phenomena*, vol. 166-167, pp. 145–148, 2010.
- [10] 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, no. 1, pp. 1–9, 2004.
- [11] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling for implicit surface," UCAM-CL-TR-562, University of Cambridge, Cambridge, UK, 2003.
- [12] C. Moenning, "Intrinsic point-based surface processing," UCAM-CL-TR-658, University of Cambridge, Cambridge, UK, 2006.
- [13] V. Liseikin, *Grid Generation Methods*, Scientific Computation, Springer-Verlag, Berlin, Germany, 1999.
- [14] O. C. Zienkiewicz and D. V. Phillips, "An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates," *International Journal for Numerical Methods in Engineering*, vol. 3, no. 4, pp. 519–528, 1971.
- [15] R. Löhner, "Progress in grid generation via the advancing front technique," *Engineering with Computers*, vol. 12, no. 3-4, pp. 186–210, 1996.
- [16] T. D. Blacker and M. B. Stephenson, "Paving: a new approach to automated quadrilateral mesh generation," *International Journal for Numerical Methods in Engineering*, vol. 32, no. 4, pp. 811–847, 1991.
- [17] C. M. Li and D. Lu, "Study of intelligent layout design of single-layer lattice shell of free form surface," *China Journal of Civil Engineering*, vol. 44, no. 3, pp. 1–7, 2011.
- [18] B. Curless, "From range scans to 3D models," *ACM SIGGRAPH Computer Graphics*, vol. 33, no. 4, pp. 38–41, 2000.

- [19] Wikipedia, 3D scanner; the Wikimedia Foundation, USA, 2016, https://en.wikipedia.org/wiki/3D_scanner.
- [20] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz, "Adaptive remeshing for compressible flow computations," *Journal of Computational Physics*, vol. 72, no. 2, pp. 449–466, 1987.
- [21] J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O. C. Zienkiewicz, "Finite element euler computations in three dimensions," *International Journal for Numerical Methods in Engineering*, vol. 26, no. 10, pp. 2135–2159, 1988.
- [22] S. H. Lo, "Volume discretization into tetrahedral-II. 3D triangulation by advancing front approach," *Computers & Structures*, vol. 39, no. 5, pp. 501–511, 1991.
- [23] R. Löhner, "Extensions and improvements of the advancing front grid generation technique," *Communications in Numerical Methods in Engineering*, vol. 12, no. 10, pp. 683–702, 1996.
- [24] R. Zhou, L. Zhang, X. Shu, and L. Zhou, "Research on surface reconstruction from dense scattered points," *Journal of Software*, vol. 12, no. 2, pp. 249–255, 2001.
- [25] J. Qian, Z. Chen, S. Zhang, and X. Ye, "The detection of boundary point of point cloud compression," *Journal of Image and Graphics*, vol. 10, no. 2, pp. 164–169, 2005.
- [26] J. Sarrate, J. Palau, and A. Huerta, "Numerical representation of the quality measures of triangles and triangular meshes," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 19, no. 7, pp. 551–561, 2003.
- [27] S. H. Lo, "Dynamic grid for mesh generation by the advancing front method," *Computers & Structures*, vol. 123, no. 1, pp. 15–27, 2013.
- [28] S. H. Lo, "A hidden-line algorithm using picture subdivision technique," *Computers and Structures*, vol. 28, no. 1, pp. 37–45, 1988.
- [29] R. Aubry, G. Houzeaux, and M. Vázquez, "A surface remeshing approach," *International Journal for Numerical Methods in Engineering*, vol. 85, no. 12, pp. 1475–1498, 2011.
- [30] G. Foucault, J.-C. Cuillière, V. François, J.-C. Léon, and R. Maranzana, "Generalizing the advancing front method to composite surfaces in the context of meshing constraints topology," *Computer-Aided Design*, vol. 45, no. 11, pp. 1408–1425, 2013.
- [31] D. T. Lee and A. K. Lin, "Generalized Delaunay triangulation for planar graphs," *Discrete & Computational Geometry*, vol. 1, no. 3, pp. 201–217, 1986.
- [32] K. Nakahashi and D. Sharov, "Direct surface triangulation using the advancing front method," in *Proceedings of the 12th Computational Fluid Dynamics Conference*, pp. 442–451, The American Institute of Aeronautics and Astronautics, San Diego, Calif, USA, June 1995.
- [33] W.-K. Jeong and C.-H. Kim, "Direct reconstruction of a displaced subdivision surface from unorganized points," *Graphical Models*, vol. 64, no. 2, pp. 78–93, 2002.
- [34] G. Taubin, "Curve and surface smoothing without shrinkage," in *Proceedings of the 5th International Conference on Computer Vision*, pp. 852–857, Cambridge, Mass, USA, June 1995.
- [35] J. Vollmer, R. Mencl, and H. Müller, "Improved laplacian smoothing of noisy surface meshes," *Computer Graphics Forum*, vol. 18, no. 3, pp. 131–138, 1999.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

