Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 13 Issue 5, May - 2026

Develop Personalized Stereoscopic Vision of Planar Lighting Products Based on Reverse Engineering Scanned Mesh Data

Hui-Chin Chang

HungKuo Delin University of Technology, Department of Creative Product Design, Taipei, Taiwan

chang.hcjang@gmail.com

***Abstract*—As the consumer market continues to refine, the influence of the niche has gradually escalated, personalized customized services have begun to have new business opportunities in the consumer market. Reverse engineering scanning is a way to obtain surface data of a physical model by scanning a measuring instrument from a known physical object or part. This article is mainly aimed at the mesh data measured by the low- and medium-priced reverse scanning equipment used by the personal studio. Through the rearrangement of the VTK data structure, efficiently complete the merging and deleting operations of degenerated meshes. Then, the number of triangle meshes is simplified by adjusting the critical value of the mesh side length and the mesh included angle, and the regular quadrangular conversion technique is used to convert the originally complicated triangle mesh into a regular distributed quadrangular mesh. Finally, extract the mesh edges, project them to the plane of the visual direction, and use them as the laser engraving pattern of acrylic material, then combined with the function of LED lighting, a personalized 3D vision flat lighting product can be completed. In this way, it can not only open up another business opportunity for small individual studio industry, but also promote the application of laser engraving machine to another artistic realm.**

***Keywords*—Reverse engineering, triangular mesh**

I. INTRODUCTION

Reverse engineering is a technology that uses known physical objects or parts to scan measurement data and reconstruct the CAD model of the physical object. In the early 1990s, reverse engineering technology began to attract great attention from the industry and academia. And with the development of computer software, hardware and measurement technology, reverse engineering has become a popular research focus in the CAD/CAM field and has become an engineering technology widely used in the industry. Therefore, reverse engineering uses specific measurement equipment and measurement methods to extract geometric coordinate point data on the surface of parts and physical objects, and then triangulates the point data to approximate the original surface of the physical object with a triangular mesh. Subsequent CAD models can be reconstructed using this triangular mesh model, and can be used with CAD software to carry out subsequent design changes or processing process planning, thereby improving the timeliness of product launch and reducing development costs.

As the technology of reverse vector acquisition equipment continues to improve, people can obtain more and more sophisticated 3D model surface data. In computer graphics, since triangles are the basic display unit directly supported by many graphics hardware, the triangle mesh model is still the main representation method of current 3D models. Complex models require higher requirements for computer processing speed, drawing speed and subsequent processing. However, in many cases, high-resolution models are not necessary, and we even have to deliberately reduce their mesh density. In other words, while keeping the geometry of the original model unchanged, we use as few triangular meshes as possible to approximate the original model to achieve our needs.

Among the existing triangular mesh model simplification methods, the geometric element deletion method is an important graphics simplification method. This method uses the geometric topology information of the original model to delete the impact on the geometric features of the model while maintaining a certain geometric error. Relatively small geometric primitives (points, edges, faces). There are mainly vertex deletion method and edge folding method.

The vertex deletion method is a simplification method that uses vertices as units and removes vertices from the model one by one according to certain criteria to reduce the mesh. The holes formed after removing the vertices are filled using the re-triangulation method. For example; Schroeder et al. [1] proposed to complete mesh simplification by vertex removal, which repeatedly selects a candidate vertex based on the characteristic degree of the geometric shape of the local area near the vertex, and this candidate vertex is the vertex to be removed next. Each vertex removal procedure includes two steps. The first step is vertex removal. This step will remove the vertex and the triangle adjacent to the vertex. After removal, the mesh model will leave a hole; the second step is triangulation, using the mesh triangulation algorithm to fill up the holes created by removing vertices. This method has the advantages of fast calculation and does not require too much memory. However, this algorithm is only suitable for polygons because it has certain difficulties in maintaining the smoothness of the surface. Ciampalini et al. [2] and Turk [3] compare the original omentum model and the simplified mesh model for error evaluation, so they can generate higher quality models, but they require more time and space.

The edge folding method is the most widely used among mesh model simplification methods. Its strategy is to repeatedly perform edge degradation, that is, to degenerate the edges of two vertices into one vertex. This simplification method can simplify at most one vertex and two triangles at a time. Representative algorithms in this regard include Hoppe [4] and Garland et al. [5]. In the method they proposed, a multi-level vertex tree structure of the mesh model can be easily established. By recording each simplification step to record the dependencies between vertices, a network organized by vertex information can be constructed. The resulting hierarchical structure is called a Vertex Binary Tree or a multi-level analytical structure of the mesh model. This vertex tree structure has been cited by many people for adaptive refinement of adaptive mesh models, for example, Hoppe [6], Xia et al. [7] and Luebke et al. [8]. The fine improvement of the adaptive mesh model means that only adjust the local area of the object is refined or roughened. It can also easily establish many different levels of original mesh models, and can quickly convert the mesh models at different levels. This method is often used in Real-Time Visualization systems and viewer-dependent multi-level model refinement.

In this article, the main purpose is to produce a flat lighting product with personalized three-dimensional vision. Therefore, in simplifying the number of triangular meshes, we only need to maintain its visual characteristics and do not focus on the error size generated after reduction. Therefore, we plan to use adjust the settings of the critical value of the mesh side length and mesh angle to repair, delete and simplify the number of triangular meshes to quickly achieve the goal.

In the field of computer graphics data applications, most meshes are mainly triangular meshes. However, in some applications, if they are presented as quadrilateral meshes, it will have better results, such as the analysis of flow field of fluid, or animated model textures...etc. Because many surface treatment operations of 3D models can be expressed in the form of solutions to partial differential equations, including feature-preserving surface smoothing, diffusion reaction mapping, texture synthesis, mesh editing, fluid dynamics, contour tracking... and so on. Regarding this field of research, one method is to first find the cutting path on the surface of the mesh model and then divide it into quadrangular blocks. Related articles include: Gu et al. [9] proposed to divide the mesh surface by corresponding to a square plane, a completely regular quadrangular mesh can be produced. And Sander et al. [10] further reduced mesh distortion through irregularly shaped cutting blocks.

Cutting on the model requires finding the cutting path, the way to find the cutting path can be through manual selection [11], or as proposed by Dong et al. [12] using Laplacian vector field combined with Morse theory to automatically find the cutting path. All these methods can divide the mesh surface into quadrangular blocks.

In this article, we plan to use regularized quadrilateral meshes conversion techniques to convert the original complex triangular mesh into regularly distributed quadrilateral meshes, and then rotate the overall meshes to a better visual orientation according to the visual attributes of the object, and then extract the boundary outline of the mesh and project it to the visual direction plane as the engraving texture for laser engraving.

II. RECODING AND PROCESSING OF TRIANGULAR MESHES

The STL (STereo Lithography) format file developed by the American 3D Systems Company in 1987 is currently a commonly used file format for recording triangular mesh models. It uses triangles on the model surface to exchange CAD data, and has now become one of the industry standards for CAD/CAM system interface file formats. STL format files are divided into binary and ASCII codes. They record the geometric data of all triangular meshes on the triangular mesh model, including the normal vector of the plane where each triangular mesh is located, and the coordinate values of the three vertices of the triangular mesh, and the data of the three edges of each triangular mesh. However, the STL format file only records the geometric information of all points, edges, and faces on any triangular mesh model, but does not include the topological relationships of them. Therefore, the STL format file records a large amount of repeated data. Each vertex is recorded almost 6 times, and each edge is recorded twice. Therefore, after reading the STL format file of the triangular mesh model, you need to first establish the topology structure of the model, that is, the adjacency and inclusion relationships of all points, edges, and faces on the triangular mesh model, and put the same geometric elements (vertices and edges) stored as a group element, this can facilitate the processing of points, edges, and surfaces on the model in subsequent programs, while saving the storage space of the model.

A. VTK data structure encoding of triangular mesh

As mentioned above, the file recording format of STL is based on a triangle as a unit, recording the unit normal vector and three vertex coordinate values of each mesh. However, there is no record of the topological relationship between meshes. For example, it is impossible to know the adjacent information of each mesh and the related information of repeated recording of vertex data. The mesh topology relationship is the basis of mesh processing, so the data structure stored in the mesh is very important. Therefore, this article plans to use the data structure of the VTK open source software system provided by Schroeder [13] and others as the basis, and then use this data structure as the basis for subsequent processing of the mesh.

VTK is an open source, freely available software system for 3D computer graphics, modeling, image processing, model rendering, scientific visualization, and 2D plotting. It supports multiple visualization algorithms and advanced modeling techniques, and utilizes distributed memory parallel processing respectively to improve speed and scalability. VTK is designed to be platform independent. This means it runs almost operating system, including Linux, Windows, and Mac; Its core functionality is written in C++ to maximize efficiency. Therefore, the data structure of its mesh can be shown in Figure 1.

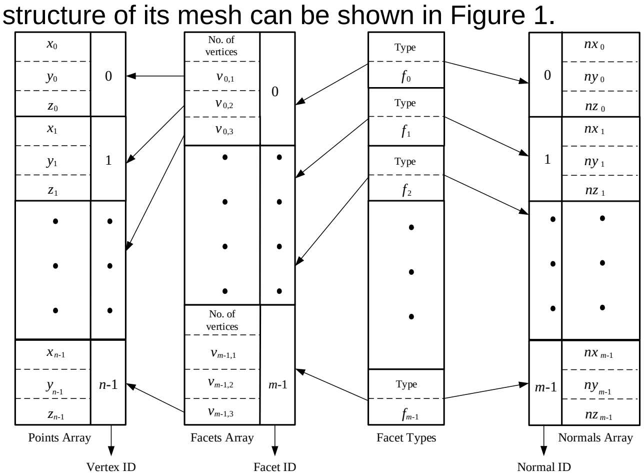

Figure 1 Data structure of VTK mesh [13]

VTK's data structure uses a dynamic array to store data, and dynamically changing the size of the array can effectively reduce memory usage. The value of the access array is represented by an integer index ID, and points to the linked data in the form of an index, effectively shortening the search time. The ID value points to the information stored in this array. The stored data contains four levels. The first layer is Facet Types, which records the type of each mesh and the Facet ID value corresponding to the Facets Array. In VTK, Facet Types may be polygons. However, in this article, we mainly use the mesh data structure to construct the model surface mesh data obtained by the inverse measurement equipment, so it is all triangles here. The second layer is Facets Array and Normals Array, where Facets Array records the number of vertices in each Facet and the Vertex ID values of the three vertices corresponding to the Points Array. The Normals Array records the normal value of each Facet, which contains the unit vector values in the x, y, and z directions. The third layer is Points Array, which records the three-dimensional coordinate values (xi, yi, zi) of all points. Recorded in this way, the coordinate value of each vertex will only be recorded once, which is similar to the original STL data format. In comparison, the disadvantage of repeated recording of vertex data is completely eliminated, so the memory capacity used is also significantly reduced. The following takes a 2D mesh as an example, as shown in Figure 2. There are 2 faces and 4 nodes in total, so the data structure of the mesh can be expressed as Table 1, Table 2 and Table 3.

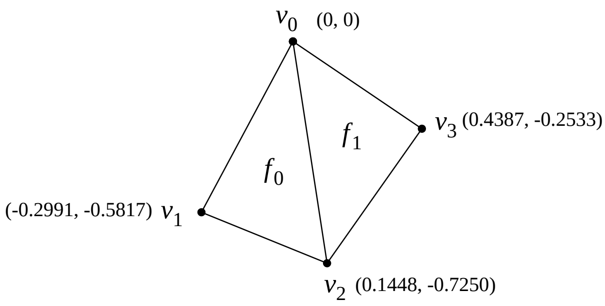

Figure 2 2D mesh data

Table 1 Data structure of 2D mesh - Vertex ID

Vertex number	X coordinate	Y coordinate
0	0	0
1	-0.2991	-0.5817
2	0.1448	-0.7250
3	0.4387	-0.2533

Table 2 Data structure of 2D mesh - Facet ID

Facet number	Node 1	Node 2	Node 3
f_0	0	1	2
f_1	0	2	3

Table 3 Data structure of 2D mesh -Normal ID

Facet number	x direction	y direction	z direction
f_0	0	0	1
f_1	0	0	1

B. Processing of defective meshes

When the three vertices of the mesh are at the same point, the three vertices of the mesh are collinear, or the mesh is too narrow and long, that is, when the three vertices of the mesh are not collinear, but an angle of the mesh is too small, or its opposite If the edge length of the edge mesh is too short, the mesh can be called a degenerate mesh, as shown in Figure 3. Because the degraded mesh lacks the information of the mesh normal vector in the entire mesh data, and many mesh processing processes require the assistance of this information, it is necessary to delete and rebuild the degraded mesh.

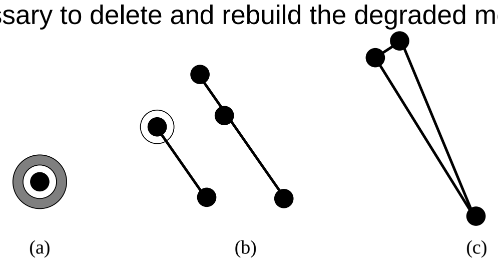

Figure 3 Degenerate mesh (a) three vertices are in common, (b) three vertices are collinear, (c) one side is too small

In this article, the deletion and reconstruction steps of the degraded mesh are as follows: load the complete model mesh M measured by the reverse engineering measurement equipment, and for each mesh M_i, record the coordinates $V_{i,1}$, $V_{i,2}$, $V_{i,3}$ of each mesh vertex. Calculate each included angle $\alpha_{i,1}$, $\alpha_{i,2}$, $\alpha_{i,3}$ of the mesh surface, and the corresponding side lengths $L_{i,1}$, $L_{i,2}$, $L_{i,3}$ of each included angle. Determine whether the angle value of $\alpha_{i,1}$, $\alpha_{i,2}$, $\alpha_{i,3}$ is "0^o" or "180^o". If true, it means that the mesh is co-point or collinear, then the index value "i" is stored in the desired Delete the mesh array A. If it is not true, continue to judge whether the angle value of $\alpha_{i,1}$, $\alpha_{i,2}$, $\alpha_{i,3}$ is between 0^o ~ 2^o, or the value of $L_{i,1}$, $L_{i,2}$, $L_{i,3}$ is less than 0.01. If it is true, then Indicates that the mesh is too long and narrow. Similarly, the index value "i" is stored in the mesh array A to be deleted. At the same time, the index values of the two vertices of the mesh edge that are too short are stored in the array B of the vertices to be merged. Finally, all meshes are judged in sequence. After all meshes are judged, the meshes recorded in array A and the vertices recorded in merge array B are finally deleted. The result of the vertex merging is shown in Figure 4.

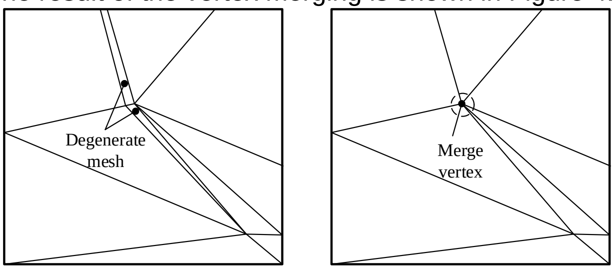

Figure 4 Degenerate mesh processing status

C. *Simplification of triangular mesh numbers*

As mentioned above, among the existing triangular mesh model simplification methods, the geometric element deletion method is an important graphics simplification method. This method based on the geometric topology information of the original model while maintaining a certain geometric error, to delete the elements that geometric features affect relatively small geometric primitives (points, edges, faces). In this article, we plan to use the method of dealing with degenerate meshes to simplify the triangular mesh by adjusting the critical value of the mesh side length for vertex merging. Taking Figure 5 as an example, there are 1020 existing meshes with a total of 556 vertices.

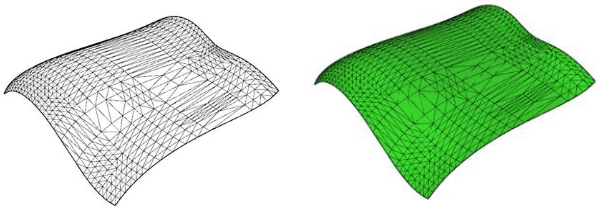

Figure 5 Original mesh

If the critical value of the mesh side length of the original mesh merged vertices is adjusted to "1.5", the number of meshes is simplified to 120 and 74 vertices, as shown in Figure 6.

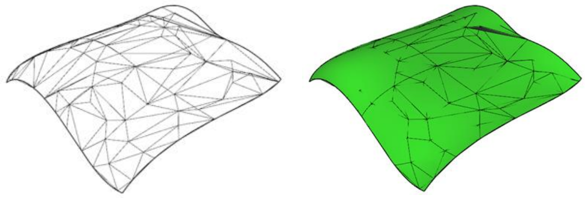

Figure 6 Simplified mesh with side length critical value adjusted to 1.5

III. CONVERSION OF QUADRANGULAR MESH

In the field of computer graphics data applications, most meshes are mainly triangular meshes. However, in some applications, if they are presented as quadrilateral meshes, it will have better results, such as the analysis of flow field of fluid, or animated model textures...etc. Because many surface treatment operations of 3D models can be expressed in the form of solutions to partial differential equations, including feature-preserving surface smoothing, diffusion reaction mapping, texture synthesis, mesh editing, fluid dynamics, contour tracking... and so on. Although traditional methods for solving partial differential equations rely on the surface features of irregular triangular meshes, in recent years, since Gu [10] provided techniques for converting 3D geometric information into 2D image information, many methods in image processing have efficient algorithms can be applied to the calculation of partial differential equations in 3D mesh models, allowing many complex or unsolved problems to find existing solutions in another space. Among such conversions, the 2D image data converted from the quadrangular 3D mesh through GIM is the easiest to apply.

In this article, the principle of quadrangular mesh processing is to use a surface that can completely cover the mesh, then set the required number of UV curves for the surface, and project these surfaces onto the mesh as the basis for the quadrangular meshing is then to adjust the vertices of the triangular mesh in each area to the intersection of the UV projection lines to complete the regular quadrangular mesh. The steps are as follows;

Step 1: Construct a guide line that represents the entire object, and a wrapped surface that can completely cover the mesh according to the direction of the guide line, as shown in Figure 7.

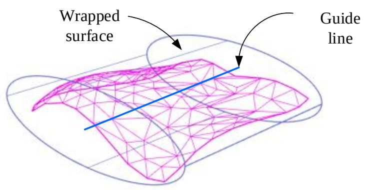

Figure 7 Wrapped surface that completely covers the mesh

Step 2: Based on the simplified mesh number, construct a matching UV curve for the wrapped surface, as shown in Figure 8.

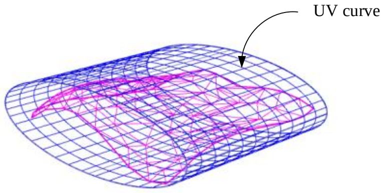

Figure 8 UV curve of wrapped surface

Step 3: Project the UV curve onto the triangular mesh, as shown in Figure 9.

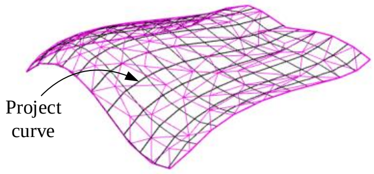

Figure 9 Projection curve on triangular mesh

Step 4: Adjust the vertices of the triangular mesh in each area to the intersection of the UV projection curves to complete the regular quadrangular mesh. The execution process is as follows.

a. Construct the data of each intersection point of the UV projection curve, such as n1, n2, n3, n4 in Figure 10.
b. Search for the vertices of the triangular mesh within or closest to the area, such as v1, v2, v3, v4 in Figure 10.

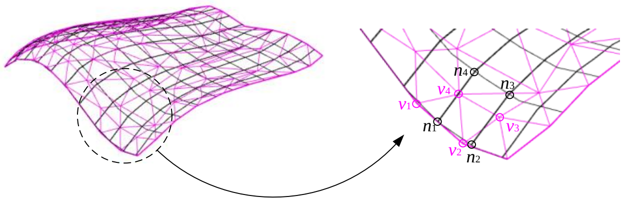

Figure 10 UV projection curve intersection and adjacent triangular mesh vertices

c. Replace the adjacent triangle mesh vertices with each intersection point of the UV projection curve, as shown in Figure 11. n1 replaces v1, n2 replaces v2, n3 replaces v3, and n4 replaces v4. The result is shown in Figure 11.

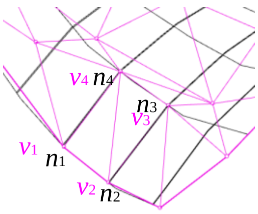

Figure 11 UV projection curve intersection replaces adjacent triangular mesh vertices

d. Merge the triangular mesh within this area to form a quadrangular mesh, as shown in Figure 12.

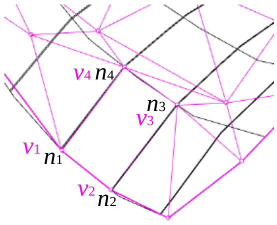

Figure 12 Triangular mesh merging results within the area

e. Repeat steps b ~ d to form a single quadrangular mesh one by one until the entire mesh change is completed, as shown in Figure 13.

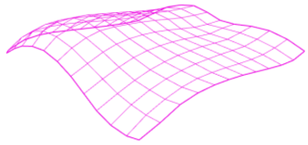

Figure 13 quadrangular meshing completed

IV. System Operation Steps

Step 1: Load the scanned model file, as shown in Figure 14. As can be seen from the picture, because it uses a low-to-medium scanning device, there are many defects.

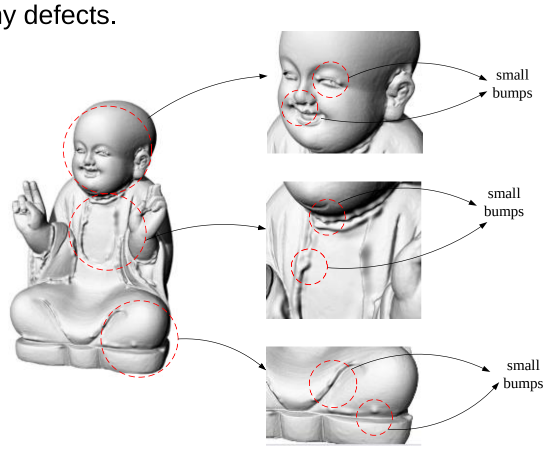

Figure 14 Scanned original data file

Step 2: Recode and complete the repair of defective meshes. The repair results and partial mesh distribution are shown in Figure 15. The current number of grids is 2,073,706.

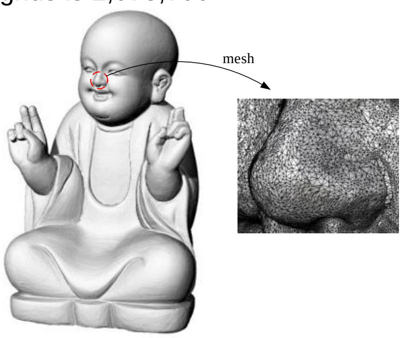

Figure 15 Repaired scanned data file

Step 3: Use the method of adjusting the critical value of the mesh side length to simplify the number of triangular meshes to approximately 20,000, as shown in Figure 16.

Figure 16 Simplified triangle mesh file

Step 4: Convert the triangular mesh into a quadrangular mesh, as shown in Figure 17.

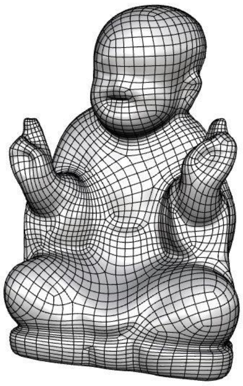

Figure 17 Converted quadrangular meshes file

Step 5: Project the edge lines of each mesh of the quadrangular mesh onto the 2D plane in the visual direction as the cutting pattern for subsequent laser processing, as shown in Figure 18.

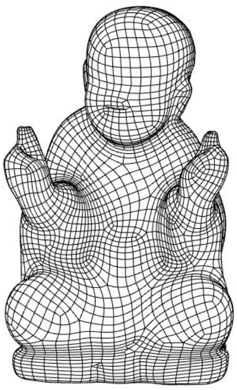

Figure 18 Projection file of the edge lines of the quadrangular meshes

Step 6: In order to increase the visual effect, manually add the cutting line of the outer edge offset to complete the final product, as shown in Figure 19.

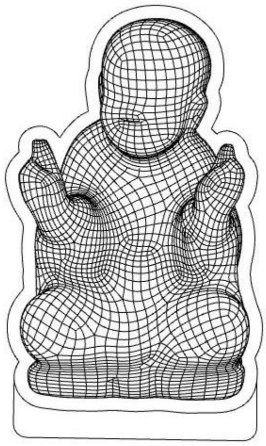

Figure 19: Laser cutting line file of the final product

Step 7: The computer simulation diagram of the finished product and the actual cutting finished product diagram, as shown in Figure 20.

(a) Computer simulation diagram (b) Acture product diagram

Figure 20 Computer simulation diagram and actual product diagram

V. CONCLUSION

This article is mainly aimed at the mesh data measured by the low- and medium-priced reverse scanning equipment used by the personal studio. Through the rearrangement of the VTK data structure, efficiently complete the merging and deleting operations of degenerated meshes. Then, the number of triangle meshes is simplified by adjusting the critical value of the mesh side length and the mesh included angle, and the regular quadrangular conversion technique is used to convert the originally complicated triangle mesh into a regular distributed quadrangular mesh. Finally, extract the mesh edges, project them to the plane of the visual direction, and use them as the laser engraving pattern of acrylic material. Finally, its feasibility was verified through practical laser processing. If combined with the function of LED lighting, a personalized 3D vision flat lighting product can be completed. In this way, it can not only open up another business opportunity for small individual studio industry, but also promote the application of laser engraving machine to another artistic realm. Figures 21 ~ 25 are the completed products in this article.

Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 13 Issue 5, May - 2026

(a) Computer simulation diagram (b) Acture product diagram

Figure 21 Finished product 1

(a) Computer simulation diagram (b) Acture product diagram

Figure 22 Finished product 2

(a) Computer simulation diagram (b) Acture product diagram

Figure 23 Finished product 3

(a) Computer simulation diagram (b) Acture product diagram

Figure 24 Finished product 4

(a) Computer simulation diagram (b) Acture product diagram

Figure 25 Finished product 5

ACKNOWLEDGMENT

This research is grateful to the Ministry of Science and Technology for encouraging technical schools and colleges to adopt practical project plans (MOST 113-2637-H-237-002) to receive funding support have enabled the successful completion of this research.

REFERENCES

1. Schroeder, W. J., Zarge, J. A. and Lorensen, W. E., “Decimation of triangle meshes. Proceeding of ACM”, SIGGRAPH 92, pp.65-70, 1992.
2. Ciampalini, A., Cignoni, P., Montani, C. and Scopigno, R., “Multiresolution Decimation based on Global Error.”, Tech. Report C96-021, CNUCE-C.N.R., Pisa, Italy, Jul.1996.
3. Turk, G., “Re-tiling polygonal surfaces”, Proceeding of SIGGRAPH’92, pp.55-64, 1992.
4. Hoppe, H., “Progressive meshes.”, In Proceedings of SIGGRAPH 96, pp. 99-108, 1996.
5. Garland, M. and Heckbert, P. S., “Surface simplification using quadric error metrics.”, In Proc. SIGGRAPH 97, pp. 209-216, 1997.
6. Garland, M. and Heckbert, P. S., “Simplifying Surfaces with Color and Texture using Quadric Error Metrics.”, In IEEE Visualization 1998, pp. 263-270, 1998.
7. Xia, J. C. and Varshney, A., “Dynamic view-dependent simplification for polygonal models.”, In Proc. 7th conference on Visualization, IEEE Computer Society Press, pp. 327–ff, 1997.
8. Luebke, D. and Erikson, C., “View-dependent simplification of arbitrary polygonal environments.”, Proceeding of ACM SIGGRAPH `97, pp. 199–208, 1997.
9. Gu, X., Gortler, S. J., and Hoppe, H., “Geometry Images.”, ACM Transactions on Graphics. v.21, n.3, 355–361. 2002.
10. Sander, P. V., Wood, Z. J., Gortler, S. J., Snyder, J., and Hoppe, H., “Multi-chart geometry images.”, Eurographics Symposium on Geometry Processing, 146–155. 2003.

11. Kraevoy, V., Sheffer, A., “Cross-parameterization and compatible remeshing of 3D models.”, ACM Transactions on Graphics. v.23 n.3, 2004.
12. Dong S., Bremer P.-T., Garland M., Pascucci, V., Hart J. C., “Spectral surface quadrangulation.”, ACM SIGGRAPH ’06, July 2006.
13. W. Schroeder, K. Martin and B. Lorensen, “The visualization toolkit : An object oriented approach to 3D graphic”, (3rd ed.), Kitware, 2003.