Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 58-71
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Auspicious Patterns and 3D Models Applied in the Creation of Personalized Products

Hui-Chin Chang

Department of Creative Product Design, Hungkuo Delin University of Technology, Taiwan
Email: chang.hcjang@gmail.com

How to cite this paper: Chang, H.-C. (2020) Auspicious Patterns and 3D Models Applied in the Creation of Personalized Products. *Journal of Computer and Communications*, **8**, 58-71.
https://doi.org/10.4236/jcc.2020.812006

Received: October 25, 2020
Accepted: December 14, 2020
Published: December 17, 2020

Abstract

Auspicious patterns are an important manifestation of traditional crafts aesthetics for Chinese culture, and it not only exhibits the clever tricks of folk art, showing more personality and characteristics of Chinese culture in the humanities and arts aesthetic concerns. It shows the traditional aesthetics, based on the harmonious and success, constructed by intelligence and humbleness, shaped by symmetry and balance. This thesis contains two topics: they are the 2D image materialization and the 3D model flattening. First is analyzing the image of the auspicious pattern, and transformed the 2D image into a solid model. The second is through the mathematical operation skills of the geometric model, the existing auspicious 3D model of the triangular mesh is scaled, appropriately rotated and divided to form a flattening model of different visual effects. Finally, these models by means of other modeling software were combined into a new 3D model, then through the 3D printer to quickly print out part of the unique personalized products, to promote the natural beauty of traditional Chinese culture.

Keywords

Auspicious Patterns, 2D Image Materialization, 3D Model Flattening

1. Introduction

The auspicious pattern is a form of literary and artistic expression in Chinese historical and cultural traditions, and it is closely related to people's lives, customs and cultural background. Its origin can be traced to the tribal totems of primitive society and the decorative patterns on some utensils at that time. People in primitive society are full of enthusiasm for life and survival, but they are also afraid of certain phenomena produced in nature, so there were sacrificial ceremonies and divination, these activities are mainly active to ask for good fortune. The production of auspicious patterns is directly related to the observation method of the ancient ancestors who viewed objects and images, as well as the recognition of patterns in the divination. Therefore, it is a kind of cultural heritage worth keeping.

Since the rapid prototyping technology was proposed in the late 1980s, it immediately received widespread attention and quickly became a research topic in the manufacturing field. With the continuous expansion of the application field of rapid prototyping technology, it has become an important part of advanced manufacturing technology. There is no doubt that the emergence of rapid prototyping technology represents another breakthrough in production engineering. The file format used in rapid prototyping systems is generally a triangular mesh data format file (STL file). Almost all developers of rapid prototyping systems have adopted this data format. In fact, all major CAD/CAM software systems have functional modules for generating STL files. The STL file format is a file format first proposed by 3D System in the United States. It is a model file obtained after triangular meshing of a 3D solid model file. It has become a data exchange standard between CAD systems and RPM system. This file format discretizes the CAD surface into a triangular mesh, and different precision will have different number of triangle mesh divisions. In the STL file, each triangle mesh has four data items, namely the coordinates of the three vertices of the triangle and the normal vector of the triangle mesh. The STL file is a collection of multiple triangle meshes. The STL file format was originally used in the Stereolithography (SLA) technology. Since STL has nothing to do with CAD systems, it quickly developed into a standard for data exchange between CAD systems and rapid prototyping machines in the rapid prototyping field. However, due to the STL format file has its own shortcomings, many scholars have proposed new data formats or amendments over the years. For example, the Delaunay triangulated mesh was proposed by the Russian mathematician Boris Delaunay in 1934, but it has the shortcomings that the boundary disappears and new nodes cannot be interpolated. In view of the shortcomings of the former, Chew proposed the Constrained Delaunay Triangulation (CDT) to improve and modify [1]. For the latter shortcoming, Edelsbrunner and Tan proposed a 2D interpolation node algorithm, which complements this function by inserting Steiner Points [2]. And Alper developed a new theory of Steiner Points insertion, called Off-Centers, by inserting this new type of Steiner Points, in addition to the optimization of the mesh quality and size, the problem of too many new Steiner Points is improved, at the same time, the number of inserted Steiner Points and the number of triangle meshes generated is reduced, thus reducing the time required for meshing [3].

It can be seen from the above that almost all scholars are focusing on researching the generation efficiency and quality of triangular meshes. The application of triangular meshes has also been widely used in 3D printing equipment. However, the deformation application of the triangle mesh of the object rarely appears in the existing literature.

Digital image processing generally refers to the use of computers to process images, the image refers to a 2D function $f(x, y)$, where x and y are spatial coordinates, and the size of f is called the intensity or grayscale of the image at that point. It can also be imagined that the image is divided into $m * n$ equal parts and each unit cell contains its specific brightness, position and other information. This unit cell is called an image element or pixel. Therefore, a grayscale digital image can also be regarded as a matrix, the row and column determine a point, and the corresponding matrix element is regarded as the gray level of the point. After the image is digitized, it can be analyzed and processed by image processing software, so that the image can be transformed into the information we need for subsequent applications [4].

Digital images contain three basic properties: resolution, number of layers and number of planes. The resolution refers to the number of pixels in an image, which is often expressed as the product of row and column pixels; the layer level is the discrete value of the light intensity distribution of the digital image, usually n-bits to represent the 2n degree value that each pixel can present, for example, an 8-bit digital image has a range of 28 = 0 - 255; the number of planes refers to how many pixel matrices are composed of a digital image. For example, the grayscale image is composed of a plane, while the color image is a digital image composed of three planes of red, blue and green [5].

Before the original image is processed, there must be a certain degree of noise or interference caused by the environment, and all external interference will destroy the quality of the image and cause a lot of trouble in analysis or processing. Therefore, in order to reduce the image noise, filtering and smoothing methods are used to eliminate the noise. Generally speaking, filtering and smoothing methods can be divided into frequency domain filtering and spatial domain filtering. The so-called frequency domain is to regard the original signal as composed of different orthogonal signals. The common method is to express by Fourier transform, and transform the original image to the frequency domain by discrete Fourier transform. Each point represents a basic frequency, and then, according to the method of frequency represents to do the processing. The spatial domain directly represents the characteristics of the image. It is a 2D function $f(x, y)$. The gray level of each point's coordinates (x, y) represents the physical characteristics of the image at this point. The characteristics of each point can also be modified with the surrounding gray value, and this research uses this method—Gaussian smoothing filter method to eliminate the noise in the image.

The purpose of edge enhancement is to mark the points with obvious brightness changes in the image, because the points with larger brightness changes can be regarded as the shape or outline of the feature, and because the points with large brightness changes mostly occur at the edge, it can also be called edge detection processing. After the image data is processed by edge detection, not only the amount of data can be greatly reduced, but also irrelevant information can be eliminated, and the important structural attributes of the image can be retained. The commonly used edge detection filter methods include Sobel operator, Laplace operator and Canny filter method [6]. In this study, the Sobel operator is used to enhance the edges in the image.

As for the application of processed images, most of them are currently used in the field of object shape or face recognition. The application of materialization it is rarely seen. Therefore, this research is based on the existing traditional auspicious image model, and reconstructs the characteristic contour appearance through image processing techniques. Then use 3D modeling software (Rhinoceros) to create a 3D physical model, and quickly print out personalized and unique products with a 3D printer to promote the inherent cultural tradition.

2. Techniques for Turning 2d Image into Materialization

In the process of turning 2D images into 3D solid model, the first step is to appropriately process the image. In this thesis, the preliminary image processing part focuses on the noise filtering and edge enhancement of the digital image. The method is explained as follows:

2.1. Filtering and Smoothing of Digital Images

This thesis uses Gaussian smoothing filter to smooth digital images. Gaussian smoothing filter method is usually called low-frequency filtering. It is mainly used to eliminate the high-frequency changes in the image, strengthen the low-frequency part, and make the changes in the image more uniform, it can produce a fuzzy image and reduce the sharper image part. It is usually used to reduce image noise and reduce the level of detail. The Gaussian smoothing mask is shown in Table 1.

From a mathematical point of view, the Gaussian blur process of an image is the convolution product of the image and the normal distribution. Since the normal distribution is also called the Gaussian distribution, this process is also called Gaussian blur. The value of each pixel is the weighted average of surrounding neighboring pixels and the value of the original pixel has the largest Gaussian distribution value, so it has the largest weight value. As the neighboring pixels are farther from the original pixel, the weight value is smaller. Figure 1 is the comparison diagram before and after image filtering.

2.2. Edge Enhancement of Digital Images

The meaning of the magnitude of the image gradient is the speed at which the brightness of the image changes. For the edge of the image, where the grayscale value changes greatly, the gradient value is also greater. Conversely, for the smoother part of the image, the grayscale value changes less, and the corresponding

Table 1. Gaussian smoothing mask.

1	2	1
2	4	2
1	3	1

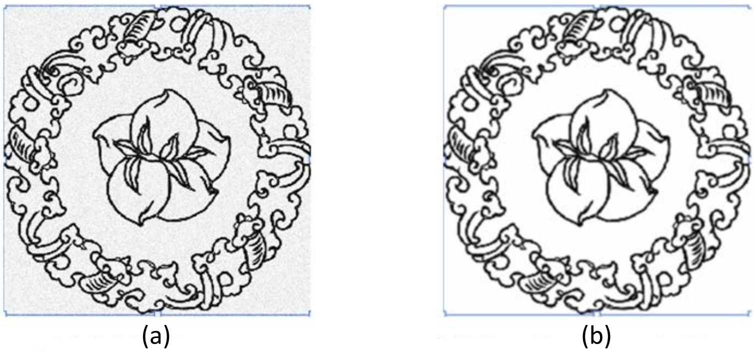


Figure 1. The comparison diagram before and after image filtering. (a) Original image; (b) Filtered and smoothed image.

gradient value is also smaller. Therefore, the calculation of the image gradient is the collection of information on the edge of the image. For images, the most common method to calculate image gradient values is to use Sobel Filter. This operation includes two operators, which are the horizontal and vertical convolution of the image *f*(*x*, *y*) to calculate the vertical gradient and the horizontal gradient, and then the strength and direction of the gradient can be obtained. The operation is as follows:

$$g_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, g_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

$$\begin{aligned} G_x &= g_x \times f(x,y) \\ G_y &= g_y \times f(x,y) \end{aligned} \quad (2)$$

Total gradient strength: $M(x,y) = \sqrt{G_x^2 + G_y^2}$;

Gradient direction: $\alpha(x,y) = \tan^{-1}(G_y/G_x)$.

Figure 2 is the comparison diagram before and after image edge enhancement.

3. Techniques for 3d Model Flattening

This thesis is based on the mathematical operation skills of the geometric model, and the existing auspicious 3D model of the triangular mesh is scaled, appropriately rotated and divided to form another model of different visual effects. Therefore, the key points of this part are as follows.

3.1. Data Acquisition of Auspicious 3D Model of Triangular Mesh

At present, all 3D printers can accept STL format, which is a triangular mesh format. The so-called STL format is an approximate triangular mesh to represent the surface or solid entities of the object. In the STL file, each triangle mesh has four data items, namely the coordinates of the three vertices of the triangle and the normal vector of the triangle mesh. The STL file is a collection of multiple triangle meshes. The output form of STL can be divided into ASCII and Binary.

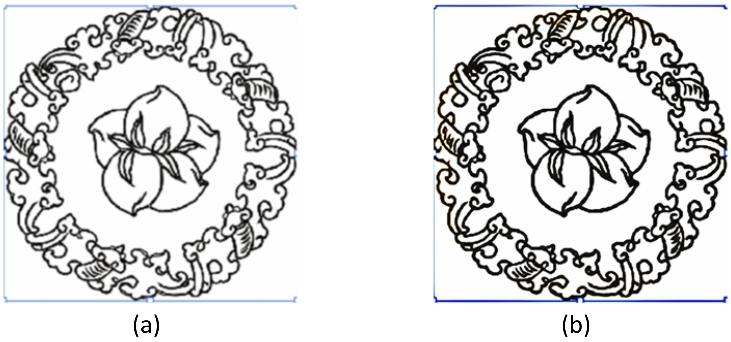


Figure 2. The comparison diagram before and after image edge enhancement. (a) Before image edge enhancement; (b) After image edge enhancement.

The ASCII STL file is just for the user to see its format and further edit, but it occupies a large space, the Binary format is a more reasonable choice. The ASCII STL file format is as follows:

```
solid name
facet normal ninjnk
  outer loop
    vertex v1x v1y v1z
    vertex v2x v2y v2z
    vertex v3x v3y v3z
  endloop
endfacet
endsolid name
```

Among them,

solid name: stands for file header.

endsolid name: stands for the end of the file.

ni, nj, nk: the normal vector of the surface.

outer loop: stands for outer loop.

vertex v1x, v1y, v1z: the coordinates of vertex 1.

vertex v2x, v2y, v2z: the coordinates of vertex 2.

vertex v3x, v3y, v3z: the coordinates of vertex 3.

Figure 3 is the auspicious 3D model of triangular mesh.

3.2. 3D Model Scaling Processing

In this study, we use a column matrix A, to represent the position vector of each end point in the triangular meshed of 3D model. Therefore, the transformation of the scale or deformation of this model in space can be expressed as $AT = B$, Where T can be regarded as a geometric operator, Therefore, the meaning of the above formula is that a set of points represented by the position vector in the matrix A is geometrically transformed to obtain a new form. In order to solve the problem that the 3D model can be converted based on any position in the 3D space, it must be expressed in homogeneous coordinates. Therefore, the point $[x\ y\ z]$ in the three-dimensional space can be determined by the four-dimensional

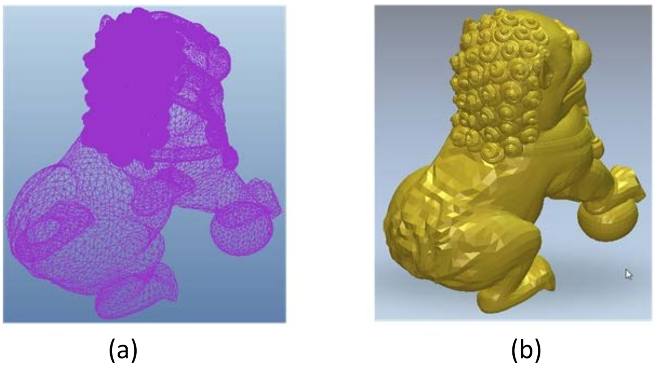


Figure 3. The auspicious 3D model of triangular mesh. (a) Wire-frame style; (b) Rendering style.

position vector $\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x' & y' & z' & 1 \end{bmatrix}\begin{bmatrix} T \end{bmatrix}$, where $[T]$ is some transformation matrix. Therefore, the generalized 4 × 4 transformation matrix for three-dimensional homogeneous coordinates is:

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix} \tag{3}$$

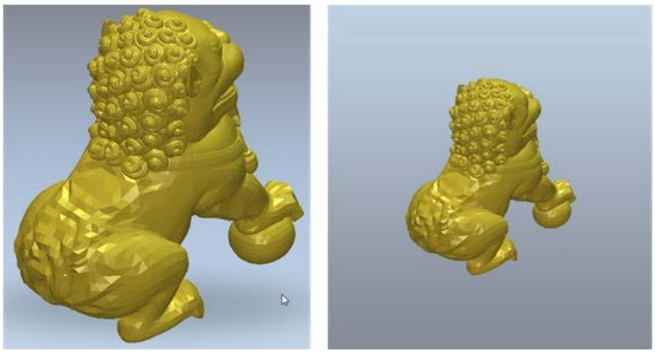
In the above formula, the upper left 3 × 3 submatrix $\begin{bmatrix} a & b & c \\ d & e & f \\ g & i & j \end{bmatrix}$ produces a linear transformation in the form of scaling, shearing, reflection and rotation. The 1 × 3 lower left submatrix $\begin{bmatrix} l & m & n \end{bmatrix}$ produces translation, and the upper right 3 × 1 submatrix $\begin{bmatrix} p \\ q \\ r \end{bmatrix}$ produces a perspective transformation. The final lower right-hand 1 × 1 submatrix $[s]$ produces overall scaling. Therefore, when the 3D model is to be scaled up and down in equal proportions, the transformation matrix $[T]$ is:

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \tag{4}$$

Figure 4 is the comparison diagram of the 3D model before and after it is reduced by 0.5 times.

3.3. 3D Model Display Orientation Control Processing

When the model is rotated multiple times in space, that is, the model will perform translation and rotation of each axis relative to the center of rotation. Therefore, the coordinates of each point of the model will change accordingly. Assuming that if an arbitrary axis passing through the point (x_0, y_0, z_0) in space,



(a) (b)

Figure 4. The comparison diagram of the 3D model before and after reduced by 0.5 times. (a) Original size; (b) 0.5 times smaller.

its direction cosine is (c_x, c_y, c_z), the rotation angle δ of this axis can be completed by the following steps:

Step 1: Translate so that the point (x_0, y_0, z_0) is at the origin of the coordinate system.

Step 2: Perform proper rotation to make the rotation axis coincides with the *Z* axis.

Step 3: Rotate about the *Z* axis by angle δ.

Step 4: Perform the inverse matrix operation of the combined rotation transformation.

Step 5: Perform the inverse matrix operation of translation.

Generally speaking, in order to make any axis passing through the origin coincide with any axis, it is necessary to rotate the other two coordinate axes twice. To make any rotation axis coincide with the *Z* axis, first rotate the *X* axis, and then rotate the *Y* axis. To determine the rotation angle α to the *X* axis (where α is used to place any axis on the *xy* plane), must project the unit vector along the coordinate axis on the *yz* plane, and the *y* and *z* components of the projection vector are c_y and c_z, respectively, they are the direction cosine of the unit vector along any axis. So we can get;

$$d = \sqrt{c_y^2 + c_x^2}\ ;\ \cos\alpha = \frac{c_z}{d};\ \sin\alpha = \frac{c_y}{d} \tag{5}$$

After rotating the *X* axis to the *xz* plane, the *z* component of the unit vector is *d*, and the *x* component is the direction cosine c_x in the *x* direction. Because the length of the unit vector is 1, the rotation angle β used to make any axis coincide with the *Z* axis is;

$$\cos\beta = d\ ;\ \sin\beta = c_x \tag{6}$$

So the complete conversion is;

$$[M] = [T][R_x][R_y][R_\delta][R_y]^{-1}[R_x]^{-1}[T]^{-1} \tag{7}$$

where the required translation matrix is;

$$[T]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\-x_0&-y_0&-z_0&1\end{bmatrix}\tag{8}$$

The transformation matrix for rotation about the X axis

$$[R_x]=\begin{bmatrix}1&0&0&0\\0&\cos\alpha&\sin\alpha&0\\0&-\sin\alpha&\cos\alpha&0\\0&0&0&1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&c_z/d&c_y/d&0\\0&-c_y/d&c_z/d&0\\0&0&0&1\end{bmatrix}\tag{9}$$

And about the Y axis

$$[R_y]=\begin{bmatrix}\cos(-\beta)&0&-\sin(-\beta)&0\\0&1&0&0\\\sin(-\beta)&0&\cos(-\beta)&0\\0&0&0&1\end{bmatrix}=\begin{bmatrix}d&0&c_x&0\\0&1&0&0\\-c_x&0&d&0\\0&0&0&1\end{bmatrix}\tag{10}$$

Finally, the rotation angle of arbitrary axis can be obtained by the Z axis rotation matrix, and its value is;

$$[R_\delta]=\begin{bmatrix}\cos\delta&\sin\delta&0&0\\-\sin\delta&\cos\delta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\tag{11}$$

Although the terms of the rotation matrix $[R_x]$ and $[R_y]$ in Equation (6) can be obtained from Equation (8) and Equation (9). But in fact, the angles α and β are difficult to calculate directly. Therefore, if the direction cosine of arbitrary axis cannot be known, but it is known that the axis passes through the second point (x_1, y_1, z_1), after normalizing the vector from the first point to the second point, the direction cosine of the axis can be obtained as follows :

$$\begin{bmatrix}c_x & c_y & c_z\end{bmatrix}=\frac{\left[(x_1-x_0)\quad(y_1-y_0)\quad(z_1-z_0)\right]}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2}}\tag{12}$$

In this way, we can get the orientation of the model after rotating arbitrary axis in space. Figure 5 shows the original model rotated around the Z axis by 110 degrees counterclockwise.

3.4. 3D Model Flattening Processing

When the 3D model is rotated to an appropriate angle, then the flatting operation can be performed. The main principle is to use a plane parallel to the visual direction as the cutting plane and find the unit normal direction of the cutting plane, then adjust the depth of this cutting plane appropriately to cut the 3D model, and set the height of the 3D model after flatting. Finally, the front model after cutting is projected along the normal direction of the cutting plane and scaled to complete the flatting operation.

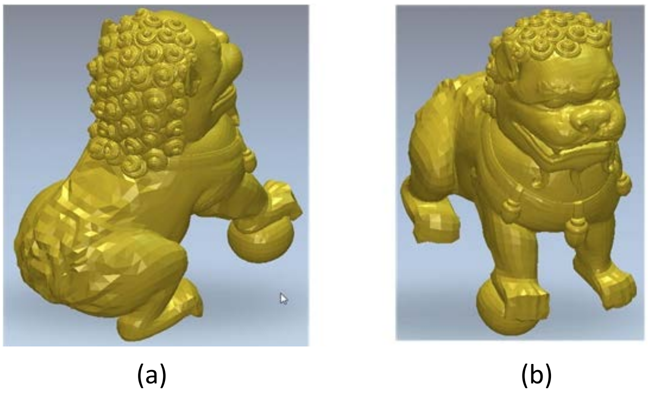


Figure 5. The original model rotated around the Z axis by 110 degrees counterclockwise. (a) Original orientation; (b) Orientation after rotation.

Generally used to represent the coordinate system of a 3D model, the plane parallel to the ground is often set as the XY plane. According to the right-hand rule, the plane parallel to the screen is the XZ plane. Assuming that the position coordinates of the vertices of the triangle meshes of the front model after cutting are $\left(X_{1x}, Y_{1y}, Z_{1z}\right)$, $\left(X_{2x}, Y_{2y}, Z_{2z}\right)$, $\left(X_{3x}, Y_{3y}, Z_{3z}\right)$, $\cdots$, $\left(X_{nx}, Y_{ny}, Z_{nz}\right)$, and the visual direction is perpendicular to the screen (XZ plane), then the normal direction of the cutting plane is the Y-axis. It is also assumed that the maximum and minimum values of the cosine in the Y-axis direction of the vertex coordinates of each triangle mesh are Y_{max} and Y_{min} respectively, the height of the 3D model after flatting is h, then the transformation matrix $[T]$ projected on the cutting plane is:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

After this projection operation, the vertex coordinates of each triangle mesh become $\left(X'_{1x}, 0, Z'_{1z}\right)$, $\left(X'_{2x}, 0, Z'_{2z}\right)$, $\left(X'_{3x}, 0, Z'_{3z}\right)$, $\cdots$, $\left(X'_{nx}, 0, Z'_{nz}\right)$. Then the vertex coordinates of each triangle mesh are translated in equal proportion along the Y axis, so the vertex coordinates of each triangle mesh become $\left(X'_{1x}, Y'_{1y} * h/\left(Y_{\max} - Y_{\min}\right), Z'_{1z}\right)$, $\left(X'_{2x}, Y'_{2y} * h/\left(Y_{\max} - Y_{\min}\right), Z'_{2z}\right)$, $\left(X'_{3x}, Y'_{3y} * h/\left(Y_{\max} - Y_{\min}\right), Z'_{3z}\right)$, $\cdots$, $\left(X'_{nx}, Y'_{ny} * h/\left(Y_{\max} - Y_{\min}\right), Z'_{nz}\right)$. This completes the flattening operation, as Figure 6 shown.

4. Production Process and Results

In this thesis, the production of personalized products is divided into (1) 2D image materialization. And (2) 3D model flattening. The two methods are as follows:

4.1. 2D Image Materialization Processing and Results

Step 1: Choose an auspicious image.

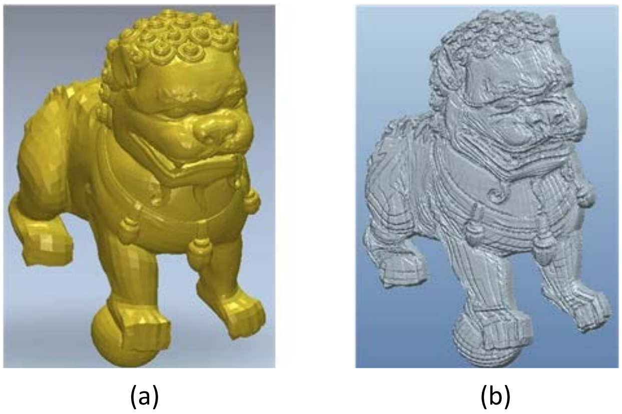


Figure 6. The original model completes the flattening operation. (a) Original style; (b) flattening style.

Step 2: Load the image file, and create the contour path of the image through image processing techniques.

Step 3: Load this contour path into the 3D modeling software.

Step 4: Through the node adjustment in the 3D modeling software, complete the complete contour path of the image.

Step 5: According to the characteristics of this auspicious image, construct 3D models of different heights from the closed contour paths of each area, to complete the 2D image materialization processing.

The entire flow chart is shown in Figure 7, and the Figure 8 is the auspicious products of 2D image materialization results.

4.2. 3D Model Flattening Processing

Step 1: Load the STL file of the auspicious 3D model.

Step 2: Adjust the scale of 3D model appropriately.

Step 3: Adjust the display orientation of 3D model appropriately.

Step 4: Using the display orientation of 3D model as the depth normal direction, and flattening the STL data of the auspicious 3D model according to the assigned depth.

Step 5: Use 3D modeling software to make matching pendants and combine them into models of unique products.

The entire flow chart is shown in Figure 9, and Figure 10 is the auspicious products of 3D model flattening results.

5. Conclusion

Since ancient times, people have been in awe of nature, and they are full of longing and hope in the face of an unpredictable future. The auspicious pattern is also a presentation of people's hope for the future. The auspicious patterns are suitable for the past and the present. They can be viewed and used, which are inseparable from people. In this thesis, the 3D model was created from two methods of 2D image materialization and the 3D model flattening, and then these

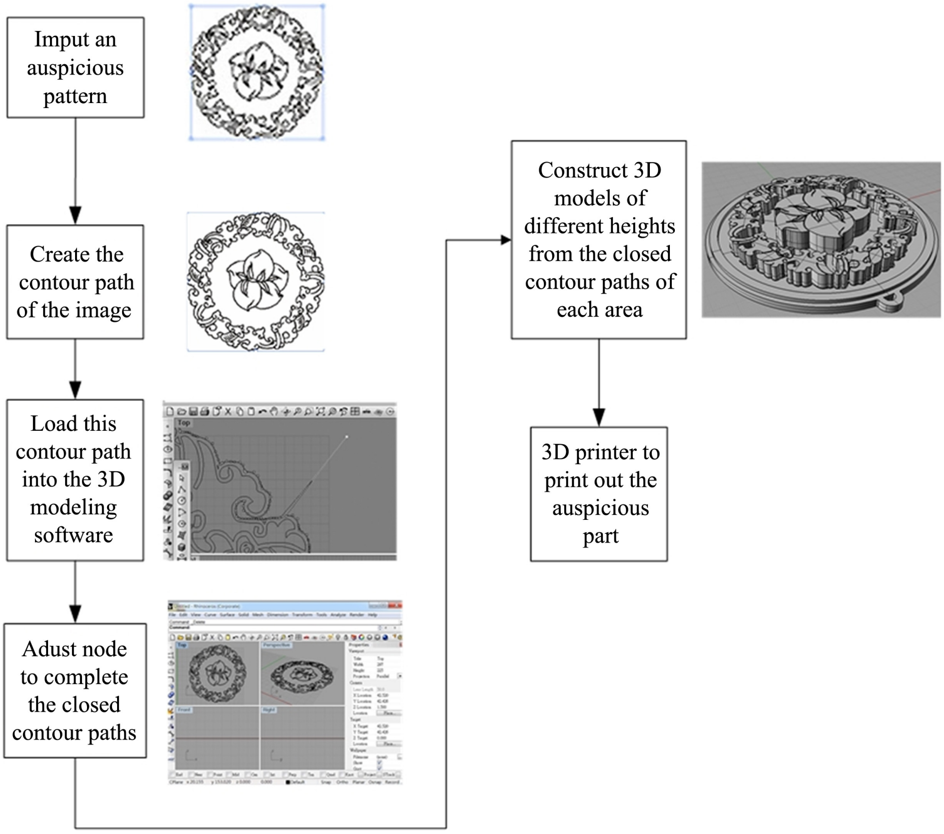

Imput an auspicious pattern
Create the contour path of the image
Load this contour path into the 3D modeling software
Adust node to complete the closed contour paths
Construct 3D models of different heights from the closed contour paths of each area
3D printer to print out the auspicious part

Figure 7.Theentire flow chart of 2D image materialization.

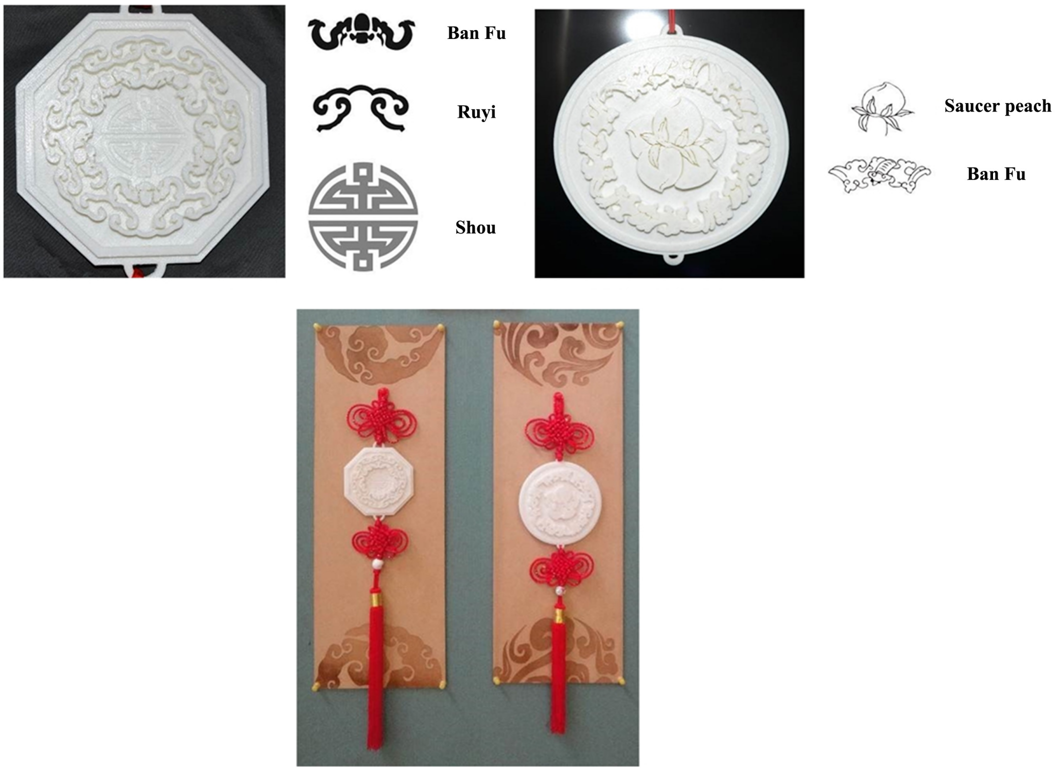


Figure 8. The auspicious products of 2D image materialization results.

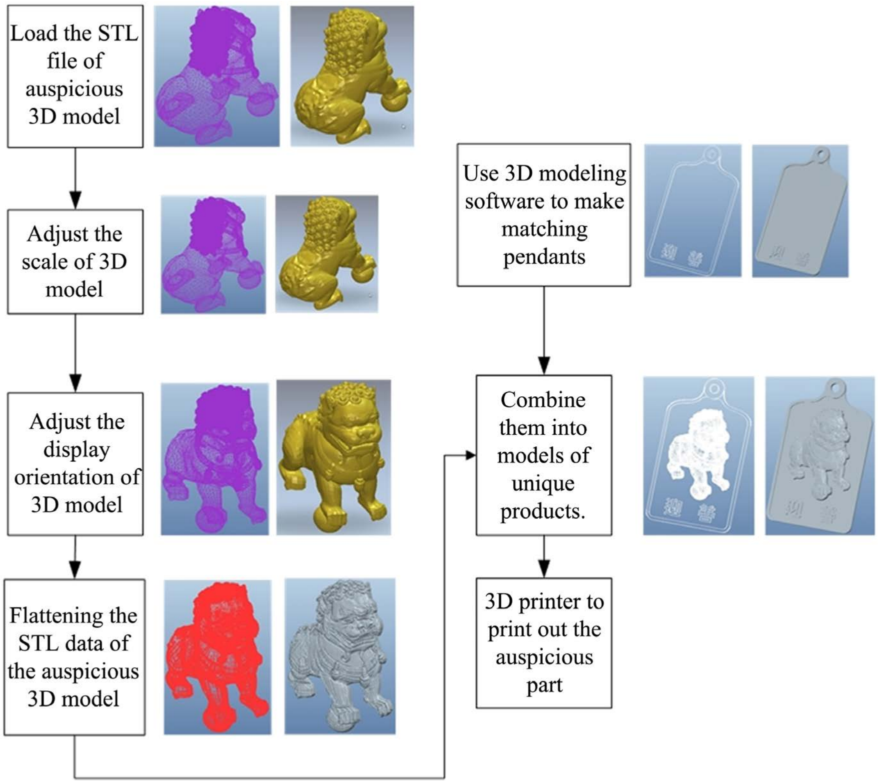

Load the STL file of auspicious 3D model
Adjust the scale of 3D model
Adjust the display orientation of 3D model
Flattening the STL data of the auspicious 3D model
Use 3D modeling software to make matching pendants
Combine them into models of unique products.
3D printer to print out the auspicious part

Figure 9. The entire flow chart of 3D model flattening.

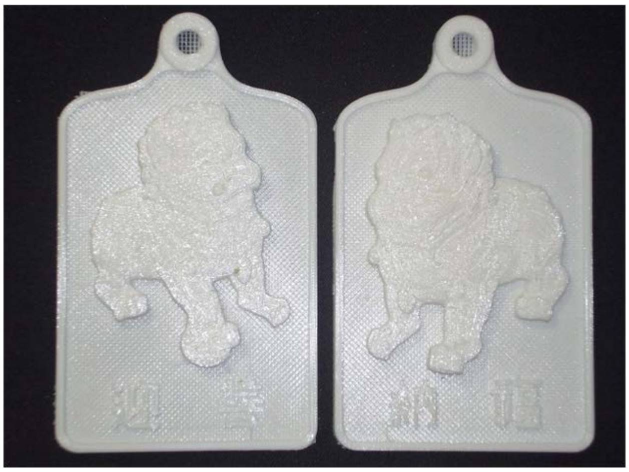


Figure 10. The auspicious products of 3D model flattening results.

models by means of other modeling software were combined into a new 3D model, through the 3D printer to quickly print out part of the unique and auspicious products to promote the inherent cultural tradition.

Acknowledgements

This research is grateful to the Ministry of Science and Technology for the funding support of the special research project for college students (104-2815-C-237-001-H), which enabled the successful completion of this research.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Chew, L.P. (1989) Constrained Delaunay Triangulations. *Algorithmica*, **4**, 97-108. https://doi.org/10.1007/BF01553881

[2] Edelsbrunner, H. and Tan, T.S. (1993) An Upper Bound for Conforming Delaunay Triangulations. *Discrete & Computational Geometry*, **10**, 197-213. https://doi.org/10.1007/BF02573974

[3] Üngör, A. (2004) Off-Centers: A New Type of Steiner Points for Computing Size-Optimal Quality-Guaranteed Delaunay Triangulations. In: Farach-Colton, M., Ed., *LATIN* 2004: *Theoretical Informatics*, Springer, Berlin, 152-161. https://doi.org/10.1007/978-3-540-24698-5_19

[4] Masters, B.R., Gonzalez, R.C., *et al.* (2009) Book Review: Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, **14**, Article ID: 029901. https://doi.org/10.1117/1.3115362

[5] Lin, J.J. (2009) Research on Application of Image Processing in Online Cell Technology and Monitoring System. Master Thesis, National Taiwan University of Science and Technology, Taipei.

[6] Cai, K.Z. (2008) The Designed and Implemented Real-Time Low-Noise Edge-Preserving Bilateral Filters. Master Thesis, Tsinghua University, Beijing.