

Research On Cylindrical Planar Image Machining System

Hui-Chin Chang

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

chang.hcjang@gmail.com

Abstract—Auspicious images are an important form of expression of traditional craft aesthetics among various ethnic groups., but these images retain in 2D image all at present. Therefore, the purpose of this article is to three-dimensionalize or convert these retained 2D plane images into cylinders to make them more personalized and interesting. This paper first converts the 2D plane color image (*.BMP) into a grayscale image through image processing technology, then use the feature points determined by its grayscale attributes relative to the cutting depth, project and superimpose them on the conic surface with 3D features, and construct a B-Spline surface with relief characteristics. Cooperate the limiting conditions of the chord length and cutter interval, and the inverse offset method comes to get appropriate CL points of cutting paths of 3D relief part, and then cylindrical the CL points of cutting paths of them.

Keywords— *Cylindrical planar image*

I. INTRODUCTION

Auspicious patterns are a form of literary and artistic expression in our country's historical and cultural traditions, which are closely related to people's lives, customs and cultural backgrounds. Its origin can be traced back to tribal totems in primitive society and decorative patterns on some utensils at that time. People living in primitive society are full of enthusiasm for life and survival, but they are also afraid of some phenomena produced in nature, so there are sacrifices and divination. These activities are mainly activities of divination, and the generation of auspicious patterns is directly related to the ancient ancestors' way of observing objects and images, and the understanding of observing patterns in divination. Therefore, the purpose of this paper is to three-dimensionalize these retained 2D plane images, and then use NC machines to carve out these features to make them more personalized and interesting, so as to promote and maintain the inherent traditional aesthetics.

Generally, NC machining can be roughly divided into rough machining, intermediate machining and finishing machining. Among them, rough machining mostly uses contour machining, and uses linear and contour tool paths to quickly remove workpiece blanks and form a prototype of the finished product, thus producing Step-shaped remaining material, after

machining in intermediate processing, most of the remaining material to be cut in finishing machining is evenly applied to the model.

As for the way of processing from point data, it can be roughly divided into two categories. One is to reconstruct the surface from point data, and then generate the tool path. The second is not to rebuild the surface, but to generate the tool path directly from the point data [1]. The number of point data must be considered when processing through point data. The tool path generated by point data through the creation of a surface is suitable for the situation with a small number of points. If the number of points is large, the method of creating a surface through points is still used, the speed will be slowed down or even crashed due to the excessive memory occupied by the surface. When there are many point data, it is more suitable to directly generate the tool path through points, but the distance between points is not small enough to obtain polygonal machining results. This range is worthy of further discussion.

In terms of offset curves, Piegl [2] proposed that the offset curves should first be distinguished whether they are straight lines, arcs, or circles. Those belonging to this category should be offset in a precise way, and those belonging to free curves should be approximated, set the sample point on the free curve, and use the sample point to make a sample curve. However, in the surface offset method in Piegl's paper, it does not explain the processing method of the curve or the self-intersection part of the surface. In terms of tool path detection, the Z-Map [3] method is to set a grid point on the XY plane, and use the grid point to obtain the Z coordinate of the grid point according to the 2D_Jacobian rule. If the surface is steep, use Edge-extended Z-map, and interpolate appropriate points there to avoid distortion of the surface. Then use the inverse offset method to find the tool center method.

There are three commonly used tool path planning methods: 1. Parametric method: the tool path is mapped from the parameter space to the workpiece surface; 2. Drive surface method: the tool path curve is obtained by intersecting a series of parallel surfaces with the workpiece surface. 3. Guide plane method: The 2D path on the guide plane is projected onto the workpiece surface [4]. This study adopts the concept of Choi [4] parameter method, sets the path spacing parameter as a constant, and uses the method of chord height to determine the step length.

The method of calculating the path spacing Luis [5] takes the Cusp-height as an important item in the calculation, and this method can obtain a relatively uniform surface roughness. Calculate the step length by the method of chord height, this method can get fewer points in flatter places and more points in places with larger curvatures. The tool interference detection adopts the inverse offset method [6] to obtain the CL surface, but it does not clearly show the actual processing method of the area near the tool. Lin [7] set the interval of u/v value on the curved surface to obtain interference detection points. You [8] used the trial-and-error method for anti-collision detection for the problems of overcut detection and collision in interference detection, and corrected the axis of the tool axis with the direction of the vector sum to avoid the interference of the tool axis.

Therefore, this paper converts the 2D plane color image into a gray-scale image, and then uses the feature points determined by its gray-scale attribute relative to the cutting depth to project and superimpose it on a conic surface with three-dimensional features, and then constructs a B-Spline surface with relief characteristics.

II. IMAGE PROCESSING RELATED TECHNOLOGY

Image processing is simply to digitize images, so that people or computers can obtain more and more useful information from the processed digital images, and make more reliable follow-up judgments, analysis, and applications. Generally speaking, the method of digital image processing can be roughly divided into three processes: front-end, middle-stage and post-processing [9].

A. Classification of BMP Image File Format

In this article, the BMP format is used as the benchmark for the original image. The so-called BMP file is the abbreviation of BitMap, which is a storage format specially designed for drawing software under Windows. The internal structure can be divided into four parts: File header, Bitmap Info Header, Color Map and Raw Data. They are introduced as follows:

- (1) File header information: The data structure of BMP file header information is defined in Microsoft Windows Software Develop Kit Volume 2, and its structure size is 14 bytes. The meaning of each byte is shown in the table below.

File location (byte)	Explanation
0,1	ID of the BMP file
2~5	File size (byte)
6,7,8,9	Reserved, need to fill with 0
0A~0D	The location of the graphic data block (byte)

Therefore, this article will use the following program structure to read the file header information of the BMP image file.

```
typedef struct {
    /* type : Magic identifier, */
```

```
    unsigned short int type;
    unsigned int size;
    unsigned short int reserved1, reserved2;
    unsigned int offset;
} FILEHEADER;
```

- (2) Bitmap Info Header: The size of the graphic description information area is 40 bytes, and the meaning of each byte is as follows:

File location (byte)	Explanation
0E~11	The size of graphical description information block
12~15	Image e width (pixel)
16~19	Image height (pixel)
1A~1B	The total number of image planes, must be 1
1C~1D	The number of bits required to record a pixel (byte)
1E~21	Images data compression method 0: no compression 1: Compress in units of 8 bits 2: Compress in units of 4 bits
22~25	Image data size (byte)
26~29	The number of pixels contained in one horizontal meter
2A~2D	The number of pixels contained in one vertical meter
2E~31	The total number of colors used by this image
32~35	The total number of significant colors in the image

Therefore, this article will use the following program structure to read the Bitmap Info Header of the BMP image file.

```
typedef struct {
    unsigned int size; /* Info Header size in bytes */
    int width,height; /* Width and height of image */
    unsigned short int planes; /* Number of colour planes */
    unsigned short int bits; /* Bits per pixel */
    unsigned int compression; /* Compression type */
    unsigned int imagesize; /* Image size in bytes */
    int xresolution,yresolution; /* Pixels per meter */
    unsigned int ncolours; /* Number of colours */
    unsigned int importantcolours; /* Important colours */
} INFOHEADER;
```

- (3) Color Map: Each color in the color table occupies four bytes. The first three bytes record the three primary colors of B, G, and R in sequence. The fourth byte is reserved and must be filled in. Enter 0, and compare the size of the color table with the image color mode as follows:

Image color mode	Size of Color Map (byte)
Monochrome	8
16 colors	64
256 colors	1024
16 million colors (Full Color)	0

Among them, the full color image has no color map. The color values of the three colors B, G, and R are directly represented in the Raw Data 44 with the size of 3 bytes per pixel.

- (4) Raw Data: The order of storing data in BMP files starts from the lower left corner of the image, from left to right, and from bottom to top, and is stored in the file as paragraphs. Before the uncompressed images data are stored in the raw data, each row of data bytes must be a multiple of 4. If a row of data bytes is less than a multiple of 4, it must be filled with 0. For a monochrome image, only one bit is required for each pixel, for a 16-color image, each pixel is required for 4 bits, for a 256-color image, each pixel is required for 8 bits, 16 million color image requires 24 bits per pixel. It can be seen that the size of the raw data will vary with the image size and the color mode of the image file.

B. Grayscale Image

Generally speaking, when we are doing image processing, we will represent each pixel in a color image as an RGB value, and each color of RGB occupies 8 bits, a total of 24 bits, so it is also called three-channel image. And a grayscale image only needs to be represented by one channel, that is, 8bit. So if we want to convert an RGB color image into grayscale, the most direct and simple way is to add the three RGB values and divide them by three to get the result we want. However, because the human eye actually has different perceptions of the brightness of the three colors, green, red, and blue. The grayscale method mostly uses the function $Gray=f(R, G, B)$ proposed by Hasan and Karam [10] to map the color space to the gray space. Which is;

$$Gray = 0.299 * Red + 0.587 * Green + 0.114 * Blue$$

Its conversion relationship is shown in Figure 1;

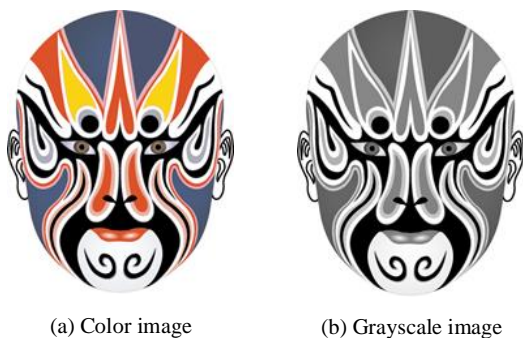


Figure 1 The result of converting a color image to a grayscale image

C. Filtering To Remove Noise And Smoothing

For subsequent processing procedures, noise removal and smoothing are usually required. However, in order to remove noise, the image contrast may be reduced, and good quality may require a lot of processing time. Therefore, a suitable method is usually selected according to actual needs. At present, there are common filtering and smoothing methods: mean filtering, box filtering, Gaussian

filtering, median filtering, bilateral filtering and custom filtering. Generally speaking, filtering methods can be divided into two types, linear filtering and nonlinear filtering. Among them, linear filtering has a core with fixed parameters, and the common ones are mean filtering, box filtering and Gaussian filtering. Non-linear filtering does not have a core with fixed parameters, and common ones are median filtering, bilateral filtering and custom filtering. In this paper, Gaussian filtering is used to remove noise and smooth digital images. Gaussian smoothing filter, also known as low-pass filter, is mainly used to eliminate the high-frequency changes on the image and strengthen the low-frequency part, make the changes of the image more uniform, which can produce a blurrier image and reduce the part of the sharper image. It is usually used to reduce image noise and reduce the level of detail. Different from the mean filter and the box filter, the weight value of each pixel is the same. The mask of the Gaussian filter will strengthen the weighting value of the center point, and reduce the weighting value far from the center point [9]. For example, the mask of Gaussian filter can be shown in Table 1.

Table 1 Gaussian smoothing mask

1	2	1
2	8	2
1	2	1

From a mathematical point of view, the Gaussian blurring process of an image is the convolution product of the image and the normal distribution. Since the normal distribution is also called Gaussian distribution, this process is also called Gaussian blur. The value of each pixel is the weighted average of the surrounding adjacent pixels. The value of the original pixel has the largest Gaussian distribution value, so it has the largest weight value. The further the adjacent pixels are from the original pixel, the smaller the weight value is. If Table 1 is used as the Gaussian smoothing mask of the 3*3 convolution product to process some of the grayscale image data in Table 2, the calculation method is as follows;

Table 2 Partial grayscale image data

197	25	106	156	159
149	40	107	5	71
163	198	226	223	156
222	37	68	193	157
42	72	250	41	75

1. Calculate the sum of the elements in the mask $(1+2+1+2+8+2+1+2+1=20)$.
2. Divide each element of the mask in Table 2 by the sum of the elements calculated in step 1. Then the mask in Table 1 becomes the mask shown in Table 3.

Table 3 New Gaussian smoothing masks

0.05	0.1	0.05
0.1	0.4	0.1
0.05	0.1	0.05

3. Recalculate part of the grayscale image data in Table 2. For example, Gaussian filtering is performed on the pixel with the pixel value of 226 in the third row and third column of Table 3. The calculation method is:

$$\begin{aligned} \text{new value} &= (40 \times 0.05 + 107 \times 0.1 + 5 \times 0.05) + \\ &\quad (198 \times 0.1 + 226 \times 0.4 + 223 \times 0.1) + \\ &\quad (37 \times 0.05 + 68 \times 0.1 + 193 \times 0.05) \\ &= 163.754 = 164 \end{aligned}$$

Table 4 New values of some gray-scale image data

III. THREE-DIMENSIONAL GRAYSCALE ATTRIBUTE FEATURE POINTS AND SURFACE DATA PROCESSING

The method of three-dimensionalizing 2D plane images is to convert the color images into grayscale images, and then determine the level of the image through the properties of the grayscale images. In this article, in order to emphasize the characteristics of relief, the parametric conic elliptic curve is used to correct the 2.5D point group originally generated by the grayscale image. After the feature points are projected and superimposed on the conic curve with three-dimensional characteristics, a B-Spline surface with relief characteristics is constructed.

A. Conic Elliptic Curve Equation

In parametric form, the coordinates of any point on the curve can be expressed as a function of one parameter. However, the parametric conic elliptic curve cannot be corrected in equiangular increments because the curvature at the two ends is too large, so the following formula is used to solve it. If we consider an ellipse whose center is the origin, the semi-major axis is a , and the semi-major axis is b , the parameters of an ellipse can be expressed as:

$$\begin{aligned} x_{i+1} &= x_i \cos \delta\theta - \left(\frac{a}{b}\right) y_i \sin \delta\theta \\ y_{i+1} &= \left(\frac{b}{a}\right) x_i \sin \delta\theta + y_i \cos \delta\theta \end{aligned}$$

(1)

B. Properties of Parametric Surface Curvature

In this article, it is the 2.5D point group generated by the stereoscopic grayscale image, so these feature points are superimposed along the normal direction of the stereoscopic B-Spline surface. And for any point

$r(u,v)$ on any parametric surface, the tangent vector in the u direction is:

$$S_u = \frac{\partial S(u,v)}{\partial u}$$

The tangent vector in the v direction is:

$$S_v = \frac{\partial S(u,v)}{\partial v}$$

Then the tangent length dS between two adjacent points can be expressed as:

$$dS = \frac{\partial S(u,v)}{\partial u} du + \frac{\partial S(u,v)}{\partial v} dv$$

either, $dS = S_u du + S_v dv$

The unit tangent vector T of this point is as follows:

$$T = \frac{dS}{ds} = S_u \frac{du}{ds} + S_v \frac{dv}{ds}$$

(2)

And the normal curvature k of this point is defined

as $\frac{dT}{ds}$, which is:

$$\begin{aligned} k &= \frac{dT}{ds} = \frac{d}{ds} \left[S_u \frac{du}{ds} + S_v \frac{dv}{ds} \right] \\ &= \frac{dS_u}{ds} \left(\frac{du}{ds} \right) + S_u \left(\frac{d^2 u}{ds^2} \right) + \frac{dS_v}{ds} \left(\frac{dv}{ds} \right) + S_v \left(\frac{d^2 v}{ds^2} \right) \\ &= (S_{uu} \frac{du}{ds} + S_{uv} \frac{dv}{ds}) \left(\frac{du}{ds} \right) + S_u \left(\frac{d^2 u}{ds^2} \right) + (S_{uv} \frac{du}{ds} + S_{vv} \frac{dv}{ds}) \left(\frac{dv}{ds} \right) + S_v \left(\frac{d^2 v}{ds^2} \right) \end{aligned}$$

If higher-order terms are ignored, then;

$$\begin{aligned} &= S_{uu} \left(\frac{du}{ds} \right)^2 + 2S_{uv} \left(\frac{du}{ds} \right) \left(\frac{dv}{ds} \right) + S_{vv} \left(\frac{dv}{ds} \right)^2 \\ &= \frac{S_{uu} (du)^2 + 2S_{uv} du dv + S_{vv} (dv)^2}{S_u^2 (du)^2 + 2S_u S_v du dv + S_v^2 (dv)^2} \end{aligned}$$

Among them; $S_{uu} = \frac{\partial^2 S(u,v)}{\partial u^2}$, $S_{vv} = \frac{\partial^2 S(u,v)}{\partial v^2}$,

$$S_{uv} = \frac{\partial^2 S(u,v)}{\partial u \partial v}$$

IV. MACHINING PATH PLANNING

Generally, NC machining can be roughly divided into rough machining, intermediate machining and finishing machining. Among them, rough machining mostly uses contour machining, and uses linear and contour tool paths to quickly remove workpiece blanks and form a prototype of the finished product, thus producing Step-shaped remaining material, and after trimming during intermediate machining, most of the remaining material to be cut in finishing machining is evenly applied to the model, as shown in Figure 2. The method adopted in the tool path planning in this paper is to generate the tool path through the inverse offset method and the point curve, and cooperate with the step length and the step over to obtain the appropriate CL point to complete the machining path.

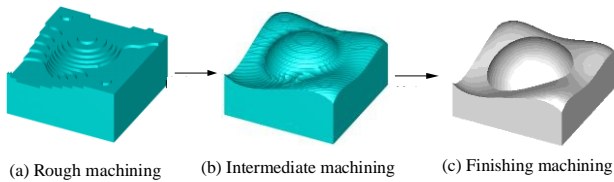


Figure 2 Model during machining process

A. Inverse Offset Method

In this paper, the method of inverse offset method is used to generate the control points of the tool path, as shown in Figure 3. First, inverse the tool and keep the center of the tool falling on the curved surface. When the tool moving along the curved surface, the outermost trajectory formed by the tool contour is the offset surface. During actual cutting, the center of the tool remains on the offset surface, and cutting along the offset surface can obtain the desired surface shape. Taking a spherical milling cutter with radius R as an example, if $\{x(m), y(n), z(m, n)\}$ represents the coordinates of point CC on the current model, then the height of the offset surface at this position $z'(m, n)$ can be obtained by the following formula [6]. This algorithm can avoid the phenomenon of taking too many points in places where the surface is relatively flat, as shown in Figure 6. The method of obtaining the maximum chord deviation can be divided into the following two situations according to the appearance characteristics of the surface [12]:

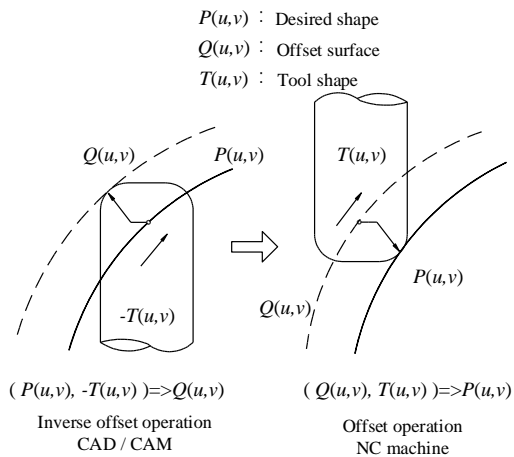


Figure 3 The concept of inverse offset operation

$$z'(m,n) = \max_{(i,j) \in I(m,n:R)} \{z(i,j) + h(i,j,m,n:R)\} \quad (4)$$

$$\text{where } I(m,n:R) = \{(i,j) | (\Delta x_{im}^2 + \Delta y_{jn}^2) \leq R^2\}$$

$$h(i,j,m,n:R) = \sqrt{R^2 - (\Delta x_{im}^2 + \Delta y_{jn}^2)}$$

$$\Delta x_{im} = [x(m) - x(i)]$$

$$\Delta y_{jn} = [y(n) - y(j)]$$

R : Ball cutter radius

Equation (4) indicates that the center position of the tool is the maximum value of "the sum of the height of the surface where each grid point is located and the height of the tool surface" within the range of the projected area of the tool, where $h(i,j,m,n:R)$ represents the height of the tool surface at the grid point position (i,j) . When the position of the grid point is obtained, all the grid points in the same column are connected into a new curve by the passing point curve, as shown in Figure 4, this curve is the tool path.

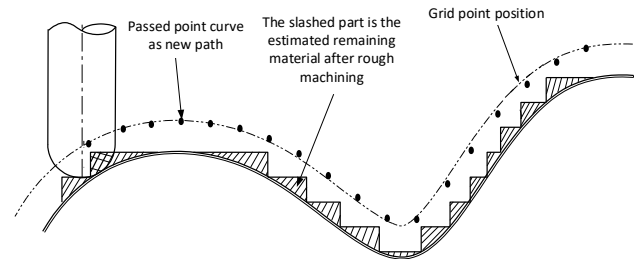


Figure 4 Building a new path from grid points

B. Toolpath Step Length

In NC machining, most of the tools are cut in a straight line. If the number of processing points is small, the processed surface will be rough. If there are many processing points, the surface of the processed workpiece will be smoother, but the time spent on pre-planning will be longer. Therefore, in order to achieve a balance point between the surface roughness of the workpiece and the pre-planning time, it is generally to find a set of straight line segments to approximate the curve and make the maximum chordal deviation less than the given processing tolerance, as shown in the figure 5.

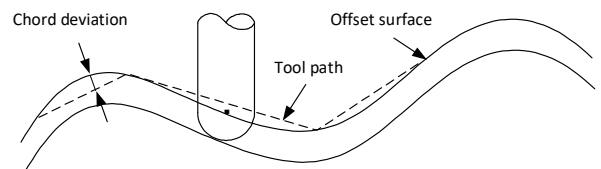


Figure 5 Maximum chord deviation during machining

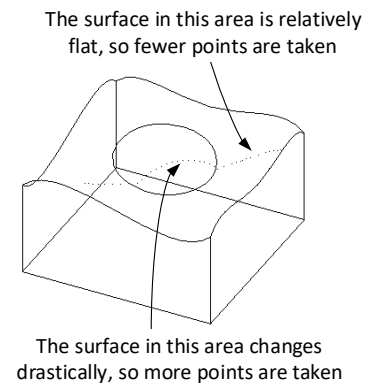


Figure 6 Extraction of CL-point

1. Convex surface

As shown in Figure 7, when θ is small, $\overline{CE} \cong R$

$$\text{Since } \overline{AE} = \overline{AC} - \overline{CE} = \rho \cos \theta - R$$

So the maximum chord deviation δ :

$$\begin{aligned} \delta = \overline{DE} = \overline{AD} - \overline{AE} &= (\rho - R) - (\rho \cos \theta - R) \\ &= \rho(1 - \cos \theta) \end{aligned} \quad (5)$$

$$\text{Where, } \theta = \frac{1}{2} \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2)$$

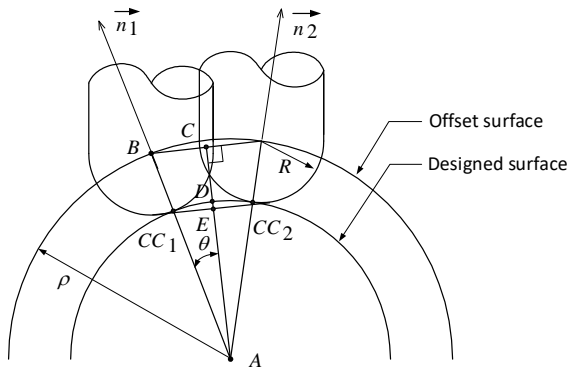


Figure 7 Chord deviation of convex surface

2. Concave surface

As shown in Figure 8, when θ is small, $\overline{CD} \cong R$

So the maximum chord deviation δ :

$$\begin{aligned} \delta = \overline{DE} = \overline{AE} - \overline{CD} - \overline{AC} &= \rho + R - R - \rho \cos \theta \\ &= \rho(1 - \cos \theta) \end{aligned} \quad (6)$$

$$\text{Where, } \theta = \frac{1}{2} \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2)$$

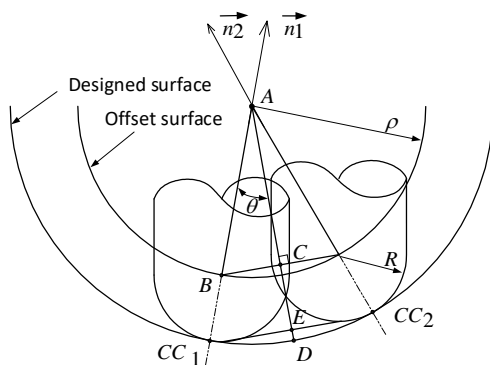


Figure 8 Chord deviation of concave surface

V. SYSTEM OPERATION STEPS

This article uses Visual C++ as the programming basis, and uses the Open CV function library as an image processing tool to successfully develop a cylindrical planar image processing system. Its execution steps are as follows:

Step 1: Load 2D planar image (*.BMP).

First load the 2D plane color image, as shown in Figure 9, and Figure 10 is the data structure of this image.



Figure 9 2D plane color image

demo.bmp																
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000h:	42	4d	98	37	08	00	00	00	00	00	36	00	00	00	28	00
00000010h:	00	00	7a	01	00	00	da	01	00	00	01	00	18	00	00	00
00000020h:	00	00	62	37	08	00	20	2e	00	00	20	2e	00	00	00	00
00000030h:	00	00	00	00	00	00	00	ff	ff	ff	ff	ff	ff	ff	ff	ff
00000040h:	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff
00000050h:	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff
00000060h:	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff
00000070h:	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff	ff

Figure 10 Color image data structure

Step 2: Convert color Image to grayscale.

In order to set the height difference of the image, the color image is converted into grayscale, as shown in Figure 11, and Figure 12 is the data structure of this grayscale image.



Figure 11 2D plane grayscale image

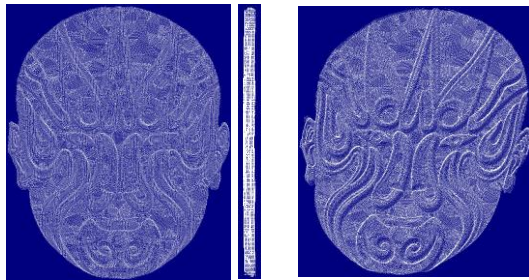
test-gray.bmp																
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000000h:	42	4d	48	28	00	00	00	00	00	00	36	04	00	00	28	00
00000010h:	00	00	64	00	00	00	64	00	00	00	01	00	08	00	00	00
00000020h:	00	00	12	27	00	00	12	0b	00	00	12	0b	00	00	00	00
00000030h:	00	00	00	00	00	00	00	00	00	00	01	01	01	00	02	02
00000040h:	02	00	03	03	03	00	04	04	04	00	05	05	05	00	06	06
00000050h:	06	00	07	07	07	00	08	08	08	00	09	09	09	00	0a	0a
00000060h:	0a	00	0b	0b	0b	00	0c	0c	0c	00	0d	0d	0d	00	0e	0e
00000070h:	0e	00	0f	0f	0f	00	10	10	10	11	11	11	11	00	12	12
00000080h:	12	00	13	13	13	00	14	14	14	15	15	15	15	00	16	16

Figure 12 Grayscale image data structure

Step 3: Generate 2.5D point group.

Set the height difference of the grayscale image to generate 2.5D point group data, which is the depth

value of the finished product to be cut. Figure 13 is the point group generated by the height difference of this gray scale image.



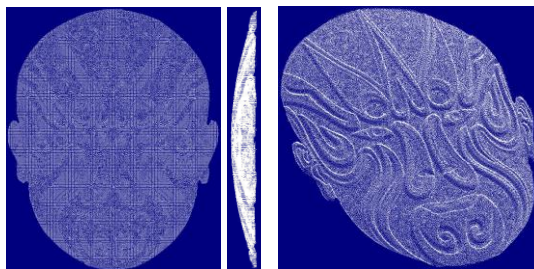
The top view and side view of point group data

The isometric view of point group data

Figure 13 2.5D point group data view

Step 4: Correct the 2.5D point group data to be distributed in a three-dimensional conical ellipse.

In order to emphasize the characteristics of the relief, the parametric conic elliptic curve is used to correct the 2.5D point group data originally generated by the grayscale image, as shown in Figure 14.



The top view and side view of point group data

The isometric view of point group data

Figure 14 three-dimensional conical ellipse point group data view

Step 5: Construct the B_Spline surface.

The corrected point group data is used to construct a B_Spline surface as a reference for the cutting surface, as shown in Figure 15.

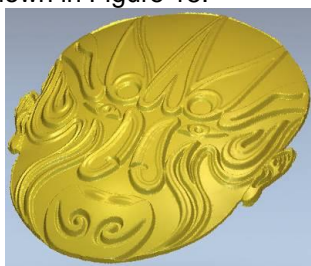
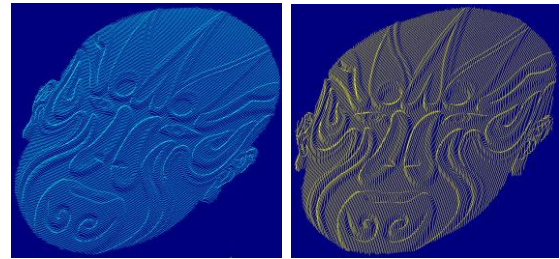


Figure 15 B_Spline surface

Step 6: Construct the CL curve in the UV direction.

Based on the toolpath step length limit criterion, find the CL curve in the appropriate UV direction, as shown in Figure 16.

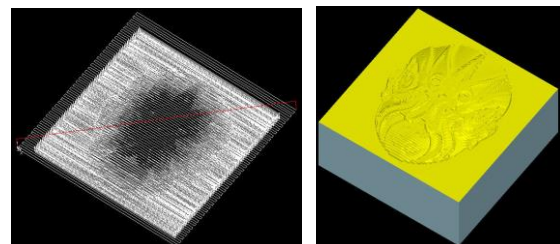


The CL curve in the UV direction

Figure 16 CL curve in UV direction

Step 7: Generate rough and finish machining paths.

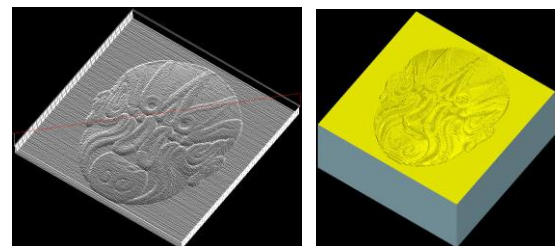
Use the inverse offset method to calculate the cutting path for rough and finish machining, as shown in Figure 17 and Figure 18.



Rough machining toolpath

Rough machining solid cutting simulation

Figure 17 Rough machining tool path and solid cutting simulation



Finish machining toolpath

Finish machining solid cutting simulation

Figure 18 Finish machining tool path and solid cutting simulation

Step 8: Cylindrical cutting paths for rough and finish machining.

Using the calculation principle of geometric projection, the machining path of rough and finishing machining is converted into cylindrical coordinate values to achieve the purpose of cylindrical machining, as shown in Figure 19.

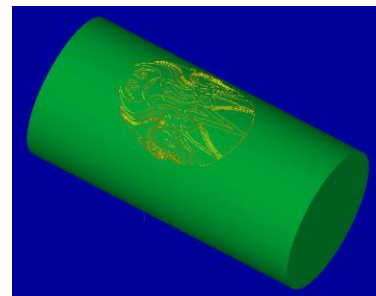


Figure 19 Solid cutting simulation of contour tool path

Step 9: Actual machining.

Finally, through the four-axis NC machine, as shown in Figure 20, the processing of the entity is completed.



Figure 20 Four-axis machine for solid machining



Figure 23 Finished products-3

VI. CONCLUSION

This paper first converts the 2D plane color image (*.BMP) into a grayscale image through image processing technology, then use the feature points determined by its grayscale attributes relative to the cutting depth, project and superimpose them on the conic surface with 3D features, and construct a B-Spline surface with relief characteristics. Cooperate the limiting conditions of the chord length and cutter interval, and the inverse offset method comes to get appropriate CL points of cutting paths of 3D relief part, and then cylindrical the CL points of cutting paths of them. In this way, not only can these auspicious patterns that still remain in 2D plane images be three-dimensional, but even these features can be carved out through NC machines to make them more personalized and interesting, so as to complete the promotion and maintain the inherent traditional aesthetics. However, Figure 21 to Figure 23 are finished products processed by this system.



Figure 21 Finished products-1



Figure 22 Finished products-2

REFERENCES

1. Liu Haiteng, "Research and Software Development of Computer-Aided Workpiece Profiling", Master Thesis, National Taiwan Institute of Technology, Taipei, 1996.
2. Piegl, L. A., Tiller, W., "Computing offsets of NURBS curves and surfaces," Computer-Aided Design, Vol. 31, pp. 147-156, 1999.
3. Choi, B. K., Lee, C. S., Hwang, J. S. and Jun, C. S., "Compound surface modeling and machining," Computer-Aided Design, Vol. 20, No. 3, pp 127-136, 1988.
4. Kim, B. H., Choi, B. K., "Guide surface based tool path generation in 3-axis milling: an extension of the guide plane method", Computer-Aided Design, Vol. 32, pp. 191-199, 2000.
5. Luis, N. L., Aitzol L., "Machining", publisher: Springer London, pp. 225-248, January, 2008.
6. Kondo, T., Kishinami, T., Saito, K., "Machining system based on inverse offset method," Japan Society of Precision Engineering, Vol. 54, No. 5, pp. 167-172, 1988.
7. Lin Canwei, "Tool Orientation Planning for Five-axis NC Machining of Pockets", Master Thesis, National Taiwan University of Science and Technology, Taipei, 2001.
8. You, C. F., Chu, C. H., "Automatic Correction of Interference in Five-Axis NC Machining of Multiple Surfaces," Journal of the Chinese Society of Mechanical Engineers, Vol. 17, No. 5, pp. 435-442, 1996.
9. Miao Shaogang, "Flexible Application of Digital Image Processing-Matlab", Quanhua Science and Technology Books Co., Ltd., 2007.
10. <https://kknews.cc/tech/vm9vmy2.html>
11. Li Lizong, "OpenCV Image Processing Using Python", Deep Wisdom Digital Co., Ltd., 2019.
12. Chang Hui Chin, "Discussion on Optimum Feed Rate for High Speed Machining", Delin Journal No. 27, pp. 69-88, 2014.