

Research on the Integration of 2D Graphics and 3D Printing System

Hui-Chin Chang

HungKuo Delin University of Technology, Department of Creative Product Design, Taipei, Taiwan
chang.hcjang@gmail.com

Abstract—A 3D printer is a machine that can print out 3D files as entities. As long as you use common 3D drawing software on the market to construct a 3D model, and then convert the file through an open source program (such as Slic3r), the 3D printer can help you print out the finished design. However, for ordinary users, how to construct the required 3D models is a technical threshold that is difficult to overcome. Therefore, even though the current price of 3D printers is within the acceptable range of the general public, they cannot be widely used by ordinary users. This paper uses the Python programming syntax combined with the OpenCV image processing library to find out the contour loop of the graph through grayscale, smooth, edge detection, find the contour, and output the contour file in DXF format. Finally, import to the 3D drawing software to construct the 3D model, and output it into an STL format file that the 3D printer can accept, so as to complete the integration of the 2D graphics and the 3D printer system.

Keywords—OpenCV, 3D printer

I. INTRODUCTION

Digital image processing generally refers to the use of computers to process images. An 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 gray scale 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 has its specific brightness, position and other information. This unit cell is called an image element or pixel. Therefore, a gray-scale digital image can also be regarded as a matrix, the row and column determine a point, and the corresponding matrix elements are regarded as the gray-scale of the point. The grayscale is an 8-bit value ranging from 0 to 255, where 0 corresponds to black and 255 corresponds to white. After the image signal is digitized, it can be analyzed and processed by image processing software, so that the image can be converted into the information we need for subsequent applications [1].

Digital images contain three basic properties: resolution, number of layers, and number of faces. Among them, the resolution refers to how many finite pixels an image is composed of, and is usually represented by the product of row and column pixels; the number of layers refers to the discrete value of the

light intensity distribution of the bit image, and each pixel is usually represented by n bits 2^n degree values that can be presented. For example, for an 8-bit digital image, the range of layers is $2^8 = 0 \sim 255$; The number of faces refers to how many pixel matrices a digital image is composed of. For example, a grayscale image is composed of one plane, while a color image is a digital image composed of three planes of red, blue and green [2].

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 the ensuing will also cause a lot of trouble in analysis or processing. Therefore, in order to reduce the noise of the image, filters are used to eliminate the noise. In general, filters can be divided into frequency domain filters and space domain filters. The so-called frequency domain is to regard the original signal as composed of different orthogonal signals. The common method is to express it by Fourier transform, and convert the original image to the frequency domain by discrete Fourier transform. Each point represents a basic frequency, and then according to the frequency Indicates the method to do the processing. The spatial domain directly represents the characteristics of the image. It is a 2D function $f(x, y)$. The gray scale of each point (x, y) represents the physical characteristics of the image at this point. The characteristics of each point can also be correlation with the surrounding gray value. In this paper, the Gaussian spatial domain filter method is used to eliminate the noise and smooth in the image.

The purpose of edge enhancement is to mark the points with obvious brightness changes in the image, because the points with large brightness changes can be regarded as the shape or outline of the feature, and then it is a technique to extract the feature. Also, since points with large brightness changes mostly occur at edges, it can also be called edge detection processing. After the image data is processed by edge detection, it can not only greatly reduce the amount of data, but also eliminate irrelevant information and retain the important structural attributes of the image. Commonly used edge detection filters include Sobel operator, Laplace operator, and Canny filter [3][4]. This article intends to use the Sobel operator to detect the edge of the graph, and then use the Canny filter to strengthen the edge of the graph. Most of the current CAD systems can accept the "DXF" file format for the input of

graphic data. The so-called DXF here is the abbreviation of AutoCAD DXF (Drawing Interchange Format or Drawing Exchange Format), it is a CAD data file format developed by Autodesk for CAD data exchange between AutoCAD and other software. At present, Autodesk has announced the DXF specification of AutoCAD 2007 from AutoCAD Release 13 released in November 1994 to March 2006 on its website. Since AutoCAD Release 10 released in October 1988, DXF supports both ASCII and binary format data [5].

Python is a widely used high-level programming language and belongs to the general-purpose programming language. As a literal translation language, Python's design philosophy emphasizes code readability and concise syntax (especially the use of spaces and indentation to demarcate code blocks, rather than using curly braces or keywords). Python enables developers to express ideas with less code than C++ or Java, the language makes the structure of programs clear, whether small or large. At the same time, like Scheme, Ruby, Perl, Tcl and other dynamically typed programming languages, Python has a dynamic type system and garbage collection function, which can automatically manage memory usage. And supports multiple programming paradigms, including object-oriented, imperative, functional and procedural programming. Therefore, the Python programming language is currently widely used in various fields such as operating system background development, data analysis, network development, artificial intelligence, web crawler, and image processing [6].

Therefore, this paper uses the Python programming syntax combined with the OpenCV image processing library to find out the contour loop of the graph through grayscale, smooth, edge detection, find the contour, and output the contour file in DXF format. Finally, import to the 3D drawing software to construct the 3D model, and output it into an STL format file that the 3D printer can accept, so as to complete the integration of the 2D graphics and the 3D printer system.

II. OPENCV BASIC ARCHITECTURE

The full name of OpenCV is: Open Source Computer Vision Library. OpenCV is a cross-platform computer vision library released under the BSD license (open source), which can run on Linux, Windows, Android and Mac OS operating systems. It is lightweight and efficient-consists of a series of C functions and a small number of C++ classes, and provides interfaces for languages such as Python, Ruby, and MATLAB, and implements many general-purpose algorithms in image processing and computer vision. Although OpenCV is written in C++ language, its main interface is also C++ language, but still retains a large number of C language interfaces. The library also has extensive interfaces to Python, Java and MATLAB/OCTAVE (version 2.5), and the API interface functions of these languages can be obtained through online documents. Now also provides support for C # and Ruby. Today, OpenCV has been widely used in the fields of

human-computer interaction, object recognition, image segmentation, face recognition, action recognition, motion tracking, robotics, motion analysis, machine vision, structural analysis, and automobile safety driving [7].

OpenCV can be divided into different modules, and its main modules are shown in Figure 1:

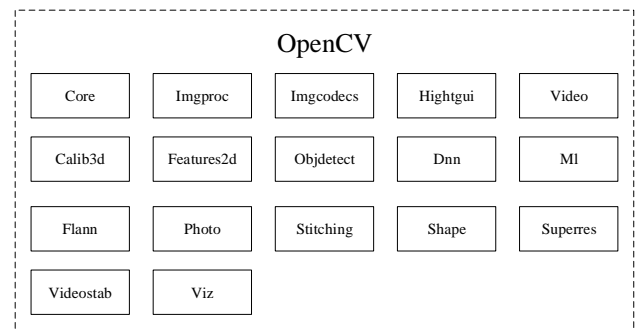


Figure 1 OpenCV basic module

The functions of the modules related to this article are as follows:

- Core : The core module is a module that defines the basic structure, and also includes the basic functions of all other modules in the library.
- Imgproc : Image processing module, including image filtering, geometric image transformation, color space transformation and histogram processing functions.
- Imgcodecs : Image file read and write processing functions.

III. 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 [8].

- (1) Front-end processing: In the application of digital image processing, the front-end processing program must first be used to enhance image contrast, grayscale and noise removal to facilitate the mid-stage processing program.
- (2) Mid-stage processing: The mid-stage processing process can be divided into image segmentation and representation and description. The purpose of image segmentation is to divide the digital image into several areas; for example, the binary method, equalization, area method, boundary method and edge method, etc., all use different threshold values to separate the main objects for reparation the post-processing. The representation and description are to display the important features of the image information as a basis for image quality adjustment; for example, the phase value distribution of a grayscale image can be used as a basis for the equalization of the entire image.
- (3) Post-processing: how to combine applications with other fields is the category of post-processing; for

example, combining neural network-like algorithms and image processing technology to identify patterns is an image post-processing processing function.

A. 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 [6] 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 2;

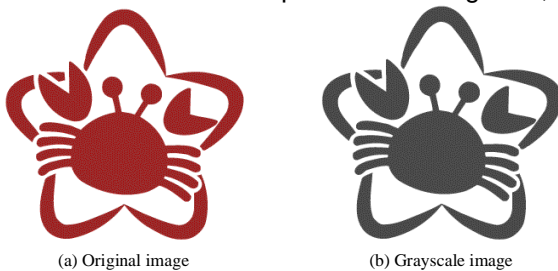


Figure 2 Grayscale image picture

B. 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:

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

Therefore, if the 5*5 filter mask is used as the benchmark, and the standard deviation of the mask in the X-axis and Y-axis directions is set to "0" to filter

out the noise in the image, the execution result is shown in Figure 3. Show;

Table 4 New values of some gray-scale image data

		164		

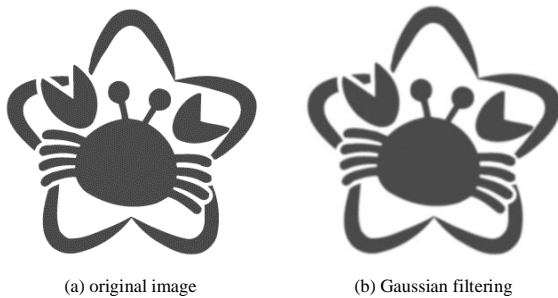


Figure 3 Comparison of the original image and Gaussian filter smoothing

C. Edge detection

After completing the filtering and smoothing of the image, the next step is to find the edge of the image and then draw the outline of the image. The three most commonly used edge detection methods are: Laplacian, Sobel and Canny, these techniques all use grayscale images, based on the difference in the grayscale of each pixel, different objects have obvious edge features at their boundaries to distinguish them. These three methods all use one-dimensional or even two-dimensional differentiation. Strictly speaking, they can be divided into two types according to their different technical principles: Laplacian was originally called the Laplacian method, which detected zero crossings of the second derivative on intensity changes, while Sobel and Canny used gradient methods, which were calculated by calculating detect changes in the first derivative of intensity of pixel luminosity for edge detection [8]. In this paper, the hybrid mode of Sobel and Canny is integrated, that is, Sobel is used to quickly calculate the gradient value of the image, and then Canny is used to complete non-maximum suppression and image edge detection, and finally complete the image contour drawing to achieve the laser engraving machine implements the requirements of the input format of scanning processing parameters. Therefore, the entire execution process of edge detection can be divided into the following steps:

Step 1: Calculate the gradient size and direction of the image – Sobel techniques

The significance of the image gradient size is the speed of image change. For the edge of the image, where the gray level value changes greatly, the gradient value is also larger. On the contrary, for the smoother part of the image, the change of the gray

level value is small, and the corresponding gradient value is also small. Therefore, the calculation of the image gradient is the collection of information on the edge of the image. In this paper, the "Sobel filter" is first used to calculate the gradient value of the image. The main reason is that "Sobel filter" is a discrete differential operator, and this method combines Gaussian filtering smoothing and differential partial derivative operations, so it can process denoising filtering and finding image edges at the same time. Among them, "Sobel horizontal filter" and "Sobel vertical filter" are shown in Table 5;

Table 5 Sobel horizontal filter and vertical filter

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

(a)Sobel horizontal filter (b) Sobel vertical filter

(a) Calculation method of the approximate value of the partial derivative in the horizontal direction

Perform the convolution product calculation of the "Sobel horizontal filter" and the original image "src", then the change of the pixel value in the horizontal direction can be obtained, that is, the value of the partial derivative G_x in the horizontal direction;

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times src$$

Among them, src is the original image, if it is assumed that 9 pixels are as shown in the following table;

-1	0	1	P1	P2	P3
-2	0	2	P4	P5	P6
-1	0	1	P7	P8	P9

To calculate the horizontal partial derivative $P5_x$ of the pixel point P5, the calculation formula is:

$$P5_x = (P3 - P1) + 2 * (P6 - P4) + (P9 - P7)$$

That is, the pixel value of the left pixel point is subtracted from the pixel value of the pixel point on the right side of the pixel point P5. Among them, since the intermediate pixel points P4 and P6 are relatively close to the pixel point P5, the weight of the pixel difference value thereof is 2.

(b) Calculation method of the approximate value of the partial derivative in the vertical direction

Perform the convolution product calculation of the "Sobel vertical filter" and the original image "src", then the change of the pixel value in the vertical direction can be obtained, that is, the value of the partial derivative G_y in the vertical direction;

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times src$$

Among them, src is the original image, if it is assumed that 9 pixels are as shown in the following table;

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} P1 & P2 & P3 \\ P4 & P5 & P6 \\ P7 & P8 & P9 \end{bmatrix}$$

To calculate the horizontal partial derivative $P5_y$ of the pixel point $P5$, the calculation formula is:

$$P5_y = (P7 - P1) + 2 * (P8 - P2) + (P9 - P3)$$

That is, the pixel value of the next row of pixel points is subtracted from the pixel value of the next row of pixel points of the pixel point $P5$. Among them, since the intermediate pixel points $P8$ and $P2$ are relatively close to the pixel point $P5$, the weight of the pixel difference value thereof is 2. However, in practice, the calculated gradient value may produce negative numbers. Therefore, if the processed image is of 8bits type, all negative numbers will be automatically truncated to "0" and information will be lost. Therefore, in order to avoid this situation, we usually take the absolute value of the calculated value during calculation to ensure that the data will not be truncated. Figure 4 is a preliminary edge of the image processed by the Sobel filter.



(a) Image after Gaussian filtering (b) Image after Sobel filter
 Figure 4 The edge of the image processed by the Sobel filter.

D. Non-maximum suppression – Canny techniques

After completing the calculations in the above steps, you can start to check the magnitude and direction of the gradient of each pixel in the image, and remove all non-edge points. In practice, it checks each pixel one by one, and determines whether the pixel is adjacent to the pixel with the same gradient direction and the maximum value, and then decides whether to remove the pixel. Therefore, this step is a process of edge thinning. For example; Table 6 is the partial area data of the image to be processed, the value in the table is the gradient size of the pixel, and the direction of the arrow is the gradient direction of the pixel.

Table 6 Gradient values of data in some regions of the image to be processed

4↑	5↑	3↑	3↑	6↑
3↑	4↑	8↑	8↑	5↗
5↑	7↑	6↑	5↑	3↑
6↑	3↑	5↑	4↗	5↑
3↑	6↑	2↑	4↑	6↑

In Table 6, the points on the black background all have their gradient directions vertically upward (ie, they can be regarded as horizontal edges). Therefore, these points are preserved and the rest are reset to "0". So these black background points will be

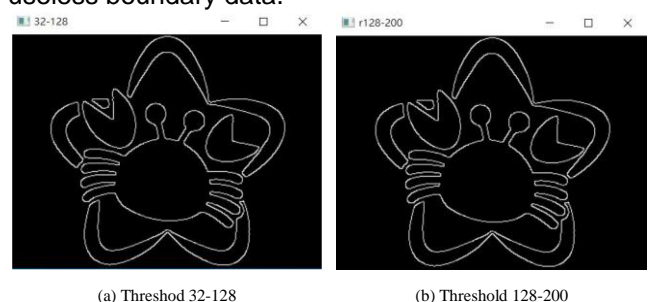
regarded as edge points. Therefore, after the above processing, only one edge point in the same gradient direction will be retained, so the purpose of edge refinement can be achieved. However, after completing the above operations, there is still a lot of noise in the image, so in this article, a "bilateral threshold" is set to filter out inappropriate pixels. The so-called "bilateral threshold" here is to set a high threshold Val_{max} and a low threshold Val_{min} , and then according to the relationship between the gradient value of each pixel and the two thresholds, to determine the attribute of the pixel, its judgment The relationship is as follows;

- (1) If the gradient value of a pixel is greater than or equal to the threshold Val_{max} , mark this pixel as a "main edge".
- (2) If the gradient value of a pixel is between the thresholds Val_{max} and Val_{min} , mark the pixel as a "secondary edge".
- (3) If the gradient value of the pixel point is less than or equal to the threshold value Val_{min} , then the pixel point mark is set to "0", that is, the pixel point is removed as an edge pixel.

Then we process those pixels labeled as "minor edges". This step is mainly to deal with whether the "secondary edge" pixel is connected with the "primary edge", so the judgment principle is;

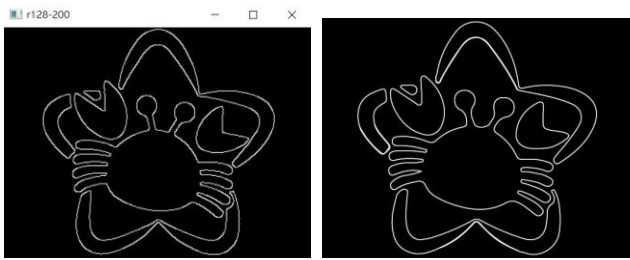
- (1) If the pixel is connected to the "main edge", change the pixel to "main edge".
- (2) If the pixel is not connected to the "main edge", remove the pixel.

Figure 5 represents of the result of processing the edge of a graph with different ranges of threshold intervals (32, 128) and (300, 500). It can be seen from the results that the size and interval of the two thresholds are related to the amount of edge data details. When the threshold is smaller, more details can be captured, but there will be more relatively useless boundary data.



(a) Threshold 32-128 (b) Threshold 128-200
 Figure 5 The comparison of threshold processing between different ranges

Although edge detection can detect the edge of the image, the edge of the processed image is likely to be discontinuous and not a whole, so it cannot meet the needs of the 3D drawing software to construct 3D solid models. Therefore, in the end, it is still necessary to connect the edge and pixels of each region with different contour curve loops, as shown in Figure 6.



(a) Original image (b) The contour curve loops
 Figure 6 The contour curve loops

IV. DXF FORMAT FILE OUTPUT

DXF files are actually composed of pairs of characters and their associated values. These words (commonly known as group codes) are used to indicate the type of the following value. Using these group code and value pairs, DXF files can be organized into sections consisting of records. A record consists of a group code and a data item, each group code and value on its own line in the DXF file. Each section starts with group code 0 followed by the SECTION string, followed by group code 2 and a string representing the section name (for example, HEADER). Each section consists of several group codes and values that define its elements, and a section ends with a group code of 0 followed by the ENDSEC string. The overall organization of the DXF file is as follows [4]:

- HEADER section. Contains general information about the drawing. It consists of the AutoCAD database version number and some system variables. Each parameter contains a variable name and its associated value.
- LASSES section. Holds information about application-defined classes whose instances appear in the BLOCKS, ENTITIES, and OBJECTS sections of the database. Class definitions are permanently fixed in the class hierarchy.
- TABLES section. Contains definitions for the following symbol tables ; APPID (Application Identification Table), BLOCK_RECORD, DIMSTYLE, LAYER, STYLE, UCS, VIEW, VPORT.
- BLOCKS section. Contains the block definitions and drawing primitives that make up the reference of each block in the drawing.
- ENTITIES section. Contains graphical objects (elements) in drawings, including block references (insert elements).
- OBJECTS section. Contains non-graphical objects in the drawing. All objects that are not entities, symbol table records, or symbol tables are stored in this section. A dictionary containing polyline patterns and groups is an example of an item in the OBJECTS section.
- THUMBNAILEDATA section. Contains preview image data of the drawing. This is an optional segment.

Because most of the 3D drawing software accepts DXF format files, this paper outputs the contour

curves outside the pattern into a graphics exchange file in DXF format, and finally imports the 3D drawing software to construct a 3D model, and outputs it as a STL format files, to complete the integration of 2D graphics and 3D printer systems, the result is shown in Figure 7.



(a) The contour curve loops (b) 3D printer finished product

Figure 7 3D printer finished product

V. SYSTEM OPERATION STEPS

As mentioned above, this paper uses the Python programming syntax combined with the OpenCV image processing library to find out the contour loop of the graph through grayscale, smooth, edge detection, find the contour, and output the contour file in DXF format. Finally, import to the 3D drawing software to construct the 3D model, and output it into an STL format file that the 3D printer can accept, so as to complete the integration of the 2D graphics and the 3D printer system. Therefore, the implementation steps in this paper are as follows;

Step 1: Load the original image, as shown in Figure 8.



Figure 8 Original image.

Step 2: Grayscale image, as shown in Figure 9.

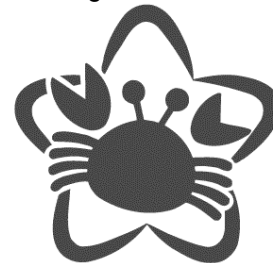


Figure 9 Grayscale image

Step 3: Filtering to remove noise and smoothing, as shown in Figure 10.



Figure 10 Filtering to remove noise and smoothing

Step 4: Detect edges with Sobel bidirectional filter, the result is shown in Figure 11.

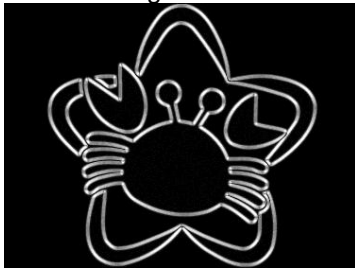


Figure 11 Sobel bidirectional filter edge detection results

Step 5: Suppress the non-maximum value with the Canny function, and the result is shown in Figure 12.

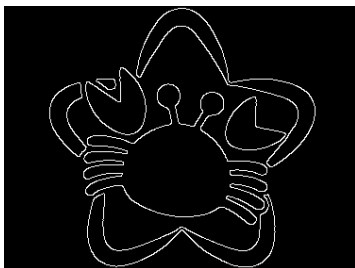


Figure 12 Illustration of Canny function suppressing non-maximum values

Step 6: Draw the contour curve, the result is shown in Figure 13.



Figure 13 The contour curve of the graph

Step 7: Output the DXF file and load it into the 3D drawing software to construct the 3D model, and finally print the finished product with a 3D printer, the result is shown in Figure 14.



Figure 14 3D printed finished product

VI. CONCLUSION

This paper uses the Python programming syntax combined with the OpenCV image processing library to find out the contour loop of the graph through grayscale, smooth, edge detection, find the contour, and output the contour file in DXF format. Finally, import to the 3D drawing software to construct the 3D model, and output it into an STL format file that the 3D printer can accept, so as to complete the integration of the 2D graphics and the 3D printer system. In addition to the image file examples shown in the previous chapters, the following image files are used to present the results completed during the execution of the system, as shown in Figure 15~Figure 18.



(a) Original image



(b) 3D printer finished product

Figure 15 Graphic example of a prouid horse



(a) Original image



(b) 3D printer finished product

Figure 16 Graphic example of a galloping horse



(a) Original image



(b) 3D printer finished product

Figure 17 Aries Constellation Graph Example



(a) Original image



(b) 3D printer finished product

Figure 18 Scorpio Constellation Graphic Example

ACKNOWLEDGMENT

This research is grateful to the Ministry of Science and Technology for encouraging technical schools and colleges to adopt practical project plans (MOST 110-2637-H-237-002) to receive funding support have enabled the successful completion of this research.

REFERENCES

1. Original work by Gonzale.Woods, translated by Miao Shaogang, Digital Image Procesing, Third Edition, Taipei: Princeton Press, 2009.
2. Lin Junjie, Application of image processing in the research of online cell technology and monitoring system, master's thesis, Department of Mechanical Engineering, National Taiwan University of Science and Technology, 2009.
3. Cai Kunzhe, Design and implement real-time low-noise edge-preserving bilateral filter, master thesis, Department of Information Engineering, National Tsing Hua University, 2008.
4. Lin Qiming, Research and Development of Glass Substrate Edge Detection System, Master Thesis, Institute of Mechanical and Precision Engineering, National Kaohsiung University of Applied Sciences, 2008.
5. <https://zh.wikipedia.org/wiki/DXF>
6. <https://zh.wikipedia.org/wiki/Python>
7. <https://kknews.cc/tech/vm9vmy2.html>
8. Miao Shaogang, Flexible Application of Digital Image Processing-Matlab, Quanhua Science and Technology Books Co., Ltd., 2007.
9. Li Lizong, OpenCV Image Processing Using Python, Shenzhi Digital Co., Ltd., 2019.