Gerber File Parsing and the Implementation Method of its Conversion to Bitmap Image

Fan Yong¹a, Wu Shuqin²b*, Xie Wu³c, Yang Jianbo³d

¹School of Mechanical and Electrical Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, People's Republic of China
²School of Architecture and Transportation Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, People's Republic of China
³School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi, People's Republic of China

aE-mail: fywsq@guet.edu.cn, bE-mail: xiesixchannels@126.com, dE-mail: yjb760617@guet.edu.cn

*Corresponding author: bE-mail: wsqfy@guet.edu.cn

Abstract: In order to obtain the standard image based on the Gerber file in the AOI (Automatic Optical Inspection), an implementation method is proposed by analysing and converting the Gerber file into a bitmap image. According to the command format rules of Gerber file, the approach is designed to analyse Gerber files by constructing the command regular expression of Gerber files. In order to achieve convenient access to the drawing data, the drawing data processing is performed by defining a class, then the drawing data is stored in a dynamic array and the adjustment factors are added in the drawing data at the same time to meet the further post-processing needs of bitmap image. This method can quickly analyse Gerber files, adjust drawing data and generate bitmap images.

1. Introduction

In the fields of electronic packaging and assembly technology, the detection algorithms of the AOI (Automated Optical Inspection) system for PCB (Printed Circuit Board) quality detection are divided into reference comparison method, design rule verification method and hybrid Law [1-3]. Among them, the reference comparison method is to compare the reference PCB image, that is, the standard PCB image, with the PCB image to be tested to determine whether the PCB to be tested has defects. One method to obtain a standard PCB image is to use an image acquisition system such as camera or scanner to take photo or scan a PCB picture without defects as a standard image [4]. The standard image obtained by this method is influenced by the factors of the camera system and the PCB production techniques, which makes the standard image deviate from the designed image, resulting in the increases of the misjudgment of detection.

Another method is to analyze the Gerber file and convert it into an image as a standard image [5]. Being converted from Gerber file directly, the standard image obtained by this method can reduce detection misjudgments and improve detection accuracy effectively. This article focuses on the latter.
Yao obtains PCB standard images by analyzing the Gerber file with regular expressions and stored data via the Map linked list [6]. Zhang applies morphological methods to modify the parsed Gerber file to create PCB standard image [7]. He defines multiple A class, which are applied to analyze the Gerber file and to store data via List linked list [8].

Among the existing PCB design and manufacturing systems, most of them (methods to obtain PCB standard images) can only browse the images but cannot output the images as bitmap format files. In addition, the images output cannot meet the multiple requirements of AOI detection, resulting in the image post-processing algorithms much more complicated.

2. Gerber file and regular expression

2.1. Gerber file

The Gerber file format is the standard for PCB design data transmission and the backbone of the electronics manufacturing industry as well. All PCB design systems can output Gerber file and all PCB manufacturing software can process them, consequently all PCB design data can be exchanged safely and effectively within PCB professionals. There are multiple current revisions of the Gerber file standard. The Gerber file described in this article follows the Gerber format specification 2019.06 revision unless stated otherwise [9].

Gerber files are provided by the PCB designer to the manufacturer. They are a series of files used to produce circuit boards, and their contents are composed of various commands. Command is the basic unit of Gerber file, which is divided into function code command and extended command. Function code command is represented by a series of characters ending with the character '*' while the extended command is command enclosed by a pair of '%' characters.

A Gerber file completely defines a PCB image, which can be displayed in the form of graphics and then can be converted into a bitmap image by parsing of the image. Multiple classes are all defined among the existing Gerber file data analysis and storage systems, but the storage structures of the linked list result in complicated data access.

2.2. Regular expression

A regular expression is a character pattern that matches a character sequence in a text. Its function is to judge whether a character sequence matches a pattern. In order to construct regular expression, certain characters and character combinations are specified with special meanings and functions. According to the character sequence, the regular expression of the character sequence is designed and constructed by applying regular expression characters and their combinations, to match the character sequence accurately.

The parsing of the Gerber file is to use a series of regular expressions to match each Gerber file command. A successful match means that the program knows what the command is and can perform corresponding operations according to the function of the command.

3. Gerber file analysis and conversion

3.1. Technical Route

The technical route of parsing Gerber files and converting them into bitmap images is:

• According to the Gerber file command format rules, design and construct Gerber file command regular expressions basing on full analysis of Gerber files and regular expressions.
• Obtain the drawing information by applying the constructed regular expression to analyze the Gerber file.
• Save the drawing data according to the analysis results.
• Add the adjustment factors to the drawing data basing on the actual demands.
• Create the bitmap object, draw the adjusted drawing data on the bitmap object and saved.

The technical route map is shown in Figure 1.
3.2. Implementation steps

**Step 1** Construct the Gerber file command regular expression according to the Gerber file command format rules.

The format characteristics of Gerber file commands are analyzed and studied. Basing on this, regular expression that matches the Gerber file commands are designed and constructed to parse and obtain drawing information from the commands accordingly. Some regular expressions are as follows:

- `^%?(AD)(D\d{2})(\w),(\d+\.\d+)(X)?(\d+\.\d+)?( X)?(\d+\.\d+)?\*%?$`,
- `^%(FSLAX)\d(\d)Y\d(\d)\*%$`,
- `^%(MO)(MM|IN)\*%$`,
- `^%(LP)(C|D)\*%$`,
- `^(X)(-?\d+)(Y)(-?\d+)(D01|D02|D03)\*$`,
- `^(X)(-?\d+)(D01|D02|D03)\*$`,
- `^(Y)(-?\d+)(D01|D02|D03)\*$`,
- `^(G0\d)?(X)(-?\d+)(Y)(-?\d+)(D01|D02|D03)\*$`,
- `^(G0\d)?(X)(-?\d+)(Y)(-?\d+)(J)(-?\d+)(D01)\*$`.

**Step 2** Apply the constructed regular expression to parse the Gerber file to obtain drawing information.

1. Define a class.
   - Class defines multiple attributes, which correspond to the command name, variables and/or parameters of the drawing information in the Gerber file. Compared with the previous method of defining different classes for different drawing information, the application of a single class method simplifies the process of drawing data storage and extraction.
2. Define eight different generic sets of class, which are respectively marked as generic set A, B, ..., H.
3. Define two local variables and mark them as local variables A and B respectively.
4. Define one generic set Z of class generic set.
5. Define a dynamic array A.
6. In accordance with the order in which the Gerber file commands appear, take out one command at a time and match it with all the regular expressions constructed in turn. When the match is successful, it indicates that a command corresponding to the regular expression is retrieved, and the drawing information contained in the Gerber file command is obtained according to the meaning and function of character expressed in the regular expression.

In Gerber files, the main commands related to drawing information include: FS commands, MO commands, AD commands, LP commands, D codes, G codes, and M codes. The drawing information mainly includes:

(1) The FS command specifies the format of the coordinate data in the Gerber file. The command starts with FS and includes: L, which means omitting leading zeros; A, which means using absolute coordinates; X, which means X coordinates; Y, which means Y coordinates; I, represents the maximum number of integers; D, represents the maximum number of decimals.

(2) MO command specifies the unit of data in the Gerber file. The command starts with MO and includes: IN for inches, or MM for millimeters.

(3) AD command is to create a diameter. The command starts with AD, and then includes: D, for aperture; nn, for aperture number, \( \geq 10 \); C, R, O, or P, respectively, for circle, rectangle, oval and polygon. Four kinds of apertures: circle, rectangle, oval or polygon size data. The four types of apertures are specifically:

1) Circular aperture including diameter and hole.
2) Rectangular aperture including X-direction size, Y-direction size and hole size.
3) Oval aperture including X-direction size, Y-direction size and hole size.
4) Polygon aperture including the diameter of the circumscribed circle, the number of sides, the included angle and the hole size.

(4) LP command specifies the polarity of the graphics state. The command starts with LP, followed by: C, which means that the polarity is clear; D, which means that the polarity is dark.

(5) D code includes D01, D02, D03 and Dnn (nn \( \geq 10 \)). D01, D02 and D03 are operation codes, which define an operation command together with coordinate data. D01, interpolation operation, means to interpolate from the current point to the operation coordinates to create a straight-line segment or arc. D02, move operation, means to move the current point to the operation coordinates, no image object is generated. D03, flash operation, means to copy the current aperture at the operating coordinates to create a flash image object. Dnn means aperture nn.

Operation code commands usually start with X, which means X coordinate, and then include X coordinate value; Y means Y coordinate and Y coordinate value; I, represents distance or offset in X direction and distance value or offset value in X direction; J, represents the distance or offset in the Y direction and the distance value or offset value in the Y direction.

(6) G code mainly includes G01, G02, G03, G36 and G37. The G code starts with G, followed by two digits. G01 specifies D01 as the linear interpolation mode, the interpolation operation of D01 will generate a straight line from the current point to the X, Y coordinate point specified by the operation, and set the current point as X, Y coordinates.

G02 and G03 specify D01 as the arc interpolation mode. The interpolation operation of D01 generates an arc from the current point to the X and Y coordinates specified by the operation. The center of the arc is specified by the offsets I and J, and the current point is set as X, Y coordinates, where G02 means clockwise rotation and G03 means counterclockwise rotation.

G36 and G37 create a group of regional image objects. G36 means region start command, G37 means region end command while D03, Dnn and M02 (End of Gerber file command) command are not allowed to be used in the region.

(7) M code, mainly M02, indicates the end of Gerber file command.

7. Save part of the drawing information.

According to the order in which the Gerber file commands appear, one command is taken out at a time, and matched with all the regular expressions constructed in turn. When the FS command is retrieved, assign its drawing information L, A, I and D to the attribute corresponding to the class.
When the MO command is retrieved, IN or MM is assigned to the attribute corresponding to the class. When the LP command is retrieved, the C or D is assigned to the attribute corresponding to the class.

8. Save AD command drawing information.

According to the order of appearance of the Gerber file commands, one command is taken out at a time, and matched with all the regular expressions constructed in turn. When the AD command is retrieved, the drawing information D, \( mn \), C (or R, O, P) and its size data are respectively assigned to the corresponding attributes of the class. C represents a circle aperture. R represents a rectangle aperture. O represents an obround aperture. P represents a polygon aperture. An instance of the class is generated accordingly and the instance is saved in the generic set A. For example, assuming that the aperture is a circular aperture C, the saved AD command drawing information is: D\( mn \), C, the diameter of the circular aperture and its hole size.

**Step 3** Save the drawing data according to the analysis result.

The analysis result mainly refers to drawing information, while the drawing data refers to a data combination composed of drawing information, by which graphics the be created. The data added to the dynamic array A each time is regarded as a set of drawing data.

The storage of drawing data includes two parts: regional drawing data storage and non-regional drawing data storage. This will be described as the following Figure 2.

**Fig. 2. Storage of drawing data**

1. Storage of non-regional drawing data.

According to the order in which the Gerber file commands appear, one command is taken out at a time, and matched with all the regular expressions constructed in turn. When the D\( mn \) command is retrieved, take the drawing information of the aperture from the generic set A and take it as the current effective aperture information. With the match search continuing, assign the information to local variable A when G01, G02 or G03 is retrieved. With the match search continuing, the description is as follows when D01 or D03 is retrieved:

1) When D01 is retrieved.

According to the value of local variable A, the drawing data is stored separately, specifically:

1) When the value of local variable A is G01.

Assign the current aperture information, the current point coordinate value, the operation coordinate value, and the data identifier to the corresponding attributes of the class to generate an instance of the class. Temporarily, store the instance in the corresponding generic set, add this generic set to the dynamic array A as a drawing data, and empty the generic set.
For example, assuming that the current aperture is a circular aperture and its corresponding generic set is B, then assign the diameter size of the aperture, the X and Y coordinates value of the current point, the X and Y coordinates value of the operating coordinates, and the data identifier to the corresponding attribute of the class and generate an instance of the class. Temporarily, store the instance in the generic set B, add the generic set B to the dynamic array A as a drawing data, and empty the generic set B finally.

2) When the value of local variable A is G02.

In the Gerber file, the aperture used to draw the arc is a circular aperture. Based on this, the storage method of the drawing data is: assign the diameter of the current aperture, the X and Y coordinates of the current point, the X coordinate value and Y coordinate value of the operation coordinate, the offset value of I, the offset value of J, and the data identifier to the corresponding attributes of class, and generate an instance of class. Temporarily, store the instance in the generic set C, add the generic set C to the dynamic array A as a drawing data, and empty the generic set C finally.

3) When the value of local variable A is G03.

Assign the diameter of the current aperture, the X and Y coordinates of the current point, the X coordinate value and Y coordinate value of the operation coordinate, the offset value of I, the offset value of J, and the data identifier to the corresponding attributes of class, and generate an instance of class. Temporarily, store the instance in the generic set D, add the generic set D to the dynamic array A as a drawing data, and empty the generic set D finally.

4) When the value of local variable A is empty.

In the Gerber file, the graphics are drawn according to the G01 command in this case. Based on this, the storage of the drawing data is handled following case 1).

(2) When D03 is retrieved.

According to the value of the local variable A, the drawing data is stored separately, specifically:
1) When the value of local variable A is G02.
   Store the drawing data following case 2) when D01 is retrieved.
2) When the value of local variable A is G03.
   Store the drawing data following case 3) when D01 is retrieved.
3) When the value of local variable A is empty.

Assign the current aperture information, the current point coordinate value, and the data identifier to the corresponding attributes of the class to generate an instance of the class. Temporarily, store the instance in the corresponding generic set, add this generic set to the dynamic array A as a drawing data, and empty the generic set.

For example, assuming that the current aperture is a circular aperture and its corresponding generic set is E, then assign the diameter size of the aperture, the hole size of the aperture, the X and Y coordinates value of the current point, and the data identifier to the corresponding attribute of the class and generate an instance of the class. Temporarily, store the instance in the generic set E, add the generic set E to the dynamic array A as a drawing data, and empty the generic set E finally.

2. Storage of regional drawing data.

(1) According to the order in which the Gerber file commands appear, one command is taken out at a time, and matched with all the regular expressions constructed in turn. When the Dnn command is retrieved, take the drawing information of the aperture from the generic set A and take it as the current effective aperture information. With the match search continuing, assign the information to local variable B when G36 is retrieved. With the match search continuing, assign the information to local variable A when G01, G02 or G03 is retrieved. With the match search continuing and under the condition that the local variable B is G36, the description is as follows according to the value of the local variable A when D01 is retrieved:

1) When the value of local variable A is G01.

Assign the diameter size of the current aperture information, the X coordinate value and Y coordinate value of the current point, the X coordinate value and Y coordinate value of the operation coordinate, and the data identifier to the corresponding attribute of the class to generate an instance of
the class. Save it in the generic set F, add the generic set F to the generic set Z as part of the drawing data of the area, and empty the generic set F finally.

2) When the value of local variable A is G02.
Assign the diameter size of the current aperture, the X coordinate value and the Y coordinate value of the current point, the X coordinate value and the Y coordinate value of the operation coordinate, the offset value of I, the offset value of J, and the data identifier to the corresponding attributes of class to generate an instance of class temporarily, save the instance in the generic set G, add the generic set G to the generic set Z as part of the drawing data of the area, and empty the generic set G finally.

3) When the value of local variable A is G03.
Assign the diameter size of the current aperture, the X and Y coordinates of the current point, the X and Y coordinates of the operation coordinates, the offset value of I, the offset value of J, and the data identifier to the corresponding attributes of class to generate an instance of class temporarily, save the instance in the generic set H, add the generic set H to the generic set Z as part of the drawing data of the area, and empty the generic set H finally.

4) With the match search continuing, add the generic set Z to the dynamic array A as the drawing data of the area, and then empty the generic set Z when G37 is retrieved.

Step 4 Add adjustment factors to the drawing data.
(1) Set adjustment factors A and B.
Set the values of A and B according to the demands of later AOI detection. Adding adjustment factors can simplify the post-image processing algorithm, improve the accuracy of the local standard image, and make the local standard image match the local PCB detected by AOI much better.

(2) Take out a group of drawing data in order according to the sequence of storing data in dynamic array A.
(3) Add adjustment factors.
Add adjustment factors A and B as follows:
1) For coordinate values:
   
   \[
   \text{NewX} = (\text{OldX} + A) \times B
   \]
   
   \(1\)
   
   and
   
   \[
   \text{NewY} = (\text{OldY} + A) \times B
   \]
   
   \(2\)

2) For distance or offset:
   
   \[
   \text{NewL} = \text{OldL} \times B
   \]
   
   \(3\)

Here, NewX, NewY, NewL respectively represents the adjusted X coordinate value, adjusted Y coordinate value, distance or offset value. OldX, OldY, OldL respectively represents the X coordinate value, Y coordinate value, distance or offset value before adjustment.

Step 5 Create bitmap objects, draw graphics, and save bitmap files.
(1) Create a bitmap object using the image device interface GDI+.
(2) According to the sequence of extracting the drawing data from the dynamic array A, draw the graphics described by each group of drawing data into the bitmap object using the drawing tools with GDI+.
(3) Save the bitmap file after drawing.

3.3. Programming implementation
According to the above technical route and implementation steps, the Microsoft Visual Studio Community is used to create a Windows forms application (.NET Framework) project, the C# language is used for programming, and the HTML Help Workshop and its programming language are used to write help files of software.

In addition to the public class automatically generated by the project and inherited from the base class Form, the namespace needs to create a new public class, that is the class defined in Step 2 above. The members of this class include variables and methods. Variables include those used in Gerber file parsing and drawing, such as various apertures, coordinates, and variables involved in related
parameters. Methods mainly involve converting the parsed parameter value (in this case, it is a string form) from a string form to an equivalent value form.

4. Results
In order to verify the feasibility of the analytical method and the accuracy of the analytical results, the top-level Gerber file of a certain circuit board is analyzed. The image obtained is shown in Figure 3(a). Comparing it with the final image shown from the mainstream software CAM350 (Figure 3(b)), the results are consistent in general. Since Figure 3(b) is obtained by means of a screenshot, there is a slight difference from Figure 3(a).

![Fig.3. The results of this article are roughly similar to CAM350](image)

At the same time, the analytical efficiency of this method is tested. The test CPU is Intel Core i7-7500U 2.7GHz, the memory is 8GHz, the development environment is Microsoft Visual Studio Community 2019, and the PCB size to be analyzed is 157mm×91mm. The test results are shown in Table 1. The analysis time in the table is the average value of 10 analysis times of the corresponding file.

| Layer name   | File size/line | Parsing time /ms |
|--------------|---------------|-----------------|
| B_Mask       | 31270         | 520             |
| B_Paste      | 77            | 17              |
| B_SilkS      | 932           | 37              |
| Bottom_layer | 14207         | 275             |
| F_Mask       | 33512         | 564             |
| F_Paste      | 824           | 40              |
| F_SilkS      | 11798         | 224             |
| GND_layer    | 11341         | 224             |
| NPTH-drl     | 14            | 3               |
| PTH-drl      | 646           | 35              |
| Top_layer    | 7042          | 174             |
| VDD_layer    | 11423         | 235             |
| Average      | 10257         | 196             |

It can be seen from the table that for a Gerber file with an average of 10,000 lines, the parsing time is less than 200ms, and the parsing speed is faster, which is better than general CAM software. This method has been applied to AOI detection system with good results.

5. Conclusions
To aim at the needs of AOI detection, this article proposes the regular expression to analyze Gerber file to obtain drawing data. In order to achieve convenient access to the drawing data, a class is
defined, and the data is stored in a dynamic array. At the same time, in order to meet the needs of image post-processing, the bitmap image is finally obtained by adding adjustment factors to the drawing data. The experimental results show that this method can parse Gerber files quickly and accurately. Virtual simulation processing is the possible direction for future work.

Acknowledgments
This work was supported by Guangxi Science and Technology Planning Project (Nos. AD19110137, AB18126063), Guangxi Key Science and Technology Planning Project (Nos. AA18118031, AA18242028), and Guangxi Key Research and Development Planning Project (Nos. 2018AB06001).

References
[1] Wang, F., Li, X.M., Xu, G. (2009) The PCB defect inspection system design based on lab windows. In: International Conference on Industrial Mechatronics and Automation. Chengdu. pp. 485-487.
[2] Guan, S., Guo, F.L. (2011) A new image enhancement algorithm for PCB defect detection. In: International Conference on Intelligence Science and Information Engineering. Wuhan. pp. 454-456.
[3] Huang, Y.W. (2018) Research and application of optical inspection technology for PCB bare board defects. Nanjing University of Science & Technology, Nanjing.
[4] Mateo Sanguino, T.J., Smolcic-Rodriguez, M. (2012) Computer-aided system for defect inspection in the PCB manufacturing process. In: IEEE 16th International Conference on Intelligent Engineering Systems (INES). Lisbon. pp. 151-156.
[5] Wang, R. (2016) A design of AOI about the defect detection based on the surface of PCB bare board. Beijing University of Technology, Beijing.
[6] Yao, J., Ye, Y.T., Zhang, J., Xie, Y., Zhou, L.L. (2012) Research of parsing Gerber file in PCB automatic optical inspection, 33: 2481-2485.
[7] Zhang, J., Ye, Y.T., Xie, Y., Liu, L., Chang, Y.X. (2013) Appearance detection for printed circuit board by Gerber file. Optics and Precision Engineering, 21: 2679-2687.
[8] He, X.W. (2018) The research of analysis of Gerber file and its graphic drawing and segmentation in PCB processing. Xidian University, Xian.
[9] Ucamco NV. (2019) The Gerber format specification. https://www.ucamco.com/en/gerber.