Based on the single-layer contour pattern, the comparison of the zigzag parallel hatch and the contour parallel hatch

Jiali Zhao1*

1 Information management and system, Shandong Normal University, Jinan, Shandong, 250300, China
*e-mail: 201824040153@sdnu.edu.cn

Abstract. Laser marking refers to marking on the surface of a product with a laser mark, text, symbol, image, etc. The hatch tool of laser marking machine can be used to hatch specified 2D-compound curve graph. The direction parallel hatch and the contour parallel hatch are two basic ways of hatching. Under the single-layer contour pattern, for the zigzag parallel hatch, we established an improved area Parity Check Filling Model. For the contour parallel hatch, we established an equidistant line generation model based on the Regular Bezier curve, and used the De Casteljau algorithm to generate equidistant lines with the given hatching line spacing. By running the relevant programs, we get the total length of hatch curves and the average elapsed time of the two hatch algorithms. According to the analysis of the result, the performance of the zigzag parallel hatch algorithm is higher than that of the contour parallel hatch algorithm.

1. Introduction
Laser marking means using laser to mark LOGO, characters, symbols, images, etc. on the surface of products. It is a widely used processing method with its advantages of high processing efficiency, non-contact operation, no consumables, slight influences on produce surface deformation, and firmness of marked content.[1]

The hatch tool of laser marking machine can be used to hatch specified 2D-compound curve graph, and the setting of different hatch parameters have a great impact on the processing effects of different materials. The direction parallel hatch and the contour parallel hatch are two basic ways of hatching. The direction parallel hatch, also known as “zigzag” hatch, has paths being moved along line segments which are parallel to an initially selected reference direction. Based on this strategy a connected path is obtained by linking these parallel segments so that they are either all traversed from right to left (or left to right) or, alternately from left to right and from right to left. Whereas the contour parallel hatch uses offset segments base the boundary curves as smooth hatch path that similar to the boundary curve. Thus, the contour parallel hatch be generated in a spiral-like fashion along curves that are at constant distances from the curve boundary. Which type of hatch is applied in practice highly depending on mark materials and the process effects on the particular machining task to be performed.
Figure 1. zigzag parallel and contour parallel hatch of multi-layer curves.

The hatching entities must be closed curvilinear polygons, and multiple mutually-nested contour objects may be filled by group-hatching. For the hatching process of figures, the existing boundary contours shall be offseted inward or outward firstly according to edge distance, before getting subject to the zigzag parallel or contour parallel hatch. The hatch curves shall be even and regular, and be kept basically parallel to each other. Neither having omissions of filling in an area, nor repeated filling of the area is allowed.

The hatching by laser marking shall generate hatched contours online in a real-time way. To meet the requirement of high-efficiency laser marking, the hatch curves should stay parallel to the figures’ boundary lines, distribute evenly to the greatest extent, and be generated automatically and quickly. Efficiency is an important indicator for the generation of hatched figures.

Based on the single-layer contour pattern, we compare the efficiency of the zigzag parallel hatch and the contour parallel hatch.

2. Materials and Methods

To simplify the problems, we make the following assumptions;

Assumption 1: It is assumed that the running speed of the same program is equal under different software, hardware and configuration environments.

Assumption 2: It is assumed that the non-horizontal straight part of the zigzag hatching curve is approximately fit with the boundary contour obtained after the boundary distance is reduced.

and here, only the hatch in horizontal direction (0° degree) is considered for the zigzag parallel hatch.

2.1. Zigzag Parallel Hatch Model

Considering the hatch of the zigzag parallel, we need to fill the zigzag curve in the single-layer outline pattern (as shown in Figure 2). Since the scan line filling algorithm has the advantages of simplicity and speed, we mainly use the Scan line filling algorithm to hatch the graphics in a zigzag shape according to the given hatch spacing.

In order to facilitate scanning and filling, we first process the figure outline. Using the proportional shrinking method,[3][5] the contour of the graph is offset and contracted inward according to a certain boundary distance. We use the improved scan line filling algorithm to scan the line filling of the figure after offset and contraction according to the hatch line spacing[2][4]. By judging the parity and distribution of the intersection of each scan line and the boundary, it is determined whether the segment belongs to the internal segment of the area. For special cases that cannot be judged, we decide whether to fill the segment by judging whether the midpoint of the line is an internal point of the area. If yes, fill
it, otherwise, move to the next paragraph to judge and fill it. After scanning and filling the entire graphic, a zigzag hatching image with parallel directions can finally be obtained.

Figure 2. The original image of the single-layer contour pattern

2.2. Establishment of the Model
Let the color or grayscale of the boundary pixel point be $N_1$, filling grayscale is $N_2$, background value for $N_0$ (The total number of points in the scan lines when $d_j \neq 0$).

The steps of model establishment are as follows:

**Step 1:** We traverse each boundary point to find the highest point. Starting from the highest point, we scan from top to bottom, and use differential technology to extract the inner and outer endpoints of the boundary of the region on the scan line, and record the obtained X coordinates where $d_j \neq 0$ (The difference on the scan line $j$).

$$d_j = \int (i+1, j) - \int (i, j)$$  \hspace{1cm} (1)

$$x(k) = \begin{cases} i=1, d_j > 0 \\ i, d_j < 0 \end{cases} \quad (k = 1, 2, ..., N)$$  \hspace{1cm} (2)

$x(k)$ (X coordinate value when $d_j \neq 0$) Where $N$ is the total number of points on the scanning line where $d_j \neq 0$.

**Step 2:** When $N = 2$, go to Step 4; otherwise, go back to Step 2.

We pass and determine whether the corresponding straight line belongs to the inner segment of the boundary region, and then fill it with lines.

$$m = \text{mod}(N, 4)$$  \hspace{1cm} (3)

When $m = 0$, the line segment from point $4k + 2$ to point $4k + 3$ belongs to the inner segment of the region, and it is filled.

$$x_r = \frac{x(4k + 2) + x(4k + 3)}{2} \quad (r = 1, 2, ..., \frac{n}{4} - 1)$$  \hspace{1cm} (4)

When $m \neq 0$, That is, when the scan line is tangent to the local extremum of the boundary, the line segment from point $2k + 2$ to point $2k + 3$ may be an inner segment or may not belong to the inner segment of the region.
If \( \int (x, j - 1) \neq N_0 \) \hspace{1cm} (5)

Fill the section in a straight line, otherwise, transfer to the next section of the processing, repeat the equation.

Step 3: Check whether it has been filled. If it has been filled, it will be finished. Otherwise, return to Step 2.

In order to make the improved scanline filling model more clear and understandable, we designed the algorithm flow chart as shown in the Figure 3 below:

Figure 3. Algorithm flow chart
2.3. Contour Parallel Hatch Model

In the single-layer contour pattern, in order to achieve parallel hatching of contours, we need to fill the contours of the equidistant curve parallel to each other in Figure 1. We adopt the Bezier method based on the approximation principle to ensure that the hatching curves generated are parallel to each other and have the same spacing. Using the geometric composition method of the Bezier curve, the points on the Bezier curve corresponding to the given parameter value u can be determined.

The construction of the equidistant line requires the unit normal vector of the Bezier curve at each point. In order to find the unit normal vector at each point on the original curve, we use the relationship between the tangent vector $V^n_0V^{n-1}_1$ and the vector of the point $V^n_0$ in the Bezier curve to find each tangent vector $T(u)$ of the original curves. According to the tangent vector at the point, the unit normal vector $N_u$ of each point can be obtained according to the tangent vector, and finally get a more accurate equidistant curve of the original curve $P_d(u)$.

2.4. Establishment of the Model

Considering the realization of parallel hatches with contours, since the curves in the graph remain parallel at the same distance, we first use the de Casteljau algorithm [2] to obtain the tangent vector to each point on the Bezier curve.

First, the original curve is set

$$ p(u) = \{x(u), y(u)\} \quad u = [0, 1] \quad (6) $$

According to the execution process of the de Casteljau algorithm, it can be seen that the $V^n_0V^{n-1}_1$ is the tangent of the Bezier curve at point $V^n_0$, where the tangent vector of the Bezier curve is parallel to the vector and the direction is the same, let is

$$ T(u) = \frac{V^n_1(u) - V^{n-1}_0(u)}{\|T(u)\|} $$

Using formula (8), we can get the unit normal vector $N(u)$ of the parameter $u$ corresponding to each point on the original graph curve with $T(u)$.

$$ N(u) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} T(u) $$

Finally, the precise equidistant curve of the original curve is obtained according to formula (9).

$$ P_d(u) = P(u) + dN(u) \quad (9) $$

3. Results & Discussion

3.1. Zigzag Parallel hatch model solution

According to the above model, we designed relevant algorithms to obtain the hatched curves, the total length of the hatching lines and the time needed to realize the hatching algorithm. We run the program in zigzag parallel way in the case of parameter group (1) and (2) respectively and the result are shown in Table 1.

| Parameters group | The Total Length of Hatching lines(mm) | The number of Hatching lines(times) |
|------------------|----------------------------------------|-----------------------------------|
| (1)              | 947.5963                               | 93                                |
| (2)              | 10075.5005                             | 911                               |
We realized the hatched curves under two sets of parameters, the image of internal contraction of boundary distance 1mm, hatch line spacing 1mm (Parameter group one) is shown below in Figure 4.

![Figure 4. zigzag parallel hatch under Parameter group 1](image)

**3.2. Contour Parallel hatch model solution**

According to the hatching model with parallel contours, we processed the contracted figure, and finally obtained the filled image with parallel contours, and obtained the total length and winding number of the hatching curve under the hatching form through calculation.

| Parameters group | The Total Length of Hatching lines (mm) | The number of circles of contour parallel hatch |
|------------------|----------------------------------------|-----------------------------------------------|
| (1)              | 1082.2849                              | 5794                                          |
| (2)              | 9366.0217                              | 69051                                         |

![Figure 5 contour parallel hatch under Parameter group 1](image)
3.3. Comparison in the single-layer contour pattern

According to the two hatch models, we processed the original graphics, and finally got the zigzag and contour filled images, and calculated the total length of hatch curves under the two hatch forms and the level of the parallel incubation curve. The number of straight lines or the number of turns of the hatching curve with parallel contours.

We compare the two hatch models under two sets of parameters, the total length of the hatch curves and the elapsed time in the parallel hatch mode and the parallel hatch mode. Let indicate $Z_i(t)$, as the elapsed time of the zigzag parallel hatching algorithm under the parameter groups, and indicate $C_i(t)$ as the elapsed time of the contour parallel hatching algorithm under the parameter groups.

In order to compare the operating efficiency of the corresponding algorithms of the two models, we designed related algorithms to obtain the average elapsed time of the two hatch algorithms based on different parameter groups, (the group 1: Internal contraction of boundary distance 1mm and hatch line spacing 1mm; the group 2: Internal contraction of boundary distance 0.1mm and hatch line spacing 0.1mm) as shown in Table 3 and Table 4.

| Table 3. Relevant data of Parameters group 1 |
|--------------------------------------------|
| Zigzag Parallel Hatch Model | Contour Parallel Hatch Model |
| Average Elapsed Time(ms) | 0.016954 | 1.0937 |

| Table 4. Relevant data of Parameters group 2 |
|--------------------------------------------|
| Zigzag Parallel Hatch Model | Contour Parallel Hatch Model |
| Average Elapsed Time(ms) | 0.181549 | 8.1927 |

$\theta_i$ Represents the ratio of the average running time of the two algorithms under the specific group of parameters.

$$\bar{\theta}_i = \frac{Z_i(t)}{C_i(t)}$$

(10)

The bigger the $\bar{\theta}_i$, it means that under the $i$-th group of parameters, the zigzag parallel hatch algorithm has a lower running speed than the contour parallel hatch algorithm; otherwise, the higher the running efficiency.

The ratio of the average operation of the two algorithms under the two sets of parameters is obtained by formula (10).

$$\bar{\theta}_1 = 0.0155 \quad \bar{\theta}_2 = 0.0222$$

In summary, a comparative analysis of the operating results of the two hatch algorithms shows that the zigzag parallel hatch algorithm has the shortest length and the shortest average elapsed time, and its operating efficiency is the highest.

4. Conclusion

Based on the single-layer nested pattern, different models were built for the two incubation forms, and the incubation curve length and average running time were calculated for the two groups of different parameters.

It is found that the running rate of the directional parallel hatching algorithm is faster than that of the contour parallel hatching algorithm. The length of the hatching curve of the directional parallel algorithm is higher than that of the contour parallel algorithm.
Because the laser marking will produce a lot of contour data, the contour of any shape has a very high requirement on the operation efficiency of the program. On the basis of the original hatching algorithm, we improve and put forward a method that can effectively improve the operation rate.

Finally, the possible problems in the two forms of incubation were analyzed, and it was found that the hatching algorithm with parallel directions might generate aliasing phenomenon. We believed that anti-aliasing technology could be used to improve the grating resolution and other related technologies to solve the problem. For the hatching algorithm with parallel contours, the self-intersection of equidistant lines may occur, and we pass the Generating Algorithm and Improved Algorithm for accuracy.

References
[1] Ning, G. (2002) Image processing in laser marking. J. Laster Technology., 26: 295-317.
[2] Zhang, J. (1993) A new area parity check filling algorithm. J. Journal of Southwest China Normal University (Natural Science Edition)., 18: 288-291.
[3] Yan, L. (2006) The isometric line of the regular Bezier curve and its computer realization. J. Computer and Digital Engineering.,34: 129-131.
[4] Qiu, G. (2017) A region filling algorithm based on equally spaced parallel lines. J. Journal of Inner Mongolia Agricultural University (Natural Science Edition)., 38: 89-92.
[5] Yu, M. (2011) Polygon shape simplification and its quality evaluation. J. Mapping and spatial geographic information., 34:152-155.