

Lecture notes in computer science: research on hybrid grid algorithm of voronoi diagram

*Binjun Wang*¹, *Qiushi Wang*², *Jingying Li*¹

¹Department of Information Technology and Cybersecurity, People's Public Security University of China, Beijing: 100240, China

²Department of IOT, First Research Institute of the Ministry of Public Security of PRC, Beijing: 100048, China

{ Bin-jun Wang 1, Department of Information Technology and Cybersecurity, People's Public Security University of China, 100240. 2-3-602#, No. 29, Leyuan load, Daxing Dstrict, Beijing, China. wangbinjun@ppsuc.edu.cn

Abstract. Characteristics and performance of two grid methods, which include the growth method and the point by point scanning method of Voronoi diagram, are analyzed qualitatively and quantitatively. The theoretical analysis and experimental results show that the growth method is faster, but the boundary of Voronoi diagram generated by the point by point scanning method is more accurate. By combining the advantages of these two grid methods, the hybrid grid algorithm is proposed: the growth method is used firstly to extend a large step, and then the point by point scanning method is used to color the remaining blank pixel. The theoretical analysis and experimental results show that the speed of the hybrid grid algorithm is close to the fastest growth algorithm, and the accuracy is as well as the point by point scanning algorithm.

1 Introduction

Voronoi diagram [1, 2] is an important topic in computational geometry field, which is often used to solve problems of location optimization, space division and path planning in the fields of geography, meteorology and mobile communication. It has important theoretical and practical value.

The key issues of Voronoi diagram is the generating algorithm. At present, Voronoi diagram generating algorithms can be classified as the vector methods and grid methods. The vector methods have the advantage of high precision, but its data structure and calculation are more complex, and only applicable to the case of simple Voronoi diagram. The typical vector methods include incremental method [3], divided method [4] and indirect method [5]. The grid method has no limitation on the generator which can be a point, a line or a surface with different partitions and weights, but it is time-consuming and the result has lower accuracy. The typical grid methods are point by point scanning method [6] and growth method which includes distance table algorithm [7] and discrete construction algorithm [8].

The divisional weighted Voronoi diagram is the most complex situation when generator is a point, and the weighted Voronoi diagram is a special case when the diagram just has

one sector. And the normal Voronoi diagram is a special case when the generators in the weighted Voronoi diagram have the same weight. Therefore, the algorithm of divisional weighted Voronoi diagram also can be used to generate normal Voronoi and weighted voronoi diagrams. The existing grid methods of divisional weighted Voronoi diagram include growth method and point by point scanning method. In this paper, the advantages and disadvantages of two algorithms are analyzed. A hybrid grid algorithm of divisional weighted Voronoi diagram is proposed. The spatial complexity of the algorithm is equivalent to that of the two algorithms, and the time complexity is close to the fastest growth algorithm, but the resulting Voronoi diagram is as accurate as the point by point scanning algorithm.

2 The relative research and analysis

2.1 Qualitative analysis

The main idea of point by point scanning method: the weighted distance between any pixel on the plane and all sectors of all generators is calculated in turn, and the generator with the nearest weighted distance is the generator that the pixel belongs to. In reference [6], weighted Voronoi diagram is generated by point by point scanning method. This idea can also be used to generate the divisional weighted Voronoi diagram by calculating the nearest generator's sector. This algorithm needs to calculate the weighted distance between any pixel on the plane and all generators. Therefore, the time complexity is constrained by the number of generators and sectors.

However, the discrete construction algorithm proposed in reference [8] has the following problems:

Problem 1: The rough boundary problem. The formation of the boundary is approximated by the "collision" of the generator synchronization extension process, and the final resulted edge is related to the order of the generators expansion process. Fig.1 (a) and Fig.1 (b) are the simulation results of the same generator in different order, and there are some differences in the boundary. Fig. 1 (c) is a comparison result of the point by point scanning method. In fact, there are inherently imprecise of the growth method on the boundary, both of the existing algorithm [7, 8] have this boundary imprecision problem.

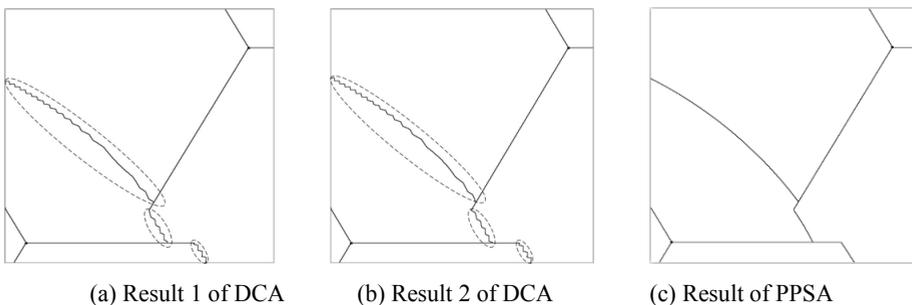


Figure.1 the results of the rough boundary problem

Problem 2: The angular increment problem. In order to scan and color pixels on extended arc with all radii, the original algorithm sets a sufficiently small and fixed angular increment in order to color the arc with large radius correctly, which means that the angular increment for small radius arcs and large radius arcs are the same. This results in the slow coloring speed of the original algorithm in the early (when the radius is small) stage. The improved method is to set the angular increment as a dynamic value that decreases as the

radius increases, see (2). It can satisfy the correctness of the algorithm with both small and large radius, and the efficiency of the algorithm is obviously higher.

Problem 3: The zebra stripe problem. It does not take into consideration of the large weight situation in original algorithm. For the generator with large weights, if the actual value of the radius (step) increment is greater than 1, it will result in some area that is not colored due to the large step. The principle is shown in Fig.2 (a). Fig.2 (b) is an example of the results, in which there are two generators with relatively large weights, one in the lower left corner and the other in the upper right corner. The result is an error.

On the other hand, when the weight of the generator is too small, the original algorithm has to compute the extended arc repeatedly, which affects the efficiency of the algorithm.

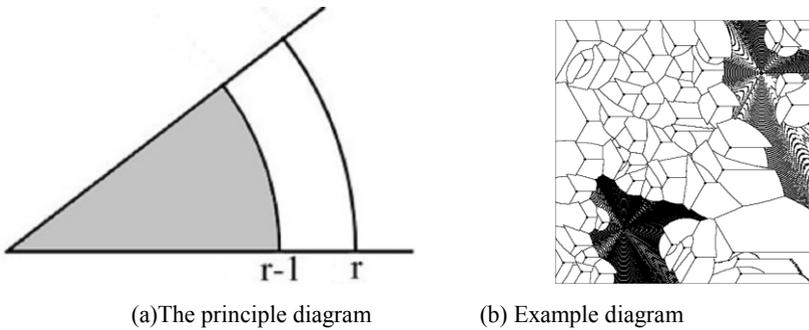


Figure 2. The zebra stripes problem

Problem 4: The discontinuous region problem. The more important problem of this algorithm is that it does not consider the case of discontinuous Voronoi diagram (property IV of reference [2]). The algorithm is only correct in the situation when the discontinuous region is absent in the Voronoi diagram! The problem is discussed in reference [9], but the termination condition is not analyzed in detail. In order to clarify the termination condition of the growth method, the theorem 1 is given firstly.

Theorem 1: For the growth method, during the algorithm process, if a certain pixel in the drawing area is first covered by a generator, the color of the pixel will not be changed during the following process.

Proof: (omitted).

Inference 1: For the growth method, when the maximum weight generator covers all pixels on the panel, the algorithm ends.

Proof: The general situation is proved firstly. For any of the generator, if all the pixels on the panel can be covered by its extension area, then the color of any pixel in the drawing area is determined by the extension process or previous algorithm process. According to Theorem 1, the algorithm ends.

Therefore, the conclusion is also true for the maximum generator.

Inference 2: For the growth method, the algorithm ends when all pixels of the drawing area are colored.

Proof: (omitted).

Inference 1 sets the termination condition of the algorithm from the perspective of the generator, while inference 2 sets the termination condition from the perspective of the panel. According to the inference 1, when the situation in Fig.3 occurs (the maximum weight generator is close to a corner and another relatively larger weight generator is close to another corner), although the panel has been completely covered and the color of each pixel on the panel is determined, the algorithm still needs to continue. Thus the efficiency is relatively low. According to inference 2, the algorithm would be more efficient with a counter. When the total numbers of pixels on the panel are all colored, the algorithm ends.

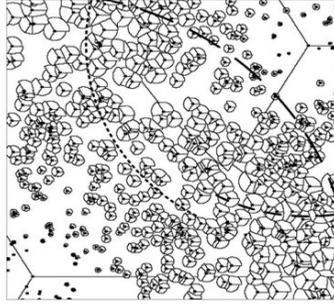
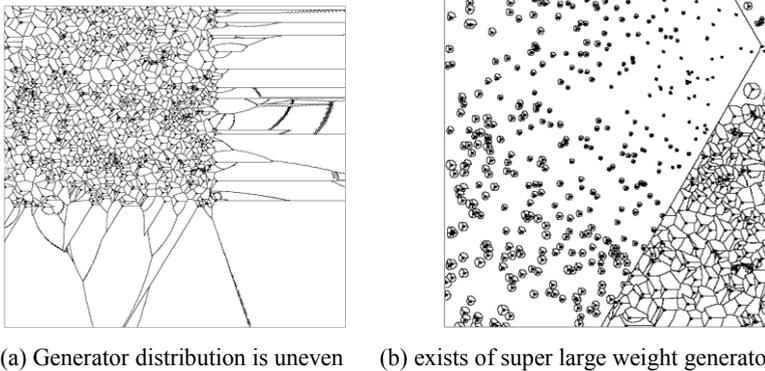


Figure 3. Cover the drawing area with the largest generator as the termination condition

It should be noted that the reference [7] also has problem 1 and 4. Problem 1 is the common problem of the growth method. Unless the parallel algorithm is used, any conventional algorithm of the growth method will have problem 1. Problem 4 is caused by the irrational design of the distance table algorithm [7] and the discrete construction algorithm [8].

In addition, according to the growth method, when the generator distribution is uneven or there is a super large weight generator, the boundaries of the majority of generators have been determined, but some individual generator expansion is not yet ended, because the algorithm needs to expand synchronously before it ends. It means that all generators still need to be calculated, which seriously affect the efficiency of the algorithm, as shown in Fig.4.



(a) Generator distribution is uneven (b) exists of super large weight generator

Figure 4. The influence problem of the growth method later steps

2.2 Quantitative analysis

Discrete construction algorithm [8] is that each pixel on the expansion arc of each generator is colored with the law of radius increment and the angle increment. Each pixel is calculated according to Eq. 1, in which (x_0, y_0) is the coordinate of the generator, r is the current expansion step of a sector of the generator, w is the weight of a sector of the generator, α is the initial angle, k is the scanned k th pixel on the extended arc, θ is the angular increment. It requires 6 times of multiplication to determine the color of each pixel in discrete construction algorithm. In particular, it also requires a sin function and a cos function as well, so the efficiency is low.

$$(x_0+w*r*\cos(\alpha+k*\theta), y_0+w*r*\sin(\alpha+k*\theta)) \tag{1}$$

In the original discrete construction algorithm [8], the angular increment is fixed. In order to meet the accuracy requirements with large radius, the fixed angular increment must

be small enough, these results in a very large repetitive workload to calculate the extended arc when the radius is small. In fact, the angular increment should be the corresponding angle of the adjacent two pixels on the extended arc. The minimum distance between two pixels on the panel is 1, and the corresponding minor arc will be slightly larger. According to the calculation relationship of radius, minor arc and angle, the angular increment should be calculated with Eq. 2 which can satisfy the correctness of the algorithm, and the efficiency of the algorithm is improved as well.

$$\Theta = 1/r \tag{2}$$

Correspondingly, according to the point by point scanning method, if there are n generators, to determine the color of any pixel on the space, it need to calculate the weighted distance between (x, y) and all the generator (xi, yi) in order to get the minimum weighted distance, as shown in Eq. 3. Where W_i (i=1, 2, ..., k) is the weight of each generator (divisional weighted Voronoi diagram is similar to this). The point by point scanning method requires k times of 2 times multiplications, 1 times division and 1times square root to determine the color of each pixel.

$$\min \{ \sqrt{(x_i - x)^2 + (y_i - y)^2} / w_i \mid i=1,2, \dots, n \} \tag{3}$$

We estimate the calculation amount of Eq. 1 and Eq. 3. Eq. 4 and Eq. 5 are Taylor's expansion of sin and sqrt functions respectively.

$$\sin x = \sum_{i=0}^n \frac{(-1)^i}{(2i+1)!} x^{2i+1} + \sin^{(2n+3)}(\xi) \frac{x^{2n+3}}{(2n+3)!} \tag{4}$$

$$\sqrt{x} = 1 + \sum_{i=2}^n \frac{(-1)^{i-1}}{i!} \frac{(2n-3)!}{(n-2)!} (x-1)^n + \frac{(-1)^n}{(n+1)!} \frac{(2n-1)!}{(n-1)!} \xi^{\frac{1}{2}-n} (x-1)^n \tag{5}$$

If the accuracy is required as 10⁻⁹, Lagrange remainder of Eq. 4

is $\left| \sin^{(2n+3)}(\xi) \frac{x^{2n+3}}{(2n+3)!} \right| \leq 10^{-9}$, Consider extreme conditions $\left| \frac{(\frac{\pi}{2})^{2n+3}}{(2n+3)!} \right| \leq 10^{-9}$, we could get n=5. In other words, if Taylor expansion of sin function iterates 5 times, its accuracy can reach 10⁻⁹. At this point, the total calculation amount of multiplication and division in Eq. 4 is 1+4*n, which is 21. The calculation amount of cos function is the same as sin function, which is 21,

too. Lagrange remainder of Eq. 5 is $\left| \frac{(-1)^n}{(n+1)!} \frac{(2n-1)!}{(n-1)!} \xi^{\frac{1}{2}-n} (x-1)^n \right| \leq 10^{-9}$, consider extreme conditions $\left| \frac{1}{(n+1)!} \frac{(2n-1)!}{(n-1)!} 2^{\frac{1}{2}-n} \right| \leq 10^{-9}$, we could get n=8. At this point, the total calculation amount of multiplication and division in Eq. 5 is 1+7n, which are 57.

Therefore, according to Eq. 1 based on the discrete construction algorithm, it needs 48 times of multiplication and division to determine the color of a pixel. According to Eq. 3 based on the point by point scanning method, it needs 60*k times of multiplication and division to determine the color of a pixel. So, on the whole, the efficiency of the discrete construction algorithm is better than the point by point scanning method.

3 Hybrid algorithm design

Based on the above analysis, it can be seen that the point by point scanning method is more accurate, and the time complexity of the algorithm is not related to the distribution of the generator. The boundary of the growth method is inherently imprecise, and the algorithm has high computational efficiency at early stage while it has low computational efficiency at later stage. In general, the efficiency of discrete construction algorithm is theoretically superior to point by point scanning method, but the discrete construction algorithm becomes slower, especially in the late stage, when the generator distribution is uneven or there is a super large weight generator.

Therefore, the idea of the hybrid algorithm we proposed is: firstly, according to the basic idea of discrete construction algorithm, all generators are preprocessed simultaneously to extend a large step (proportional to the weight of each generator) to the situation where most of the generators are in contact with each other, and the sector filling algorithm we proposed is used to fill the sectors of a generators efficiently, in which the overlapping pixel is set into black color. Secondly, turn all the black pixels in the first step into white color. Thirdly, the remaining white pixels are filling color by point by point scanning method. Finally, the boundary of the Voronoi diagram is extracted. The hybrid algorithm not only guarantees the efficiency, but also improves the accuracy of the Voronoi diagram.

3.1 The data structure design

The major data structure of the algorithm is the data structure of the generator. Each generator consists of two-dimensional coordinates, the weight, the angle, the color of each sector, and so on. Among them, the angular range is $[0, 360]$, the weight range is $[0, \infty]$.

```
struct Generator
{
    public int x;           //x coordinate
    public int y;           //y coordinate
    public float w[];       //Sector weight array
    public int a[];         //Sector angle array
    public int RGB[][3];    //Sector color array
} AllGen[];
```

3.2 The algorithm design

Algorithm 1: hybrid grid algorithm of Voronoi diagram

```
S1: //Initialization stage
All the pixels on the panel are set to white.
S2: // Preprocessing stage
S2.1: set the value of k. //reference Eq. 6
S2.2: for(each sector j of each generator i){
S2.2.1:  $R = k * \text{AllGen}[i] \rightarrow w[j]$ ; //calculate radius
S2.2.2: algorithm2( $\text{AllGen}[i] \rightarrow x$ ,  $\text{AllGen}[i] \rightarrow y$ ,  $\text{AllGen}[i] \rightarrow a[j]$ ,  $\text{AllGen}[i] \rightarrow a[j+1]$ , R,
AllGen[i]. RGB);
}
S3: //deal with overlap stage
turn all blank pixels to white;
S4: // Calculate residue pixels stage
for (all pixel on the panel){
```

```
    if (the pixel is white){  
        according to the point by point scanning method, calculate the color of pixel, and set the  
        pixel with the color;  
    }  
}
```

S5: //calculate the boundary stage

Scanning the panel, and extracting boundaries of the divisional weighted Voronoi diagram.

S6: end.

Algorithm 2 is used to color a sector of a generator. Since the initial coverage area of each generator may overlap, the algorithm will set the pixel to black if it is in the overlapped area during the progressive scanning and filling. Filling algorithm is relatively mature, such as the seed algorithm [10] and the progressive scan algorithm [11], and so on. A sector colored with filling algorithm is more special, we proposed a more efficient algorithm to do so.

Algorithm 2: (omitted).

4 Hybrid algorithm performance analysis and correctness proof

4.1 The performance analysis of the algorithm

The preprocessing stage of algorithm 1 calculates the color of each sector of each generator by a large step with weight r synchronously using algorithm 2 in which the main calculation is the boundary of a sector. Obviously, from the macroscopic point of view, this algorithm is faster than the discrete construction algorithm which calculates the extended arc pixel by pixel, and also faster than the point by point scanning method.

In addition, due to the use of the point by point scanning method in the later stage of hybrid grid algorithm, it solves the repetitive calculation problem in the later stage of growth method when the generator distribution is uneven or there is a super large weight generator, and the inherently imprecise of growth method (problem 1) are solved as well.

One of the key issues affecting the efficiency of this algorithm is the design of the initial step size. If the initial step is too large, the overlap area in preprocessing will be large, which means requiring more recalculation of the overlapping area color by the point by point scanning method; On the other hand, if the initial step is too small, there will be too many blank areas, the advantages of the preprocessing stage of this algorithm will be loosed. Ideally, the coverage area of all sector of each generator is just a balance between the blank area and the overlap area.

Without loss of the generality, we assume that the generators are equidistribution (Uneven distribution means that there are more generators in some space, and less generators in the other, which does not affect the correctness of the algorithm). Assuming that the total area is m , then the maximum value of k which meet Eq. 6 is basically the requirement for first step in algorithm 1.

错误!未找到引用源。

(6)

4.2 The correctness proof of the algorithm

In the preprocessing stage, firstly, each sector of generators extends in a large step, according to the principle of discrete construction algorithm, the area which is not overlapped has already determines the color, the area which is not yet processed will be colored according to point by point scanning method in late stage, in which the accuracy of the hybrid grid algorithm is the same as the point by point scanning method.

How about the correctness of the overlapped area? Because the algorithm 2 sets the overlapped area to black, and S3 in algorithm 1 replace all the black pixels to white. In S4 of the algorithm 1, the color of each pixel in the overlapped area is determined by the point by point scanning method. So, the accuracy of the overlapped area is the same as the point by point scanning method.

5 The experimental results and analysis

The experiment environment is the software of Visual Studio 2010 C#, the hardware of Intel ®Core (TM) i5 with CPU 2.4GHz and 2.0GB RAM. We reproduce the original discrete construction algorithm [6] (ODCA) and point by point scanning method [7] (PPSM) respectively, and implement the improved discrete construction algorithm (IDCA) and the proposed hybrid grid algorithm (HGA).

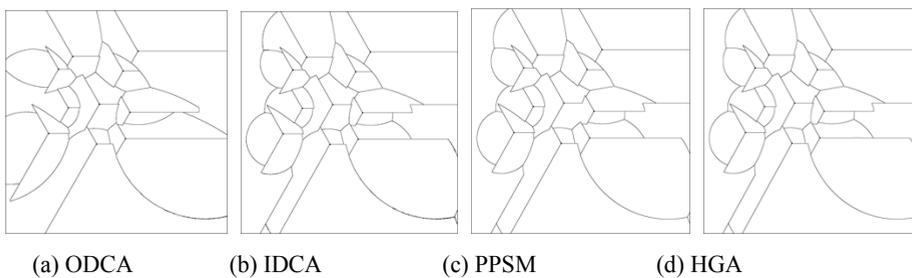


Figure 5. The result contrast of four algorithms

Fig.5 is the result of the ODCA, IDCA, PPSM and HGA with $800 * 800$ grid size and 10 generators.

Table 1 the time efficiency of these methods with different panel size and number of generators.

panel size	number of generators	ODCA	IDCA	PPSM	HGA
300*300	10	0.9	2.4	1	1.2
	50	8.8	4.6	1.9	1.6
	100	11.9	4.6	3.4	2.1
	200	16.3	2.4	6.4	3.1
	500	26.8	2.5	8.5	7.3
500*500	10	9.8	4.1	2.8	3.2
	50	25.1	11.3	5.3	4.5
	100	32.5	6.6	7.2	5.8
	200	44.9	7.3	15.1	8.5
	500	71.2	11.4	24.9	17.2
800*800	10	27.3	9.8	7.6	8.2
	50	57.8	15.4	10.8	11.1
	100	88.6	23.3	16.5	15.5
	200	113.2	17.8	27.6	21.1
	500	178.9	20.6	58.9	42.7

It can be seen from the experimental results in Fig.5, Table 1 and Fig.5 that the original discrete construction algorithm is not correct and the efficiency is not high. The improved discrete construction algorithm is ideal implementation of the growth method, in which the result is correct and algorithm is more efficient, but the result of Voronoi diagram boundary is not accurate. The hybrid grid algorithm is faster than the point by point scanning method, and quite the same as the improved discrete construction algorithm. The trend is that the improved discrete construction algorithm is better.

When those algorithms are in application, if the accuracy requirement for Voronoi diagram boundary is not high, the improved discrete construction algorithm can be used, otherwise, the hybrid grid algorithm is a better choice.

6 Summary

In this paper, the properties and performance of growth method and point by point scanning method are analyzed from qualitative and quantitative aspects. A hybrid grid algorithm is proposed, which combines the advantages of growth method and point by point scanning method. At the same time, a special algorithm is proposed to improve the efficiency of filling the sector color of generators, which further improves the generation efficiency of the hybrid grid algorithm while processing divisional weighted Voronoi diagram.

The problems need further research is as follows: Firstly, the determination of the initial step k can be further refined. Secondly, the overlapped area determined color according to the point by point scanning method only related to the generators that overlaps each other. Thirdly, after a color of a pixel is defined by point by point scanning method, all pixels on the line between the pixel and a generator maybe can easily calculate. These studies can further improve the performance of the hybrid grid algorithm.

References

1. Voronoi G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques

- positives parfaites[J]. *Journal für die reine und angewandte Mathematik*, 133, 97-178(1908).
2. B.N. Boots. Weighting thiesen polygons. *Economic Geography*, 56(3), 248-259(1980).
 3. Ohya T. Improvements of the Incremental Method for the Voronoi Diagram with Computational Comparison of Various Algorithms [J]. *Journal of the Operations Research Society of Japan*, 27(4), 69-97(1984).
 4. Shamos M I. and Hoey D. Closest-point Problems, In: *Proceedings of the 16th Annual Symposium on the Foundations of Computer Science*, 151-162(1975).
 5. Fortune S. Voronoi diagrams and Delaunay triangulations [J]. *Computing in Euclidean geometry*, 1, 193-233(1992).
 6. Youhui Zhang. *A Study on the Weighted Voronoi Diagrams* [D]. Beijing University of Aeronautics and Astronautics, 1998.
 7. Watanabe T, Murashima S. A Method to Construct a Voronoi Diagram on 2-D Digitized Space in O (1)Computing Time[J]. *Transactions of the Institute of Electronics Information & Communication Engineers*, 79(1996).
 8. iling Ma. *The Construction of Divisional Weighted Voronoi Diagrams and Count of Area* [D]. Hebei Normal University, 2004.
 9. Jingying Li, Binjun Wang. A Study on the Correctness of Discrete Construction Method [J]. *Journal of People's Public Security University of China (Science and Technology)*, 22(3), 94-97(2016)
 10. Lieberman H. How to Color A Coloring Book. *computer graphics*. 12, 111-116(1978)
 11. Theo Pavlidis. Filling Algorithms for Raser Graphics [J]. *computer graphics and image processing*, 10, 126-141(1979)