OPTICAL FILLING ALGORITHM

(Preliminary version)

Y. B. KARASIK
Computer Science Department, Tel Aviv University, Tel Aviv, Israel
e-mail: karasik@math.tau.ac.il

Abstract

The technique of optical computing is applied to develop a new filling algorithm which fills the interior of a polygon in constant time whereas conventional filling algorithms have at least $\Omega(n \log N)$ time complexity, where n is the number of boundary pixels of a polygon, and N is the diameter of the polygon expressed in terms of pixels.

1 Introduction

The filling of the interior of a polygon is an important problem in image processing, pattern recognition and computer graphics. This problem has received much attention from computer scientists who have proposed many solutions intended for ordinary display devices [10], [8], the best of which [1], as far as we know, has $O(n \log N)$ time complexity, where n is the number of pixels contained in the boundary and N is the number of pixels contained in the diameter of a polygon.

The same problem also emerges in optical image processing where images are represented on special optical devices called spatial light modulators (SLM) in the form of distributions of transmittance (or reflection) coefficients of their pixels [5].

The main advantage of SLM, in comparison with usual display devices, is that we can control the transmittance coefficient of light both optically and electronically. The first way is preferable because optically we can control all pixels in parallel.

Optical control of the transmittance coefficient proceeds as follows. Let $A(i,j)$ be the amplitude of an incident plane lightwave at pixel (i,j). If the SLM is in control mode, then the transmittance coefficient of the pixel becomes $A(i,j)$ (in working mode SLM simply transmits incident light through each pixel relaxing its amplitude proportionally to the optical density of the pixel).

Obviously, two plane waves $A(n,m) \cdot e^{i(\omega t+const)}$ and $B(n,m) \cdot e^{i(\omega t+const)}$ can be added to yield a new wave $(A(n,m) + B(n,m)) \cdot e^{i(\omega t+const)}$. Thus, optically, we can perform

• **addition:** $C(n,m) = A(n,m) + B(n,m)$ of two images in parallel for every n, m.

Analogously, in parallel for each pixel we can also perform the following operations:

• **subtraction** [7]: $C(n,m) = A(n,m) - B(n,m)$;

• **scalar multiplication**: $C(n,m) = k \cdot B(n,m)$ which is obtained as a result of the passage of light through both the SLM containing $B(n,m)$ and the SLM whose pixels have optical density k. We assume that photodetectors cannot distinguish between two electromagnetic waves if their amplitudes differ by less than unit under appropriate scaling. Thus, elements of the matrices we deal with are positive integers and scalar multiplication of an image is, in fact, the operation: $C(n,m) = \lfloor k \cdot B(n,m) \rfloor$.

• **filtering of an image at a given level** L [12]:

$$C(n,m) = Filter_L(B(n,m)) = \begin{cases} 1, & \text{if } B(n,m) \geq L; \\ 0, & \text{otherwise;} \end{cases}$$

• **convolution** [11]:

$$C(n,m) = \sum_{i=1}^{N} \sum_{j=1}^{M} A(n-i, m-j) \cdot B(i,j) = A(n,m) * B(n,m);$$

• **correlation** [9]:

$$C(n,m) = \sum_{i=1}^{N} \sum_{j=1}^{M} A(i+n, j+m) \cdot B(i,j) = A(n,m) \odot B(n,m);$$

Convolution and correlation can serve as the basis for a variety of more complicated operations on 2-dimensional images. For example, if

$$B(n,m) = \begin{cases} 1, & \text{if } n = p, m = q \\ 0, & \text{otherwise,} \end{cases}$$

then $C(n,m) = A(n,m) * B(n,m) = A(n-p, m-q)$ is the image $A(n,m)$ shifted by the vector (p,q). Since, convolution is performed optically in constant time, we can translate any image by a vector also in constant time. Moreover, if

$$B(n,m) = \begin{cases} 1, & \text{if } (n,m) \in \{(p_k, q_k)\}_{k=1}^{M} \\ 0, & \text{otherwise,} \end{cases}$$

then

$$C(n,m) = \sum_{k=1}^{M} A(n - p_k, m - q_k).$$

Thus, we can place several copies of image A at given positions simultaneously in constant time.

Now let A and B be binary images and let B consist of L pixels. Then

$$C(n,m) = A(n,m) \odot B(n,m) = \begin{cases} L, & \text{if } B_{nm} \subseteq A \\ l < L, & \text{otherwise,} \end{cases}$$

where B_{nm} is the copy of B moved to the pixel (n, m).

Thus, having performed filtering of the image C at level L we can recognize all occurrences of image B in image A and determine locations of these occurrences.

Suppose now we need to substitute all occurrences of image B in image A by another image, B_1. This can be performed in constant time as follows:

Step 1. Compute correlation $C_1 = A \odot B$ to determine all occurrences of B in A.

Step 2. Perform filtering $C_2 = Filter_L(C_1)$ to determine the locations of these occurrences.

Step 3. Perform convolution $C_3 = B * C_2$, which gives the union of all occurrences of B in A.

Step 4. Compute $C_4 = A - C_3$ to determine those pixels of A which are not contained in C_3.

Step 5. Compute $C_5 = B_1 * C_2$ to substitute all occurrences of B by B_1.

Step 6. Compute the final result $C = C_5 + C_4$.

The technique described is called symbolic substitution (SS) [3].

We see that optical computing provides a variety of very complicated operations on images which can be performed in constant time. Therefore, based on these operations, we can try to reduce time-complexity of the filling. The previous attempt to achieve this objective undertaken in [6] gave an optical filling algorithm having only $O(N)$ time-complexity, where N is the number of vertices of a polygon to be filled. This paper aims to propose a new filling algorithm having $O(1)$ time-complexity.

2 An algorithm for reconstruction of any plane image from its refined boundary

To obtain the idea of the algorithm turn from a discrete plane consisting of pixels to a continuous plane consisting of points and consider any horizontal line $y = a$ intersecting the plane figure.

What segments of this line lie in A? To answer the question we should assign a weight to each intersection point $(x_i(a), a)$ by the following recurrence formulae:

$$weight(x_i(a), a) = \sum_{k=1}^{i-1} weight(x_k(a), a) + \begin{cases} 1, & \text{if the } i^{th} \text{ intersection is of} \\ & \text{the first type (see fig. 1 a, b);} \\ \frac{1}{2}, & \text{if the } i^{th} \text{ intersection is of} \\ & \text{the second type (see fig. 1 c, d);} \\ -\frac{1}{2}, & \text{if the } i^{th} \text{ intersection is of} \\ & \text{the third type (see fig. 1 e, f);} \\ 0, & \text{if the } i^{th} \text{ intersection is of} \\ & \text{the fourth type (see fig. 1 g, h),} \end{cases}$$

where i is the number of intersection in order from left to right.

It is easy to see that each segment of the line whose left endpoint has an odd weight, lies in A.

We can also attribute a weight to other points (x, y) by the following formulae:

$$weight(x,y) = \sum_{x_i(y)<x} weight(x_i(y), y).$$

Obviously, only points with an odd weight lie in A.

Now attribute a type to each point of ∂A as follows: the type of a boundary point is the type of the intersection between ∂A and the horizontal line passing through this point. Since interior points of horizontal boundary segments cannot be intersection points, we attribute to them the fourth type. Thus, we have:

$$\partial A = \bigcup_{i=1}^{4} S_i,$$

where S_i is the set of boundary points of the i^{th} type.

Theorem 2.1

$$weight(x,y) = S_1 * ray(x,y) + S_2 * \frac{1}{2} \cdot ray(x,y) + S_3 * -(\frac{1}{2}) \cdot ray(x,y),$$

where

$$ray(x,y) = \begin{cases} 1, & \textit{if } x \geq 0, y = 0; \\ 0, & \textit{otherwise,} \end{cases}$$

and convolution between set S and any function $f(x,y)$ is defined as follows:

$$S * f(x,y) = \sum_{(p,q)\in S} f(x-p, y-q).$$

We omit the proof in this version.

Let us now turn from the plane consisting of points to the plane consisting of pixels and generalize the approach stated above.

Instead of the function $ray(x, y)$ consider the image

$$RAY(n,m) = \begin{cases} 1, & \text{if } m = 1, n \geq 1; \\ 0, & \text{otherwise.} \end{cases}$$

Instead of boundary points of the i^{th} type consider boundary pixels of the i^{th} type (see fig. 2). We assume here that each boundary pixel is adjacent to exactly two others because those boundary pixels which do not satisfy this property can be removed from the image without breaking its topology (we omit the proof of this assertion). The removal is performed in constant time using technique of SS (we omit details in this version).

Let S_i be now a set of boundary pixels of the i^{th} type and

$$WEIGHT(n,m) = S_1 * 2 \cdot RAY(n,m) + S_2 * RAY(n,m) - S_3 * RAY(n,m).$$

It is easy to see that the interior of A consists only of those pixels (n,m) which have $WEIGHT(n,m)$ not proportional to 4.

Hence, we can propose the following filling algorithm.

Step 1. Remove redundant boundary pixels which are not of type 1, 2, 3 or 4.

Step 2. For each $i = \overline{1,3}$ recognize boundary pixels of the i^{th} type . For this aim correlate the boundary with masks depicted on fig. 2 respectively and filter the resulting images at level 3. As a result we obtain images S_1, S_2 and S_3.

Step 3. Compute

$$WEIGHT(n,m) = S_1 * 2 \cdot RAY(n,m) + S_2 * RAY(n,m) - S_3 * RAY(n,m).$$

Step 4. Compute

$$W(n,m) = \lfloor WEIGHT(n,m)/4 \rfloor.$$

Step 5. Compute

$$INTERIOR(n,m) = WEIGHT(n,m) - 4 \cdot W(n,m) = \begin{cases} 2, & \text{if the pixel } (n,m) \text{ is interior;} \\ 0, & \text{otherwise.} \end{cases}$$

Since each step of the algorithm requires constant time for its performance, we obtain:

Theorem 2.2 *The interior of a plane image can be filled optically in constant time.*

References

[1] H. H. Atkinson, I. Gargantini, and T. R. S. Walsh, Filling by quadrants or octants, *Computer vision, graphics, and image processing*, vol. 33, pp. 138–155 (1986).

[2] H. J. Caulfield, M. A. G. Abushagur, Hybrid analog-digital linear algebra processors, *Proc. of SPIE*, vol. 634, pp. 86–95 (1986).

[3] S. D. Goodman and W. T. Rhodes, Symbolic substitution applications to image processing, *Applied Optics,* 27, pp. 1708–1714 (1988).

[4] D. G. Feitelson, *Optical Computing: A survey for computer scientists,* MIT Press, Cambridge, Massachusetts, London, England, 1988.

[5] A. D. Fisher, J. N. Lee, The current status of two-dimensional spatial light modulator technology, *Proc. of SPIE,* vol. 634, pp. 352–371 (1986).

[6] Y. B. Karasik, M. Sharir, Optical computational geometry, *Proc. 8th ACM Symposium on Computational Geometry,* 1992 (to appear).

[7] E. Marom, Real-time image subtraction using a liquid crystal light valve, *Optical Engineering,* 25(2), pp. 274–276 (1986).

[8] T. Pavlidis, Filling algorithms for raster graphics, *Computer graphics and image processing*, vol. 10, pp. 126–141 (1979).

[9] W. T. Rhodes, Optical signal processing: Fourier Transform and Convolution/Correlation, *Proc. of SPIE,* vol. 634, pp. 57–63 (1986).

[10] U. Shani, Filling regions in binary raster images: a graph-theoretical approach, *Computer graphics*, vol. 14, pp. 321–327 (1980).

[11] A. B. VanderLugt, Signal detection by complex spatial filtering, *IEEE transactions, Information Theory,* IT-10(2), April 1964, pp. 139–145.

[12] C. Warde and J. Thackara, Operating modes of the microchannel spatial light modulator, *Optical Engineering,* 22(6), pp. 695–703 (1983).

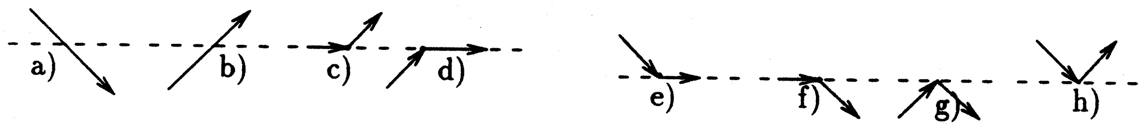

Figure 1: four types of intersections; directions of boundary segments are shown by arrows, points of intersections are indicated as rings.

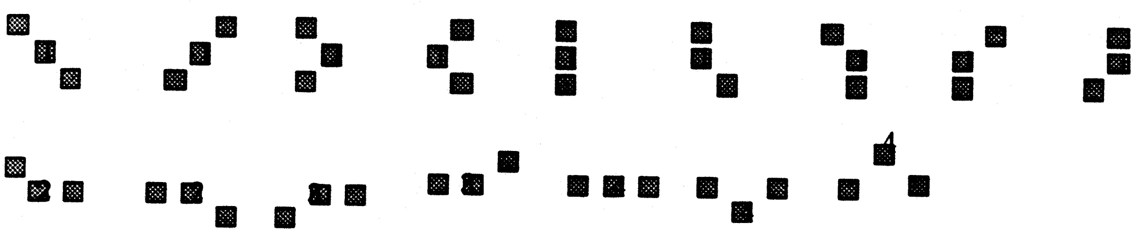

Figure 2: boundary pixels of the first, second, third and fourth types marked as 1, 2, 3 and 4 respectively.