ELSEVIER

Pattern Recognition Letters 19 (1998) 1111–1117

Pattern Recognition Letters

Hexadecagonal region growing

Enrique Coiras *, Javier Santamaría, Carlos Miravet

SENER, Ingenieria y Sistemas. Severo Ochoa, 4. Tres Cantos, PTM 28760, Madrid, Spain

Received 7 April 1998; received in revised form 19 June 1998

Abstract

In spite of their non-isotropic nature, local growths based on 4-connected or 8-connected neighborhoods are frequently used for binary image dilation, mostly because they are very easy to implement. Although better dilation techniques exist, they usually require bigger masks, more complex routines, exotical grid definitions or globally processing the image. In this article we propose a new 3×3 implementation of the hexadecagonal model for a better aproximation of isotropic growth on a square grid that does not have the problems of its traditional incarnations.

Keywords: Growth process; Digital image processing; Dilation; Erosion; Voronoi diagram; Distance mapping; Region growing

1. Introduction

Isotropic growth modelling for digital image processing has many applications, such as growth pattern determination of cells, calculation of the area of influence of diverse elements present on images, medial axis extraction, generation of Voronoi diagrams, etc. However, the difficulty to develop an algorithm easy to implement and with little processing requirements has forced the use of growth models quite far from the isotropic behaviour.

Binary shape growing methods have usually been restricted to 3×3 morphological dilation, using either 8-connected or 4-connected neighborhoods (Rosenfeld and Pfaltz, 1968). As image dilation and erosion operations are usually only applied a small number of iterations to an image, the anisotropic nature of these 8- and 4-neighbor growth models is not very critical. But it is also true that this anisotropy prevents a wider use of these models.

Other deterministic growth models, such as diffusion (Thompson and Rosenfeld, 1995), create approximately isotropic growth patterns, but are difficult to implement and the discretization of the diffusion equations can lead the algorithm to unwanted stationary states. Chamfer 3/4 distance calculation (Borgefors, 1986) can also be used for deterministic dilation of binary images, computing the distance map and then filling those pixels nearer than a certain threshold. This technique relies on local 3×3 region analysis and generates non-regular octagonal growth patterns, that approximate better the circular shape than the 4- and 8-neighbors masks. However, it is not a local technique in the sense that the whole image has to be processed to simulate the dilation. Chamfer 5/7/11 distance calculation (Borgefors, 1986) gets better results, but depends on the analysis of 5×5

* Corresponding author. E-mail: enrique.coiras@mad.sener.es.

0167-8655/98/$ – see front matter
PII: S0167-8655(98)00091-9

local regions. Other global techniques (Serra, 1982) allow for hexagonal and dodecagonal approximations to circular growth, but can only operate on hexagonal grids.

Time-delays for 8-neighbor growth (Thompson and Rosenfeld, 1997), are complex models that can generate circular (and many other) shapes, but only after a fixed number of iterations. The intermediate stages, though, are not circular. Some other procedures provide euclidean distance determination (Danielsson, 1980), but require extra memory for distance computations or minimum path storage, and more complex calculations.

In (Kulpa and Kruse, 1983) several algorithms for circular propagation are proposed. Empirical hexadecagonal growth results are presented, but no specific model is proposed. Also, the corner rounding technique used for its implementation prevents its application to figure propagation.

In this article we present a detailed analysis of the hexadecagonal growth pattern, and a new simpler implementation of its growth algorithm. For completion, a study of the regular octagonal growth model and a comparison to the other exisiting 3×3 local dilation models, as well as the 5×5 Chamfer 5/7/11, are included.

2. Regular octagonal growth

The iterated application of the 8-n or 4-n dilation algorithms to an isolated pixel of a binary image, generates either a square, in the case of the 8-neighbor growth, or a diamond in the case of the 4-neighbor (Fig. 1).

In effect, starting the growth process from an isolated pixel at the origin, $o = (0, 0)$, after n iterations, the set of pixels contained in the dilated shape, for the 4-n model, is

$$C_4(n) = \{(x, y) \in \mathbb{Z}^2 : |x| + |y| \leqslant n\}, \tag{1}$$

where x and y are integers and $\mathbb{Z}$ is the set of integers. The resulting shape is a diamond with vertices at $(n, 0)$, $(0, n)$, $(-n, 0)$ and $(0, -n)$.

In the case of the 8-n growth, the set is

$$C_8(n) = \{(x, y) \in \mathbb{Z}^2 : |x| \leqslant n, |y| \leqslant n\}, \tag{2}$$

which is a square with vertices at (n, n), $(-n, n)$, $(-n, -n)$ and $(n, -n)$.

A better approximation to the isotropic model, which grows as an irregular octagon (Fig. 1(c)), can be easily obtained by the combination of 4-n and 8-n models, alternatively applying one or the other (Rosenfeld and Pfaltz, 1968). And, since the area of $C_4(1)$ is a better aproximation to the area of a circle of radius 1 than the area of $C_8(1)$, 4-n growth should be applied first (Das, 1992). After n iterations the following set results:

$$C_{4,8}(n) = \left\{(x, y) \in \mathbb{Z}^2 : |x| \leqslant n, |y| \leqslant n, \; |x| + |y| \leqslant \left\lfloor \frac{3}{2} n \right\rfloor \right\}, \tag{3}$$

where $\lfloor x \rfloor$ means the nearest integer not exceeding x.

The dimensions of the resulting octagon, though, depend only on the number of times that each growth model is applied, but not on their order of application, as noted in (Kulpa and Kruse, 1983). Thus, if 4-n and 8-n growths are

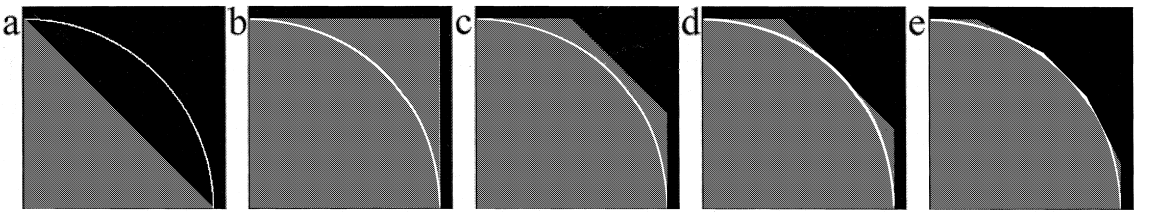

Fig. 1. Application of different growth models to an isolated pixel placed at the bottom-left corner of the image after 255 iterations, and the ideal circular arc with a radius of 255 pixels. (a) 4-neighbor, (b) 8-neighbor, (c) octagonal growth (iterated 4-n, 8-n), (d) regular octagonal growth and (e) proposed hexadecagonal growth. Only first quadrants are shown.

applied n_4 and n_8 times, respectively, the set of points belonging to the dilated shape are

$$C_{4,8}(n_4, n_8) = \{(x, y) \in \mathbb{Z}^2 : |x| \leqslant n, |y| \leqslant n, \\ |x| + |y| \leqslant n_4 + 2n_8\}, \quad (4)$$

where $n = n_4 + n_8$ is the total number of iterations. The vertices of this octagon are $(n, \pm n_8)$, $(n_8, \pm n)$, $(-n_8, \pm n)$ and $(-n, \pm n_8)$. Note that for $n_4 = 0$, the octagon degenerates into a square, and for $n_8 = 0$ into a diamond, in concordance with 8-*n* and 4-*n* models.

This octagonal approximation can be improved making all the edges tangent to the circumference, generating a regular octagon. To achieve this, an equillibrium between 8-*n* and 4-*n* growths must be reached.

In the case of a regular octagon, circumscribed to a circle of radius n, the position of the rightmost upper vertex is $(n, n(\sqrt{2} - 1))$. For the $C'_{4,8}$ octagon that vertex is at (n, n_8), so, for it to be regular, $n_8 = n(\sqrt{2} - 1)$ or $n_8/n = \sqrt{2} - 1$.

Then, in order to decide whether 4-*n* or 8-*n* growth should be applied, the fraction n_8/n has to be evaluated at every iteration of the algorithm. If it is greater than $\sqrt{2} - 1$, 4-*n* growth is applied; on the other case 8-*n* is used. And, as noted before, 4-*n* growth should be used for the first iteration.

For a small number of iterations, the value of n_8/n can be approximated by

$$\frac{n_8}{n} = \sqrt{2} - 1 \approx \frac{1}{2} - \frac{1}{12}. \quad (5)$$

Thus, applying 8-*n* growth just for the iterations that are multiples of 2 but not of 12, a regular octagon will be obtained. This control rule is valid for circumferences of at most 410 pixels of radius. Better approximations can be achieved by adding terms to the series (the next term is $-1/410$).

The results of this control rule for the octagonal growth method can be seen in Fig. 1(d).

3. Hexadecagonal growth model

The vertices of the regular octagon are the pixels that deviate most from the ideal circumference. It would be very useful, then, to have a way to control the "growing speed" in the directions of these vertices.

It should be noted that, from the third iteration of the algorithm, every point in the octagon boundary can be classified by its 3×3 neighborhood in one of the two following types.

- *Edge pixels*: Five of their neighbors belong to the octagon. They can be identified by the masks of Fig. 2(a) and (b), and their four 90° rotations.
- *Vertex pixels*: Only four of their neighbors belong to the octagon. They can be identified by the masks of Fig. 2(c) and (d), and their rotations.

Vertex pixels are then easy to identify, either testing that they have only 4-neighbors that belong to the octagon or by means of the adequate 3×3 masks. More powerful methods for local boundary analysis can be found in (Borgefors and Sanniti di Baja, 1996).

A way to control the growing speed of the vertex pixels would be to delete them (perform a "corner rounding" stage (Kulpa and Kruse, 1983)) in some iterations. As Fig. 3 shows, the iterated application of vertex elimination equals the truncation of the octagon vertex with a new edge of slope -2 (or 2, $-1/2$, 1/2, depending on the vertex).

The problem of suppressing vertices is that the algorithm needs two passes on each iteration: one

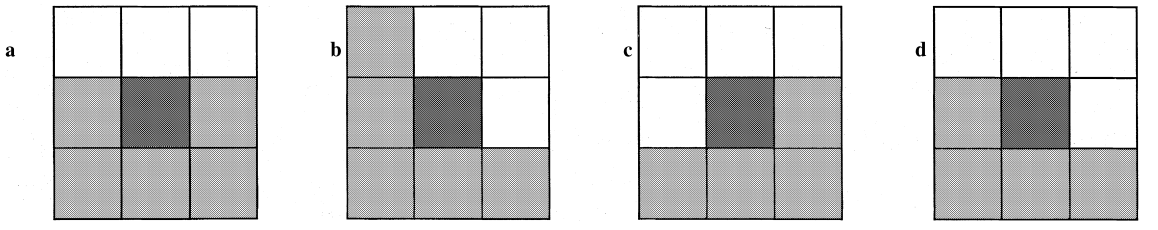

Fig. 2. (a,b) Masks for the edge pixels. (c,d) Masks for the vertex pixels.

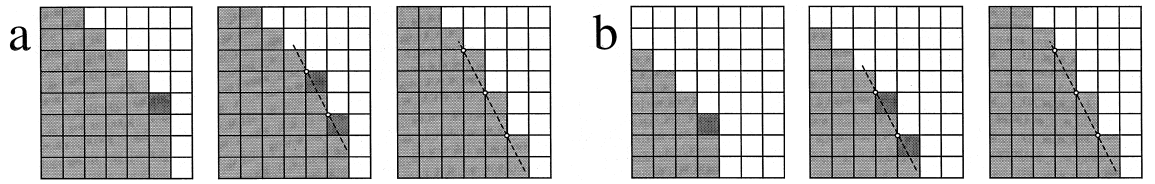

Fig. 3. (a) Results of iterated suppression of vertex pixels (two iterations). Deleted pixels are shown in dark grey. (b) Results of growth inhibition of vertex pixels (two iterations). The final result is equivalent to the one in Fig. 3(a). Dark grey pixels did not grow in the corresponding iteration. The first growth was a 4-*n*, and the second an 8-*n* one, in concordance with the regular octagonal growth model.

for the growth and one for suppression. Also, when the dilation algorithm is applied to general binary images, problems appear where the interference of several growing shapes occur (Kulpa and Kruse, 1983), as vertex points cannot be identified by the use of masks in those cases.

This problem can be avoided if an alternative approach to vertex suppression is used, in the form of growth inhibition of vertex pixels on some iterations (Thompson and Rosenfeld, 1997). As Fig. 3 shows, growth inhibition of vertex pixels is equivalent to their suppression. The proof will be omitted, but can be deduced from the analysis of the application of 4-*n* and 8-*n* growth patterns to the edge and vertex configurations of Fig. 2.

A control rule for vertex growth inhibition should now be devised, similar to the one we used for the regular octagonal growth. As before, the coordinates of the right-most upper vertex should be determined.

A vertex suppression (or inhibition) stage, moves the right-most upper vertex a pixel downwards. So, for a certain iteration, n, the suppression or inhibition technique should be applied as many times as the vertical distance between the right-most upper vertex of the regular octagon and the right-most upper vertex of the 16-sided polygon resulting of the truncation by a line of slope -2 tangent to the circumference of radius n. As the position of the right-most upper vertex in the truncated octagon is $(n, n(\sqrt{5}-2))$, the number of suppressions, n_s, is $n_s = n(\sqrt{2}-1) - n(\sqrt{5}-2) = n(1+\sqrt{2}-\sqrt{5})$ or $n_s/n = 1+\sqrt{2}-\sqrt{5}$. This value can be approximated by

$$\frac{n_s}{n} = 1+\sqrt{2}-\sqrt{5} \approx \frac{1}{5} - \frac{1}{45}, \tag{6}$$

which is valid for circumferences of at most 2718 pixels of radius. Thus, vertex growth inhibition has to be applied in those iterations that are multiple of 5 but not of 45. The application of this correction results in a 16-sided polygonal (or hexadecagonal) growing pattern (Fig. 1(e)). This control rule for growth inhibition is also valid for the corner rounding approach, and can be seen as a theoretical foundation for the empirical hexadecagonal procedure presented in (Kulpa and Kruse, 1983).

The final algorithm, then, is as follows:

```
for k:=1 to R
  for every pixel p in the boundary
    if NOT((p is a vertex) AND (k module
    5=0) AND (k module 45!=0))
      if ((k module 2=0) AND (k module
      12!=0) AND (k module 410!=0))
        grow p as 8-n
      otherwise
        grow p as 4-n
```

This algorithm can be easily optimized to make the modulus computations once per iteration instead of once per boundary pixel. Furthermore, the vertex configuration verification can be done by just counting the number of neighbors of the pixel that belong to the polygon, instead of checking against the eight possible vertex masks. The reason is that the configurations in which a vertex growth should be inhibited but do not satisfy the four neighbor condition evolve to the same resulting configuration by 8-*n* or 4-*n* growths regardless of the inhibition of the vertex growth. As the mask matching process is not needed, the filtering

problems regarding corner rounding (as in (Kulpa and Kruse, 1983)) are avoided.

4. Results and applications

The quality of each approximation to circular growth can be evaluated by the comparison of an ideal circle and the polygonal shapes resulting from the iterated application of the different growth models to an isolated pixel. Table 1 shows the area error, distance errors and area to square of number of iterations ratio for each growth model.

The area error is computed as follows:

$$\text{Area\_error} = 100\left|\frac{A_m - A_c}{A_c}\right|, \tag{7}$$

where A_m is the area of the polygon associated to the growth model, and A_c is the area of the circle inscribed in that polygon. The distance error is obtained in a similar fashion

$$\text{Distance\_error} = 100\left|\frac{r'_c - r_c}{r'_c}\right|,$$

where r_c and r'_c are the radius of the inscribed and circumscribed circles. The growth model assigns a distance value equal to the radius of the inscribed circle to all the points of the polygon boundary, so the biggest error in the distance measure will correspond to the vertex that deviates the most from the inscribed circle. By definition, the euclidean distance from the center of growth to that vertex is the radius of the circle circumscribed to the polygon.

For the comparisons, it must be pointed out that the 4-*n* model grows as an inscribed polygonal model to the ideal circular growth, as opposed to the other growth models, which generate circumscribed polygonal growths to the ideal circular model. Also, the Chamfer 3/4 and 5/7/11 distances used for dilation simulation have been divided by $\sqrt{10}$ and $\sqrt{26}$, respectively, instead of 3 and 5 which are the usual normalization factors, to generate a circumscribed polygonal growth.

The results of Table 1 clearly show the superiority of the proposed hexadecagonal model over the other 3×3 growth patterns, and even the simulation of dilation based on the 5×5 Chamfer 5/7/11 model.

In Fig. 4, which shows a comparison of the different growth models for binary image dilation, it can be easily appreciated how the proposed hexadecagonal model surpasses the other 3×3 growth methods in their approximation to the ideal circular growth.

Apart from the most evident applications, such as dilation and erosion algorithms, or skeleton extraction, there exist other areas of application where traditional growth algorithms cannot be used because of their limitations. That is the case of Voronoi diagram generation from isolated image pixels (Tsai, 1993) or segments (Chou, 1995). The Voronoi diagram for a set of elements divides the space in a set of regions where every region is the set of points that are closer to a certain element than to the others. Voronoi diagram generation is a very difficult analytical problem, but is easily solved by labeling the elements and expanding them with a growth algorithm, such as our hexadecagonal model. Furthermore, the connectivity graph of the elements is easily recovered by testing the labels of the adjoint regions, and setting a new edge for every frontier in the diagram.

Table 1
Area to square of number of iterations ratio, area and distance errors for each growth model

Growth model	Area to n^2 ratio	% area error (%)	% distance error (%)
4-neighbors (inscribed)	2	36.34	70.71
8-neighbors	4	27.32	29.29
Octagonal	3.5	11.41	10.56
Chamfer 3/4 distance	3.33333...	6.10	10.56
Regular octagonal	3.31371...	5.48	7.61
Chamfer 5/7/11 distance	3.24156...	3.18	3.52
Proposed hexadecagonal algorithm	3.18677...	1.44	2.67
Ideal circular growth	$\pi = 3.14159...$	0	0

Fig. 4. Comparison of the analysed 3×3 growth methods for binary image dilation, after 24 iterations. (a) Binary image with several isolated segments, (b) 4-neighbor dilation, (c) 8-neighbor dilation, (d) dilation by (4,8) octagonal growth, (e) simulation of dilation by Chamfer 3/4 distance, (f) dilation by regular octagonal growth and (g) proposed hexadecagonal growth.

Fig. 5 shows the result of the application of the algorithm for the extraction of the Voronoi diagram of a set of arbitrary shapes.

Other operations, such as skeletonization (Sanniti di Baja, 1994) or determination of pixel clusters (Haralick and Shapiro, 1992), are based on the calculation of distance maps. The proposed algorithm can also generate global distance maps, just by labelling the pixels filled on every iteration with the iteration number. The results of Table 1 show that the hexadecagon resulting from the proposed model is a better approximation to the ideal circle than the hexadecagon associated with the Chamfer 5/7/11 model. Hence, the distance mapping based on the proposed hexadecagonal growth should be preferred to the Chamfer 5/7/11 hexadecagonal distance.

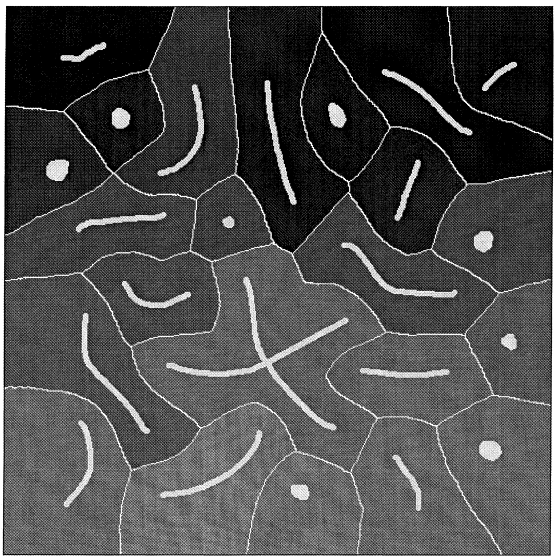

Fig. 5. Voronoi diagram of a set of arbitrary shapes.

The same techniques used for rescaling the Chamfer distances on (Vossepoel, 1988; Borgefors, 1991) could be applied to the proposed model to further refine the distance calculations.

Fig. 6 shows the distance maps for the segments of Fig. 4(a) generated by Chamfer 3/4 octogonal mapping, Chamfer 5/7/11 hexadecagonal mapping, hexadecagonal mapping using the proposed hexadecagonal growth model, and euclidean distance mapping.

5. Conclusions

A new procedure for hexadecagonal approximation to circular growth has been developed and presented in this paper. It is based on the analysis of the same 3×3 regions used by the most common local deterministic growth algorithms, but with far better behaviour than those methods. This new implementation does not need a rounding stage, or the additional memory resources required in its traditional forms. The quality of the proposed hexadecagonal approximation and the ease of application of this new algorithm widens the area of application of this type of local dilation technique.

The growth method is based on the regulated application of 8-*n* and 4-*n* dilations to obtain a regular octagonal growth model, which is then refined to obtain hexadecagonal growth patterns by inhibiting the growth of the vertex pixels of the octagon. The growth inhibition technique described in this article can be applied to regions bigger than 3×3 to achieve better results. However, for the usual image sizes (512×512, 1024×1024...) the approximation error of the algorithm achieved with these 3×3 regions can be

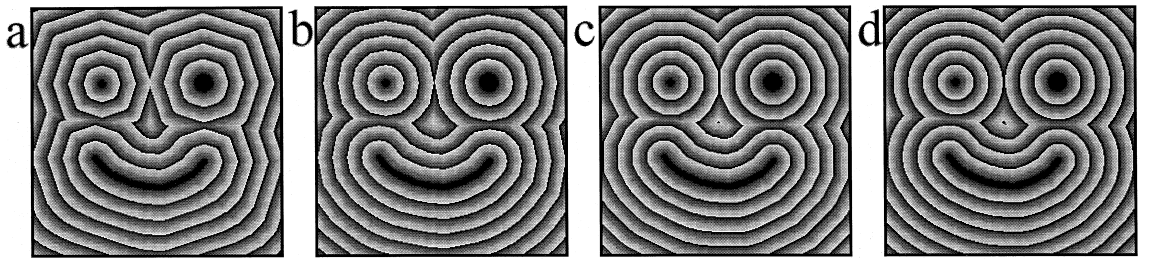

Fig. 6. Distance mappings for the segments of Fig. 4(a). (a) Chamfer 3/4 distance mapping, (b) Chamfer 5/7/11 distance mapping, (c) hexadecagonal distance mapping using the proposed algorithm, and (d) euclidean distance mapping. To better appreciate the maps pattern, the distances are shown module 32. The images are 256×256 pixels and the maximum distance evaluation errors are 12.39 pixels for Chamfer 3/4 distance, 3.44 pixels for Chamfer 5/7/11 and 2.16 pixels for the proposed hexadecagonal distance model.

considered small. The negligible errors and the ease of implementation make this new hexadecagonal algorithm an excellent substitute for the 4-*n*, 8-*n*, octagonal and chamfer distance based dilation methods.

The proposed hexadecagonal growth model also has other areas of application, such as distance mapping or Voronoi diagram generation. The hexadecagonal distance mapping resulting from the application of the proposed growth model is a better approximation to the ideal circular growth than that of the "standard" hexadecagonal distance, the Chamfer 5/7/11 model.

Acknowledgements

The authors wish to thank the referees for their comments on the earlier version of the paper.

References

Borgefors, G., 1986. Distance transformation in digital images. Computer Vision, Graphics, Image Processing 34.

Borgefors, G., 1991. Another comment on "A note on "distance transformation in digital images"". CVGIP: Image Understanding 54 (2).

Borgefors, G., Sanniti di Baja, G., 1996. Analyzing nonconvex 2D and 3D patterns. Computer Vision and Image Understanding 63 (1).

Chou, J.J., 1995. Voronoi diagrams for planar shapes. IEEE Computer Graphics and Applications, March.

Danielsson, P.E., 1980. Euclidean distance mapping. Computer Graphics and Image Processing 14.

Das, P.P., 1992. Best simple octagonal distances in digital geometry. Journal of Approximation Theory 68.

Haralick R.M., Shapiro L., 1992. Computer and Robot Vision. Addison-Wesley, Reading, MA.

Kulpa, Z., Kruse, B., 1983. Algorithms for circular propagation in discrete images. Computer Vision, Graphics, and Image Processing 24.

Rosenfeld, A., Pfaltz, J.L., 1968. Distance functions on digital pictures. Pattern Recognition 1.

Sanniti di Baja G., 1994. Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform. Journal of Visual Communication and Image Representation 5 (1).

Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press, London.

Thompson, S., Rosenfeld, A., 1995. Isotropic growth on a grid. Pattern Recognition 28 (2).

Thompson, S., Rosenfeld, A., 1997. Growth processes based on 8-neigbor time delays. Pattern Recognition 30 (2).

Tsai, V.J.D., 1993. Fast topological construction of Delaunay triangulations and Voronoi diagrams. Computer and Geosciences 19 (10).

Vossepoel, A.M., 1988. A Note on "Distance transformations in digital images". Computer Vision, Graphics, and Image Processing 43.