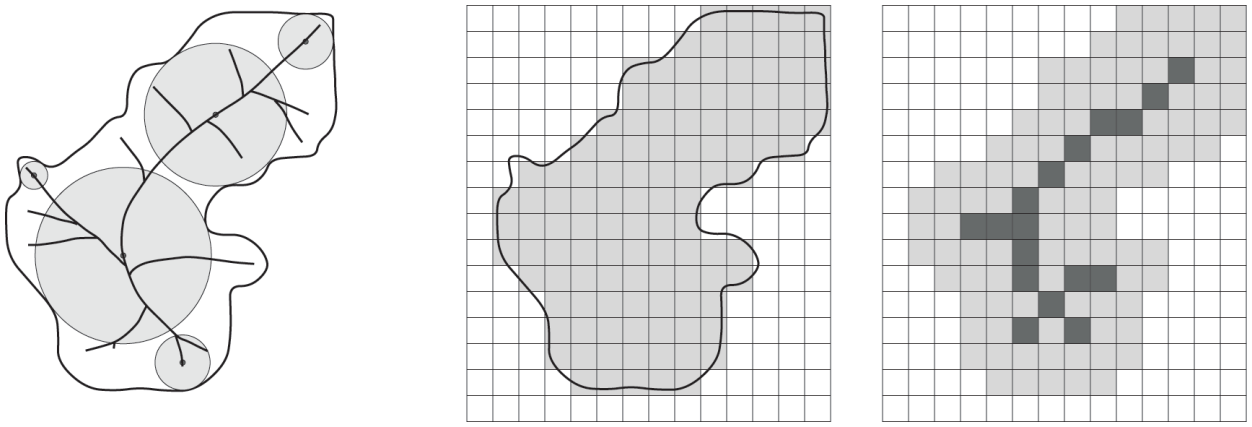
Medial Axes

The *medial axis* of a planar region S consists of all centers of maximum disks in S. (A maximum disk is one that is not contained in a larger disk which is also contained in S.)

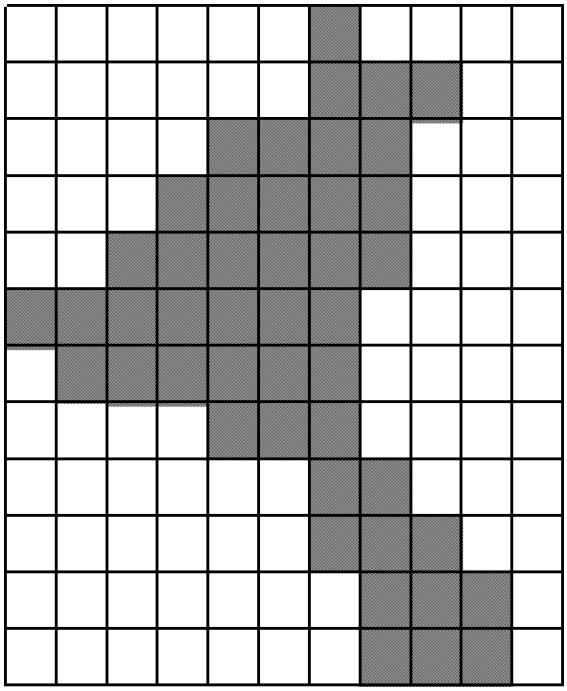


Left: a set in the Euclidean plane and its medial axis. Middle: the Gauss digitization of this set at some grid resolution. Right: illustration of an expected outcome when calculating the medial axis. The chosen grid resolution restricts the complexity of the medial axis. In the real plane, any cavity "splits" the medial axis into two branches.

The use of the medial axis for picture analysis has been proposed by H. Blum in 1967, and in 1968 by L. Calabi and W. E. Hartnett.

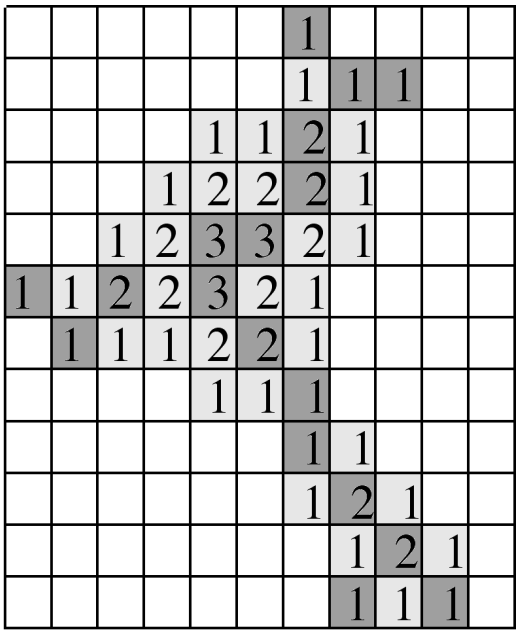
Example in the Grid

result of a d_4 distance transform:



						1				
						1	1	1		
				1	1	2	1			
			1	2	2	2	1			
		1	2	3	3	2	1			
1	1	2	2	3	2	1				
	1	1	1	2	2	1				
				1	1	1				
						1	1			
						1	2	1		
							1	2	1	
							1	1	1	

Local maxima of distance values (within the 4-neighborhood of a pixel) are the centers of maximum disks.



						1				
						1	1	1		
				1	1	2	1			
			1	2	2	2	1			
		1	2	3	3	2	1			
1	1	2	2	3	2	1				
	1	1	1	2	2	1				
				1	1	1				
						1	1			
						1	2	1		
							1	2	1	
							1	1	1	

For grid metric d_α, pixel p, and $k \geq 0$, we have balls

$$B_\alpha^{(k)}(p) = \{q : \ d_\alpha(p,q) \leq k\}$$

Thus: $B_\alpha^{(0)}(p) = \{p\} \subset B_\alpha^{(1)}(p) \subset B_\alpha^{(2)}(p) \subset \cdots$

$\alpha = 4$: $B_\alpha^{(k)}(p)$ is a diagonally oriented square centered at p

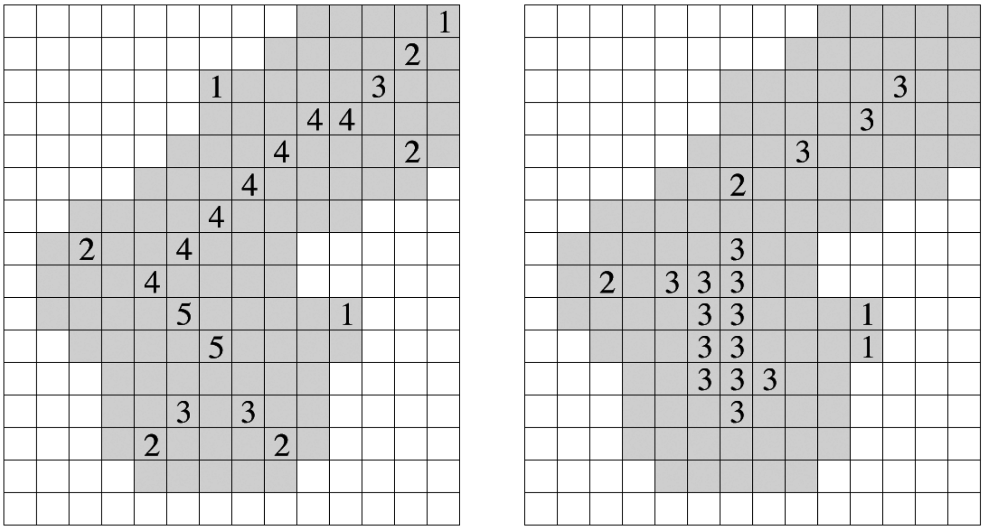
$\alpha = 8$: $B_\alpha^{(k)}(p)$ is an upright square centered at p

If $p \in \langle P \rangle = P^{-1}(1)$ and $k < d_\alpha(p, \langle \overline{P} \rangle)$, then $B_\alpha^{(k)}(p) \subseteq \langle P \rangle$, and all of the $B_\alpha^{(k)}(p)$s contain p, so $\langle P \rangle$ is the union of all balls $B_\alpha^{(k)}(p)$ with $p \in \langle P \rangle$ and $k < d_\alpha(p, \langle \overline{P} \rangle)$.

In the d_α distance transform of P, each pixel p of $\langle P \rangle$ has value $d_\alpha(p, \langle \overline{P} \rangle)$.

Definition 1 *Pixel p belongs to the medial axis $M_\alpha(\langle P \rangle)$ of $\langle P \rangle$ iff $d_\alpha(p, \langle \overline{P} \rangle)$ is a local maximum of the d_α distance transform of P within the α-neighborhood of p.*

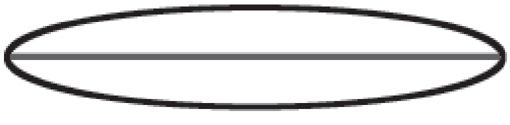
The picture in which the value of p is $d_\alpha(p, \langle \overline{P} \rangle)$ if $p \in M_\alpha(\langle P \rangle)$ and 0 otherwise is called the d_α *medial axis transform* (MAT) of $\langle P \rangle$.



Medial axis transforms for distance transforms (d_4 and d_8 as shown in Lecture 08 for this set). Left: d_4. Right: d_8.

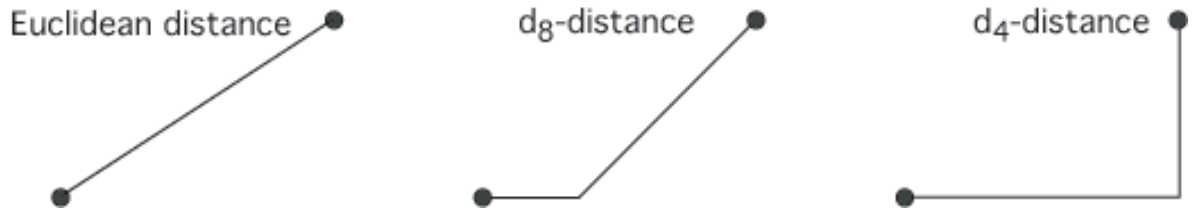
The pixels of $M_\alpha(\langle P \rangle)$ are centrally located in $\langle P \rangle$, so they constitute a kind of "skeleton" of $\langle P \rangle$. This skeleton may not be connected even if $\langle P \rangle$ is simply connected, and it may be two pixels thick if $\langle P \rangle$ has even width.

> The MAT was originally described as result of a "prairie fire" ignited along the border of $\langle P \rangle$ and defined $M(\langle P \rangle)$ as the locus of points at which the grass fire meets itself. However, a pixel on the medial axis is not necessarily characterized by two different shortest α-paths from p to $\langle \overline{P} \rangle$.



MAT following a Euclidean distance transform are in general closest to our "expectations".

Euclidean Distance is the 'Ideal Case'



Danielsson's Algorithm

(designed for calculating a Euclidean distance transform in which the distances differ from Euclidean distance by at most a fraction of the grid constant; published in 1980)

To each pixel $p = (x, y)$ of P the algorithm assigns a pair $f(x, y) = (u, v)$ of integers that is initially

(0,0) if $p \in \langle \overline{P} \rangle = P^{-1}(0)$ and

(D, D) if $p \in \langle P \rangle$, where D is greater than the diameter of P (the greatest distance between any two pixels of P).

We then scan P and update the $f(x, y) = (u, v)$ values as described in the algorithm on page 7. The intention is that finally $\sqrt{u^2 + v^2}$ is the minimum Euclidean distance between $p = (x, y)$ and pixels in $\langle \overline{P} \rangle$.

The min-Operation in the Algorithm

In the algorithm we use an operation $\min\{(u_1, v_1), (u_2, v_2)\} =$

(u_1, v_1), if $u_1^2 + v_1^2 < u_2^2 + v_2^2$

(u_2, v_2), if $u_2^2 + v_2^2 < u_1^2 + v_1^2$

(u_1, v_1), if $u_1^2 + v_1^2 = u_2^2 + v_2^2$ and $u_1 < u_2$

(u_2, v_2), if $u_1^2 + v_1^2 = u_2^2 + v_2^2$ and $u_2 \leq u_1$

Three Scans in Both Steps

The algorithm (on the next page) lists two steps. In each step, we have three successive scans of all pixels. The orientation of coordinate axes is as shown below:

Let P be defined on $\mathbb{G}_{m,n} = \{(x, y) : 1 \leq x \leq m \ \wedge \ 1 \leq y \leq n\}$.

In the first scan of Step 1 (top to bottom, left to right), we compare the pair $f(x, y)$ with the pair $f(x, y-1)$, for all $p = (x, y)$ in the picture. We assume that all grid points outside of $\mathbb{G}_{m,n}$ have value 0, and have been initialized by $f(x, y) = (0, 0)$. Thus we are able to start the comparisons in row 1 by comparing with these assumed values in row 0. – Analogously we proceed in the other scans.

Step 1 (*Scan 1*) For each pixel of P (from top to bottom, from left to right), replace each $f(x,y)$ with

$$\min\{f(x,y), f(x,y-1)+(0,1)\};$$

(*Scan 2*) then replace in a second scan (from top to bottom, from left to right) each $f(x,y)$ with

$$\min\{f(x,y), f(x-1,y)+(1,0)\};$$

(*Scan 3*) then replace in a third scan (from top to bottom, from right to left) each $f(x,y)$ with

$$\min\{f(x,y), f(x+1,y)+(1,0)\}.$$

Step 2 (*Scan 1*) For each pixel of P (from bottom to top, and, e.g., from left to right) replace each $f(x,y)$ with

$$\min\{f(x,y), f(x,y+1)+(0,1)\};$$

(*Scan 2*) then replace in a second scan (from left to right, and, e.g., from top to bottom) each $f(x,y)$ with

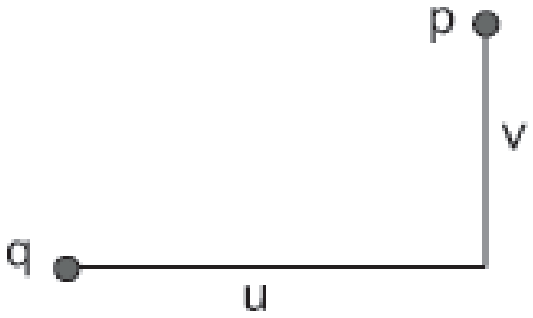
$$\min\{f(x,y), f(x-1,y)+(1,0)\};$$

(*Scan 3*) then replace in a third scan (from right to left, and, e.g., from top to bottom) each $f(x,y)$ with

$$\min\{f(x,y), f(x+1,y)+(1,0)\}.$$

Note that Scans 2 and 3 in Steps 1 and 2 are identical. In all scans we use the assumption that pixels outside of $\mathbb{G}_{m,n}$ have value 0, and stay with $f(x, y) = (0, 0)$ all the time.

When the scans are complete, we have $f(x, y) = (u, v)$ at pixel $p = (x, y)$, and the value of u should be the difference between the x coordinates of p and the nearest pixel q of $\langle \overline{P} \rangle$, and the value of v should be the difference between their y coordinates.



The Euclidean distance between p and q would then be $\sqrt{u^2 + v^2}$.

Unfortunately, this is not always the case; the $f(x, y)$ values are not always exactly equal to the nearest-pixel coordinate differences.

									0,1	0,1	0,1	0,1	0,1
								0,1	1,1	0,2	0,2	0,2	1,0
						0,1	0,1	1,1	1,2	2,2	0,3	0,2	1,0
						0,1	0,2	1,2	2,2	1,3	2,2	0,2	1,0
					0,1	1,1	1,2	1,3	2,2	1,2	0,2	1,1	0,1
				0,1	1,1	1,2	2,2	1,2	0,2	1,1	0,1	0,1	
		0,1	0,1	1,1	1,2	2,2	1,2	1,1	0,1	0,1			
	0,1	1,1	0,2	1,2	2,2	3,0	2,0	1,0					
	1,0	0,2	2,2	1,3	2,3	3,0	2,0	1,0					
	0,1	1,1	1,2	2,2	3,2	3,1	2,1	1,1	0,1	0,1			
		0,1	1,1	2,1	3,1	3,2	2,2	1,2	1,1	0,1			
			1,0	2,0	3,0	0,4	3,0	2,0	1,0				
			1,0	2,0	2,2	0,3	2,2	2,0	1,0				
			0,1	1,1	0,2	0,2	0,2	1,1	1,0				
				0,1	0,1	0,1	0,1	0,1					

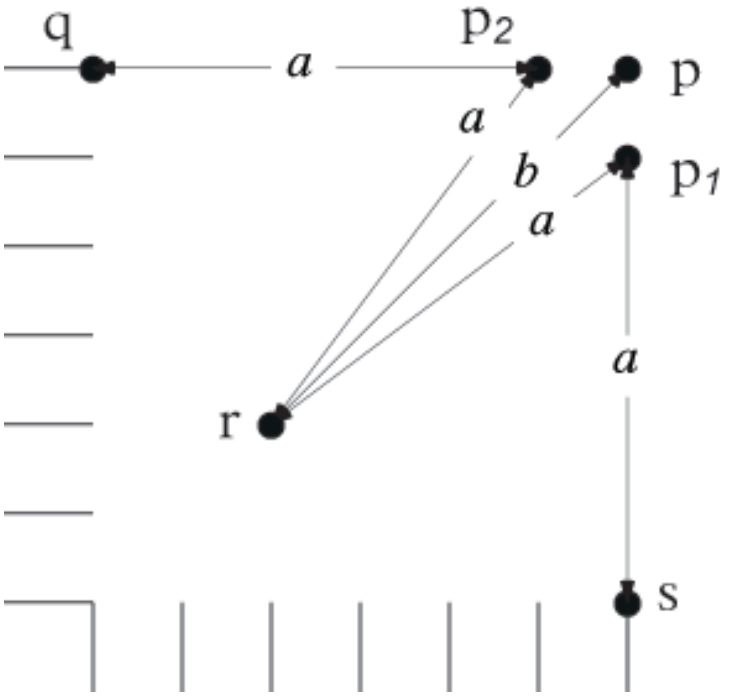
									1	1	1	1	1
								1	1.41	2	2	2	1
						1	1	1.41	2.24	2.83	3	2	1
						1	2	2.24	2.83	3.16	2.83	2	1
					1	1.41	2.24	3.16	2.83	2.24	2	1.41	1
				1	1.41	2.24	2.83	2.24	2	1.41	1	1	
		1	1	1.41	2.24	2.83	2.24	1.41	1	1			
	1	1.41	2	2.24	2.83	3	2	1					
	1	2	2.83	3.16	3.61	3	2	1					
	1	1.41	2.24	2.83	3.61	3.16	2.24	1.41	1	1			
		1	1.41	2.24	3.16	3.61	2.83	2.24	1.41	1			
			1	2	3	4	3	2	1				
			1	2	2.83	3	2.83	2	1				
			1	1.41	2	2	2	1.41	1				
				1	1	1	1	1					

In this example, the calculated $(f(x, y)$ values are all correct.

Left: the pair of (final) values at a pixel of $\langle P \rangle$ are its x and y coordinate differences from the nearest pixel of $\langle \overline{P} \rangle$. Right: corresponding values of the Euclidean distance, rounded to two decimal places.

Sometimes we have triples a, c, d of integers such that $a^2 = c^2 + d^2$ (see figure for $a = 5$; c and d are not shown and are $c = 3$ and $d = 4$).

Assume that q, r, s are points in $\langle \overline{P} \rangle$, p, p_1, p_2 are points in $\langle P \rangle$, and that there is no other point in $\langle \overline{P} \rangle$ which is closer to p than Euclidean distance b. Note that $b < a + 1$ (due to triangularity of the Euclidean metric).



In this case, the algorithm generates value $f(p) = (0, a + 1)$ and thus the distance value $a + 1$. Actually, it should be the distance value b. (Note: there are more cases where this algorithm does not calculate the correct Euclidean distance.)

Independent Row and Column Scans

Initialization: $f(x,y) = 0$ for $(x,y) \in \langle \overline{P} \rangle$, and $f(x,y) = D$ (D as on page 5) at pixels $(x,y) \in \langle P \rangle$ (note: $f(x,y)$ is now just a scalar); f is a picture of the same size $m \times n$ as P, and we also use a 1D array g of size n (note: n is the number of rows).

Step 1 (*Rowscan 1*) Replace (each row from left to right) each $f(x,y)$ with $\min\{f(x,y), f(x-1,y)+1\}$;
(*Rowscan 2*) then replace (each row from right to left) each $f(x,y)$ with $\min\{f(x,y), f(x+1,y)+1\}$.

Step 2 (*Columnscan*) For each column x, do
(i) for $1 \leq y \leq n$ calculate

$$g(y) = \min\{\sqrt{f(x,i)^2 + (y-i)^2} : 1 \leq i \leq n\};$$

(ii) replace column x in array f by g (i.e., $f(x,y)$ by $g(y)$).

Efficiency improvements: you may decide not to use the root when calculating $g(y)$; you do not have to go further than $f(x,y)$ up or down when calculating the minimum; if $f(x,i) = 0$ then you do not have to consider values above (below) i.

This algorithm is a Euclidean distance transform (see, e.g., T. Saito and J.I. Toriwaki, 1994). (The article has been cited, but the algorithm is not in the textbook.)

Coursework

Related material in textbook: Sections 3.4.4 and 3.4.3.

A.9. [5 marks] Discuss the Euclidean distance transform (i.e., independent row and column scans) by

(i) an argumentation why it is always correct,

(ii) an example of a low-resolution binary picture where you illustrate the calculated values in array f after the two rowscans, and after the final minimization in each column,

(iii) discuss the time complexity of your implementation of this algorithm for binary pictures of size $2^n \times 2^n$, for $n = 6, \ldots, 10$ (optionally, you may also discuss the three suggested ways for optimization),

(iv) calculate for a few selected pictures the MAT based on the Euclidean distance transform (see Appendix of this lecture).

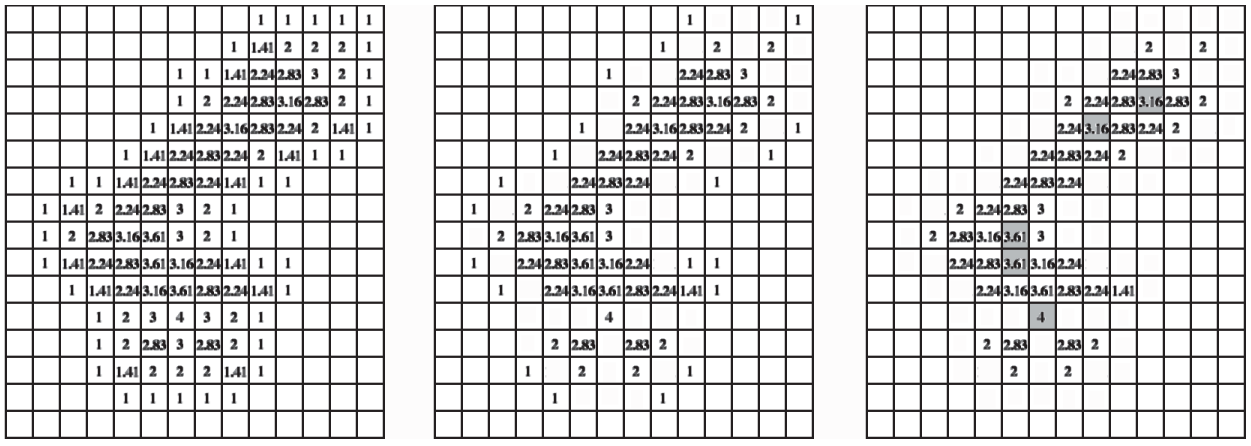
Hint: for **(iv)** you detect local maxima in $N_8(p)$, where the distance difference along isothetic edges can be 1, but along diagonal edges you allow a difference of $\sqrt{2}$.

Appendix: Euclidean MAT

The MAT for the Euclidean distance transform requires a modification compared to Definition 1 on page 3 (Lecture 09).

Definition 2 *Pixel p belongs to the medial axis $M_e(\langle P \rangle)$ of $\langle P \rangle$ iff $d_e(p, \langle \overline{P} \rangle)$ is greater than $d_e(q, \langle \overline{P} \rangle) - 1$, for all $q \in A_4(p)$, and greater than $d_e(q, \langle \overline{P} \rangle) - \sqrt{2}$, for all $q \in A_8(p) \setminus A_4(p)$.*

Example: $d_e(p, \langle \overline{P} \rangle) = 5$ and maximum $d_e(q, \langle \overline{P} \rangle) = 5.5$, for all four points $q \in A_4(p)$: A step along an isothetic edge is of length 1, i.e., these values would still allow that p is a center of a maximum disk. — Maximum $d_e(q, \langle \overline{P} \rangle) = 6.2$, for all four points $q \in A_8(p) \setminus A_4(p)$: A step along a diagonal edge is of length $\sqrt{2}$, i.e., these values would still allow that p is a center of a maximum disk.



Left: Euclidean distance transform. Middle: Euclidean MAT. Right: Removal of all 1*s* (considered to be "noise" along borders), and local maxima in 8-neighborhoods are labeled by gray squares.