Note on fifteen 2D parallel thinning algorithms

M. Couprie

Institut Gaspard-Monge, Unité Mixte de Recherche CNRS-UMLV-ESIEE UMR 8049 — Laboratoire A2SI, Groupe ESIEE — Cité Descartes, BP99, 93162 Noisy-le-Grand Cedex France

Abstract

We present a study of fifteen parallel thinning algorithms, based on the framework of critical kernels. We prove that ten among these fifteen algorithms indeed guarantee topology preservation, and give counter-examples for the five other ones. We also investigate, for some of these algorithms, the relation between the medial axis and the obtained homotopic skeleton.

Key words: Parallel thinning, verification of algorithms, digital topology, homotopy, critical kernels

Introduction

During the last 40 years (the first parallel thinning algorithm was proposed by D. Rutovitz in 1966 [31]), many 2D parallel thinning methods have been proposed, see in particular [1,28,8,16,14,13,18,10,2]. Proving that such an algorithm always preserve topology is not an easy task, even in 2D. The proofs found in the literature are often combinatorial and cannot be extended to 3D, a fortiori to higher dimensions. For the 2D case, C. Ronse introduced the *minimal non simple sets* [29] to study the conditions under which simple points can be removed in parallel while preserving topology. This leads to verification methods for the topological soundness of thinning algorithms. Such methods have been proposed for 2-D algorithms by C. Ronse [29] and R. Hall [15], they have been developed for the 3-D case by T.Y. Kong [19,20] and C.M. Ma [25], as well as for the 4-D case by C-J. Gau and T.Y. Kong [11,21]. For the 3D

Email address: m.couprie@esiee.fr (M. Couprie).

case, G. Bertrand introduced the notion of *P-simple points* [3] as a verification method but also as a methodology to design parallel thinning algorithms [4,7,22,23].

In [5], G. Bertrand introduces a general framework for the study of parallel thinning in any dimension in the context of abstract complexes. As shown in [6], this framework allows to retreive both the notion of minimal non-simple set and the notion of P-simple point. A new definition of a simple point is proposed in [5], this definition is based on the collapse operation which is a classical tool in algebraic topology and which guarantees topology preservation. Then, the notions of an *essential face* and of a *core* of a face allow to define the *critical kernel* $\mathcal{K}$ of an object X. The most fundamental result proved in [5] is that, if a subset Y of X contains $\mathcal{K}$, then X collapses onto Y, *i.e.,* Y is a retraction of X.

In [6], the particular case of 2D structures in spaces of two and three dimensions is considered. Several new parallel thinning algorithms are proposed and compared with the existing ones, when possible. For example, one of these new algorithms is proved to include the medial axis and to be minimal for this property; this algorithm has no equivalent in the literature.

Thanks to the general framework of critical kernels, and to the results proved in [6] for the 2D case, we analyse in this report the topological soundness of fifteen parallel thinning algorithms. To limit the study, we do not consider here any algorithm based on directional sub-steps or sub-grids.

This analyzis is performed "automatically" with the help of a computer program. Similar computerized tests have already been proposed by R. Hall [15], C-M. Ma [24] for 2D, based on the notion of minimal non-simple sets [29], and by C-M. Ma [24] for 3D.

Here, we prove the topological soundness of ten among the fifteen analyzed algorithms. For the five other ones, we show the counter-examples found by our program. We also investigate, for some of these algorithms, the relation between the medial axis and the obtained homotopic skeleton.

Some basic notions are recalled in Sec. 1, the verification method is described in Sec. 2, and the following sections are devoted to the different algorithms under study. A general discussion concludes this report.

1 Basic notions

In most papers on digital topology, a binary image is considered as a finite subset of $\mathbb{Z}^2$. However, an alternative interpretation consists in considering an image as a finite set of *pixels*, that is, unit squares which have all their vertices in $\mathbb{Z}^2$. The latter interpretation is taken in [6], in order to make a link between the framework of critical kernels and digital topology. However, both interpretations are clearly equivalent and can be easily translated into each other.

We denote by $\mathbb{G}^2$ the set of all the pixels, also called the *square grid*, and we consider only finite subsets of $\mathbb{G}^2$.

14	13	12	11	10
15	3	2	1	9
16	4	P	0	8
17	5	6	7	23
18	19	20	21	22

Fig. 1. Numbering scheme for the pixels in the vicinity of P.

Let P be a pixel in $\mathbb{G}^2$. The pixels in the vicinity of P are identified by the numbers $0, \ldots, 23$ (see Fig. 1), more precisely, those pixels are denoted by $\Gamma_0(P), \ldots, \Gamma_{23}(P)$. The set $\Gamma^*(P) = \{\Gamma_i(P) \mid 0 \leq i \leq 7\}$ is called the *neighborhood of* P; each of these pixels is called a *neighbor of* P. The four pixels of $\Gamma^*_S(P) = \{\Gamma_0(P), \Gamma_2(P), \Gamma_4(P), \Gamma_6(P)\}$ are called the *strong neighbors of* P. If P, Q are pixels and if $Q \in \Gamma^*(P)$ (resp. $\Gamma^*_S(P)$), then we say that P, Q *are adjacent* (resp. *strongly adjacent*). We set $\Gamma(P) = \Gamma^*(P) \cup \{P\}$ and $\Gamma_S(P) = \Gamma^*_S(P) \cup \{P\}$. Notice that Γ^* and Γ^*_S correspond to the usual notions of 8- and 4-adjacency, respectively.

Let X be a subset of $\mathbb{G}^2$ (the "object"). We denote by $\overline{X}$ the complementary set of X (the "background"). We say that a pixel $P \in X$ is a *border pixel* if it is strongly adjacent to a pixel in $\overline{X}$.

A sequence $\pi = \langle x_0, \ldots, x_l \rangle$ of pixels in X is a *path in* X *(from* x_0 *to* x_l*)* if x_i and x_{i+1} are adjacent for each $i = 0, \ldots, l-1$. We say that X is *connected* if, for any pair of pixels x, y in X, there is a path in X from x to y. We say that $Y \subseteq X$ is a *connected component of* X if Y is connected, and if Y is maximal for these two properties (*i.e.*, if we have $Z = Y$ whenever $Y \subseteq Z \subseteq X$ and Z connected). The notions of *strong path*, *stronly connected*, *strong connected component* are defined analogously, using the strong adjacency.

Intuitively a pixel P of X is *simple* if its removal from X "does not change the topology of X". In [5], G. Bertrand introduces a definition of a simple n-dimensional element based on the operation of collapse [12]. In the square grid, we retreive thanks to this definition (see [6]) a well-known characterization of simple pixels given by A. Rosenfeld [30].

Property 1. *Let $X \subset \mathbb{G}^2$, and let $P \in X$. The pixel P is simple for X if and only if:*
i) P is a border pixel; and
ii) $\Gamma^*(P) \cap X$ is non-empty and connected.

The following notations will be used in sections $3 - 15$ (algorithms).

Let $X \subset \mathbb{G}^2$ and let $P \in X$. In the sequel, for any $i = 0 \ldots 23$, we denote by P_i the boolean value which is 1 if $\Gamma_i(P) \in X$ and 0 otherwise. We denote by $\overline{P_i}$ the negation of the boolean P_i, in other words, $\overline{P_i} = 1 - P_i$.

We denote by $D(P)$ the number of strong connected components of pixels of X in the neighborhood of P.

We denote by $B(P)$ the number of neighbors of P which belong to X.

We denote by $C(P)$ the number of patterns "01" in the ordered sequence $P_0P_1 \ldots P_7P_0$, in other words, $C(P) = \frac{1}{2}\sum_{i=1}^{8} |P_{i \bmod 8} - P_{i-1}|$. This number is sometimes called the *crossing number* of P in the literature.

2 Verification methodology

We present here a notion introduced in [6], which allows for testing the topological soundness of parallel deletion of simple pixels. The original definition is not given here for the sake of simplicity, instead we give a characterization which is proven in [6] to be equivalent to the definition.

Let $X \subset \mathbb{G}^2$. We say that a pixel is *crucial (for X)* if it is matched by one of the masks depicted in Fig. 2. We say that a set C of crucial pixels forms a *crucial clique* if any two distinct pixels in C are adjacent to each other, and if C is maximal for this property.

In [6], it has been proved that an algorithm which does never remove in a single step any non-simple pixel nor any crucial clique, always produces a retraction of the original object, in other words, it always preserves topology.

Definition 2. *Let $X \subset \mathbb{G}^2$ and let $Y \subseteq X$.*
We say that Y is a crucial retraction of X if:
i) Y contains each pixel of X which is not simple; and
ii) Y contains at least one pixel of each crucial clique of X.

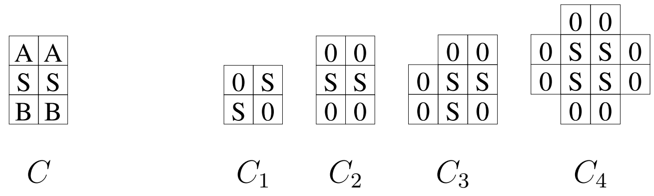


Fig. 2. Patterns and masks for crucial pixels and cliques. The 11 masks corresponding to these 5 patterns are obtained from them by any series of $\pi/2$ rotations. The label 0 indicates pixels that must belong to the set $\overline{X}$. The label S indicates pixels that must belong to the set S which is the set composed of all simple pixels of X. For mask C, at least one of the pixels marked A and at least one of the pixels marked B must be in X. If one of these masks matches the sets $\langle X, S\rangle$, then all the pixels which correspond to a label S in the mask are recorded as "matched".

Property 3 ([6]). *Let $X \subset \mathbb{G}^2$ and let $Y \subseteq X$.*
If Y is a crucial retraction of X, then Y is a retraction of X.

Notice that the configurations of Fig. 2 also characterize the minimal non-simple sets as defined by C. Ronse [29].

Let $X \subset \mathbb{G}^2$, let $A(X)$ denote the result of one step of a parallel thinning algorithm A on the input X. We suppose furthermore that the fact that a pixel P belongs to $A(X)$ or not depends only on the set $X \cap \Gamma^2(P)$, where $\Gamma^2(P) = \Gamma(\Gamma(P))$. We say that an algorithm A which satisfies this condition is *25-local*. We say that an algorithm A is *symmetrical* if $A(R(X)) = R(A(X))$, for any shape X and any rotation R by a multiple of $\pi/2$).

Let P be any pixel in $\mathbb{G}^2$, let $\mathcal{X}_1$ be the set of all the subsets of $\Gamma^2(P)$ which contain P. There are $2^{24} = 16,777,216$ such subsets. Let $\mathcal{X}_2$ be the set of all the sets X of $\mathcal{X}_1$ such that P is not simple for X. Clearly, if an algorithm A is 25-local and if P belongs to $A(X)$ for any X in $\mathcal{X}_2$, then whatever its input, algorithm A does not remove any non-simple pixel in a single step.

The family $\mathcal{X}_2$ can be generated by a computer program by producing and filtering $\mathcal{X}_1$, but the following strategy avoids to generate unnecessary subsets. First, we generate the subsets X of $\Gamma^*(P)$ such that P is simple for $X \cup \{P\}$ (there are 116 such subsets). Then, for those sets, we "complete" them with all possible subsets of $\Gamma^2(P) \setminus \Gamma(P)$ (there are 2^{16} such subsets). In this way, only the necessary $9,175,040$ subsets are generated and tested.

The same is done for the different kinds of crucial cliques. Let us take just one example, the other kinds of crucial cliques are managed in a similar way. Let P_1 and P_2 be two strongly adjacent pixels belonging to the same row, let $\mathcal{X}_3$ be the set of all the subsets of $\Gamma^2(P_1) \cup \Gamma^2(P_2)$ which contain both P_1 and P_2. Let $\mathcal{X}_4$ be the set of all the sets X of $\mathcal{X}_3$ such that $\{P_1, P_2\}$ is a crucial clique for X. This can be checked using the masks C and C_2 of Fig. 2. If either P_1 or P_2 belongs to $A(X)$ for any X in $\mathcal{X}_4$, then whatever its input, algorithm A does not remove any crucial clique of this kind in a single step. To avoid generating the $2^{28} = 268,435,456$ sets of $\mathcal{X}_3$, we apply the same strategy as for non-simple pixels and generate only the $50,593,792$ sets of $\mathcal{X}_4$. This has to be done also for P_1 and P_2 belonging to the same column, in the case where algorithm A is not symmetrical. For each of the three remaining kinds of crucial cliques, the number of configurations (regardless of the rotations) is equal to $1,048,576$. On the whole, testing the topological soundness of a thinning algorithm with this procedure takes only a few minutes with an ordinary desktop computer.

To summarize, we have the following property.

Property 4. *Let A be a 25-local thinning algorithm. If all the tests discussed above succeed for A, then whatever the set $X \subset \mathbb{G}^2$, the set $A(X)$ is a crucial retraction of X. Reciprocally, if there exists sets X such that A deletes a non-simple pixel or a crucial clique of X, then the above procedure finds at least one counter-example.*

3 Rutovitz, 1966 [31]

This is, to our best knowledge, the first parallel thinning algorithm ever proposed.

Let P be a pixel in $\mathbb{G}^2$, let $X \subset \mathbb{G}^2$. We say that P *is R-deletable* if the five following conditions hold:
i) $P \in X$
ii) $B(P) \geq 2$
iii) $C(P) = 1$
iv) $P_2 \wedge P_0 \wedge P_4 = 0$ or $C(P_2) \neq 1$
v) $P_2 \wedge P_0 \wedge P_6 = 0$ or $C(P_0) \neq 1$

Algorithm RUT66 (**Input /Output** : set X)
01. **Repeat**
02. $\quad Y \leftarrow$ set of pixels in X which are R-deletable
03. $\quad X \leftarrow X \setminus Y$
04. **Until** $Y = \emptyset$

Remark 5. *Algorithm RUT66 does not preserve topology, as shown by the following counter-example.*

In the configuration of Fig. 3, all the four object pixels are R-deletable. Thus, this entire connected component is deleted by the algorithm. This fact is well known and has been pointed out by several authors. It is also well known that Rutovitz' algorithm may be easily "repaired" by adding the "restoring mask" of Fig. 12(11).

0	0	0	0
0	1	1	0
0	1	1	0
0	0	0	0

Fig. 3. Counter-example for algorithm RUT66.

4 Pavlidis 1981 [27,28]

In [27,28], T. Pavlidis presents a parallel thinning algorithm with several variants. With one of these variants, a theorem is stated which says that a perfect reconstruction of the original object may be achieved from the labeled skeleton. We show that the theorem is false and that the proposed proof is incomplete.

We will use, as much as possible, the same vocabulary and notations as in [28].

The *contour* of a set of pixels X is defined as the set of pixels in X which have at least one strong neighbor not in X. In the following illustrations, the pixels which are not in X, called *background pixels*, will be given the value 0 ; the contour pixels will be given the value 2 ; and the pixels which belong to X and which are not contour pixels will be given the value 1.

A contour pixel is called *multiple* [1] if it satisfies one of the four following conditions.
(a) It has at most one nonzero neighbor.
(b) Its neighborhood conforms to either of the patterns shown in Fig. 4(a,b), or those obtained from them by rotations of multiples of $\pi/2$, where at least one of each group of pixels marked with A or B must be nonzero, and where pixels marked D may have any value.
(c) It has no neighbor labeled 1.
(d) Its neighborhood conforms to the pattern shown in Fig. 4(c), or those obtained from it by rotations of multiples of $\pi/2$, where at least one of each pair of pixels marked with A or B or C must be nonzero. If both pixels labeled C are nonzero, then the values of pixels labeled A and B can be anything.

A *corner pixel* if one whose neighborhood conforms to the pattern shown in Fig. 4(d), or those obtained from it by rotations of multiples of $\pi/2$, where the pixel labeled X must be nonzero.

[1] In [27,28] this definition corresponds to pixels which are either multiple or "tentatively multiple". This is the condition which is used for the thinning algorithm allowing reconstruction.

A	A	A
0	P	0
B	B	B

(a)

D	D	D
D	P	0
D	0	2

(b)

A	A	C
0	2	2
B	B	C

(c)

0	0	0
0	P	2
0	2	X

(d)

Fig. 4. (a,b,c,d): Patterns used to define the thinning algorithm PAV81.

Algorithm PAV81 (**Input** /**Output** : set X)
Repeat
    $F \leftarrow$ set of all contour pixels of X
    $M \leftarrow$ set of all pixels in F
    which are either multiple pixels or corner pixels
    $Y = F \setminus M$
    $X \leftarrow X \setminus Y$.
Until $Y = \emptyset$

Property 6. *For any subset X of $\mathbb{G}^2$, the result of PAV81 after one step of execution is a crucial retraction of X.*

At the end the algorithm, the skeletal pixels (that is, the remaining ones) are labelled with the number of the iteration at which they first appeared as a contour pixel. The label of a pixel p will be denoted by λ_p.

We recall the notions of 4-distance (or city block distance) and 4-ball in order to have a simpler definition of reconstruction, which is introduced in [27,28] as an algorithm.

Let $d_4(x, y)$ denote the 4-distance between pixels x and y, that is, $d_4(x, y) = |y_1 - x_1| + |y_2 - x_2|$, where x_1, x_2 (resp. y_1, y_2) denote the coordinates of pixel x (resp. y). Let $B_4(x, r)$ denote the 4-ball of center x and radius r with respect to the distance d_4, that is, $B_4(x, r) = \{y \mid d_4(x, y) < r\}$.

Claim ([28], theorem 1). Let X be the original object, S the skeleton obtained by the above thinning algorithm. The labeled skeleton allows perfect reconstruction of the original image, in other words,

$$\bigcup_{p \in S} B_4(p, \lambda_p) = X$$

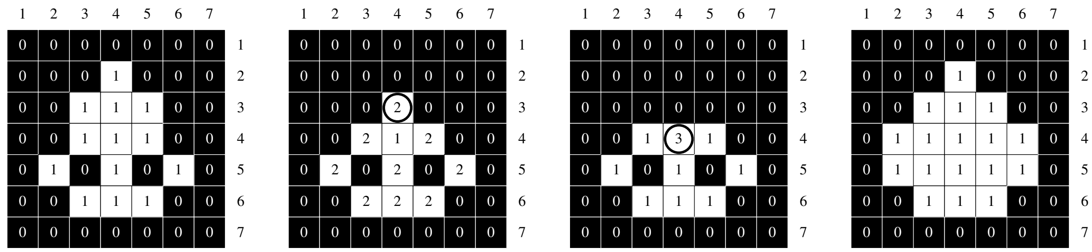
The proof given in [28] uses the two following lemmas, and concludes without other arguments that the theorem holds.

Lemma 1 *During the thinning process, a deletable pixel always has a strong neighbor that remains in the set.*

Lemma 2 *If a pixel is placed in the skeleton after the first iteration, then all its four strong neighbors belonged to the set in the previous iteration.*

We can see with the following counter-example that these two lemmas are indeed not sufficient to prove the theorem.

Remark 7. *The previous claim is false, as proved by the following counter-example (Fig. 5).*

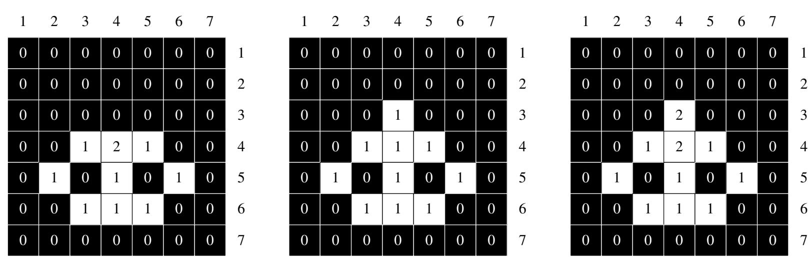


(a) (b) (c) (d)

Fig. 5. (a): Original image. (b): Intermediate result after the first thinning step, with contour pixels labeled 2. (c): Final skeleton with labels. (d): Reconstructed object.

The problem occurs at step 2, when the pixel x circled in Fig. 5(b) is examined. This pixel does not satisfy any condition for being a multiple pixel, thus it is not retained in the skeleton. Consequently, its south neighbor y (circled in Fig. 5(c)) which is a skeleton pixel, will be labeled by step number 3. It will thus generate a ball of radius 3 during the reconstruction. Nevertheless, x satisfies lemma 1 and y satisfies lemma 2.

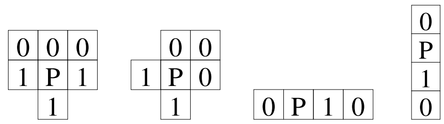
Even if pixel y is labeled by its distance to the background (that is, 2), the reconstruction is still not correct (see Fig. 6(a,b)). A correct result, allowing exact reconstruction, is shown in Fig. 6(c).



(a) (b) (c)

Fig. 6. (a): Skeleton with pixels labeled by their distance to the background. (b): Object reconstructed from (a). (c): Correct labeled skeleton allowing exact reconstruction.

5 Chin, Wan, Stover and Iverson, 1987 [8]



(1) (2) (3) (4)

Fig. 7. Masks for the Chin et al.'s algorithm. (1, 2): thinning masks (with all their $\pi/2$ rotations). (3, 4): restoring masks.

Algorithm CWSI87 (**Input** /**Output** : set X)

```
Repeat
    Y ← set of pixels in X which match anyone of the thinning masks
        but no restoring mask of Fig. 7
    X ← X \ Y
Until Y = ∅
```

Property 8. *For any subset X of $\mathbb{G}^2$, the result of CWSI87 after one step of execution is a crucial retraction of X.*

6 Holt, Stewart, Clint and Perrott 1987 [16]

The following description of the Holt et al.'s algorithm is borrowed from [14] (see Sec. 8).

Let P be a pixel in $\mathbb{G}^2$, let $X \subset \mathbb{G}^2$. We say that P is *deletable* if $P \in X$, $1 < B(P) < 7$ and $D(P) = 1$. We say that P is *H-deletable* if P is deletable and none of the following conditions hold:
i) $P_2 = P_6 = 1$ and P_0 is deletable
ii) $P_0 = P_4 = 1$ and P_6 is deletable
iii) P_0, P_7 and P_6 are deletable

Algorithm H87 (**Input** /**Output** : set X)

```
Repeat
    Y ← set of pixels in X which are H-deletable
    X ← X \ Y
Until Y = ∅
```

Property 9. *For any subset X of $\mathbb{G}^2$, the result of H87 after one step of execution is a crucial retraction of X.*

7 Zhang and Wang, 1988 [33]

Let P be a pixel in $\mathbb{G}^2$, let $X \subset \mathbb{G}^2$. We say that P *is ZW-deletable* if the five following conditions hold:
i) $P \in X$
ii) $2 \leq B(P) \leq 6$
iii) $C(P) = 1$
iv) $P_2 \wedge P_0 \wedge P_4 = 0$ or $P_{12} = 1$
v) $P_2 \wedge P_0 \wedge P_6 = 0$ or $P_8 = 1$

Algorithm ZW88 (**Input /Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which are ZW-deletable
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Remark 10. *Algorithm ZW88 does not preserve topology, as shown by the same counter-example as for Rem. 5.*

8 Hall 1989 [14]

In [14], R. Hall proposes a variant of algorithm H87, that we call here H89, and proves the topological soundness of both algorithms, using combinatorial arguments.

Algorithm H89 is similar to H87, just replacing "$1 < B(P) < 7$" by "$2 < B(P) < 7$" in the definition of a deletable pixel, with the aim of preserving some "diagonal branches".

Property 11. *For any subset X of $\mathbb{G}^2$, the result of H89 after one step of execution is a crucial retraction of X.*

9 Wu and Tsai, 1992 [32]

Algorithm WT92 (**Input /Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which match anyone of the masks of Fig. 8
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Remark 12. *Algorithm WT92 does not preserve topology, as shown by the same counter-example as for Rem. 5.*

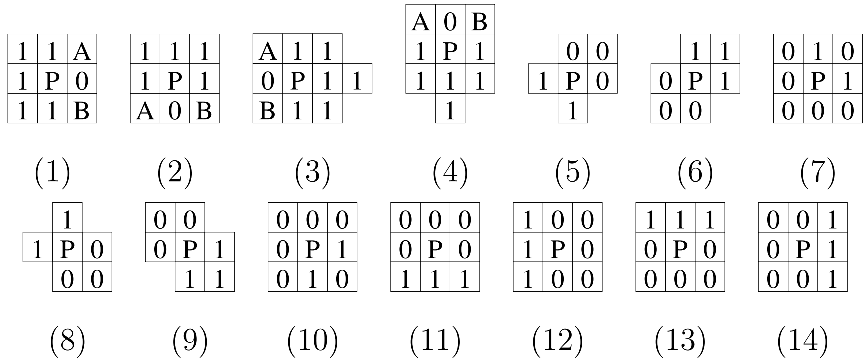


Fig. 8. Masks for the Wu and Tsai's algorithm. For masks 1, 2, 3 and 4, at least one of the pixels A, B must be in X.

It can be seen that each one of the four pixels of Fig. 3 can be matched by one of the masks (5), (6), (8) or (9).

10 Guo and Hall 1992 [13]

Let P be a pixel in $\mathbb{G}^2$, let $X \subset \mathbb{G}^2$. We define the following boolean expressions:
$G(P) = P_0 \wedge P_2 \wedge P_4 \wedge P_6$
$L(P) = [(\overline{P_2} \wedge P_6 \wedge \overline{P_{20}}) \wedge (P_1 \vee P_0 \vee P_7) \wedge (P_5 \vee P_4 \vee P_3)] \vee (\overline{P_0} \wedge P_4 \wedge \overline{P_{16}})$

We say that P *is GHa-deletable* if the four following conditions hold:
i) $D(P) = 1$
ii) $G(P) = 0$
iii) $B(P) > 2$
iv) $L(P) = 0$

Algorithm GH92a (**Input** /**Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which are GHa-deletable
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

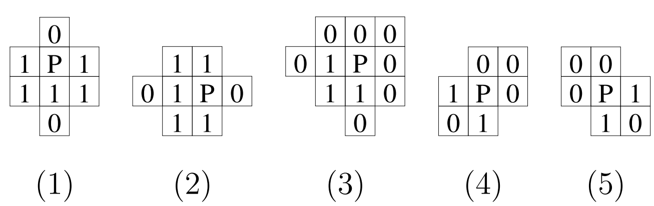


Fig. 9. Masks for the Guo and Hall's algorithms.

We say that P *is GHb-deletable* if the four following conditions hold:
i) $D(P) = 1$
ii) $G(P) = 0$

iii) $B(P) > 2$
iv) The neighborhood of P does not match any of the masks (1, 2, 3) in Fig. 9

We say that P *is GHc-deletable* if P is GHb-deletable or if the neighborhood of P matches either of the masks (4, 5) in Fig. 9.

Algorithms GH92b and GH92c are similar to GH92a, just replacing "GHa-deletable" by "GHb-deletable" or "GHc-deletable", respectively.

Property 13. *For any subset X of $\mathbb{G}^2$, the results of GH92a, GH92b and GH92c after one step of execution are crucial retractions of X.*

11 Jang and Chin 1992 [17]

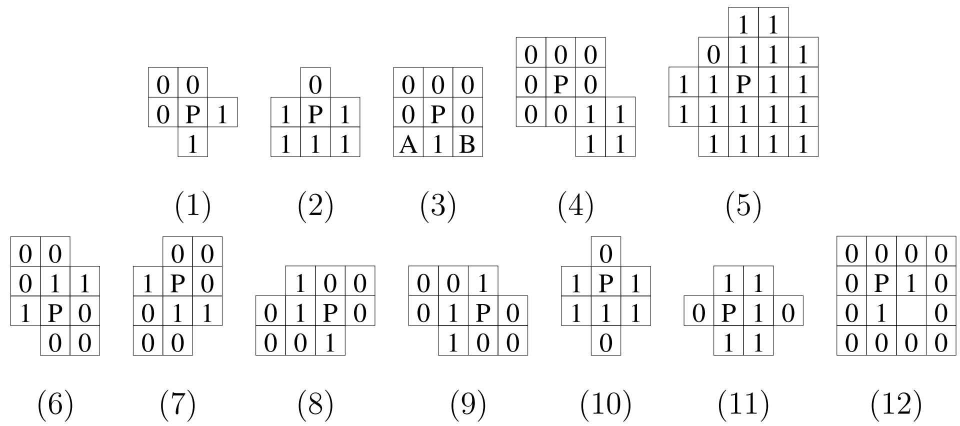


Fig. 10. Masks for the Jang and Chin's JC92 algorithm. (1, 2, 3, 4, 5): thinning masks (with all their $\pi/2$ rotations). (6, 7, 8, 9, 10, 11, 12): restoring masks (12 with all its $\pi/2$ rotations). For mask 3, at least one of the pixels A, B must be in X.

Algorithm JC92 (**Input /Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which match anyone of the thinning masks
        but no restoring mask of Fig. 10
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Remark 14. *Algorithm JC92 does not preserve topology, as shown by the following counter-example.*

In the configuration of Fig. 11, it can be seen that the three pixels which form a connected component may be deleted (the "corner" pixel matches mask 1, the other two pixels match some rotations of mask 3, and none of these three pixels match any restoring mask). Thus, this entire connected component is deleted by the algorithm.

0	0	0	0
0	1	1	0
0	1	0	0
0	0	0	1

Fig. 11. Counter-example for algorithm JC92.

12 Jang and Chin 1993 [18]

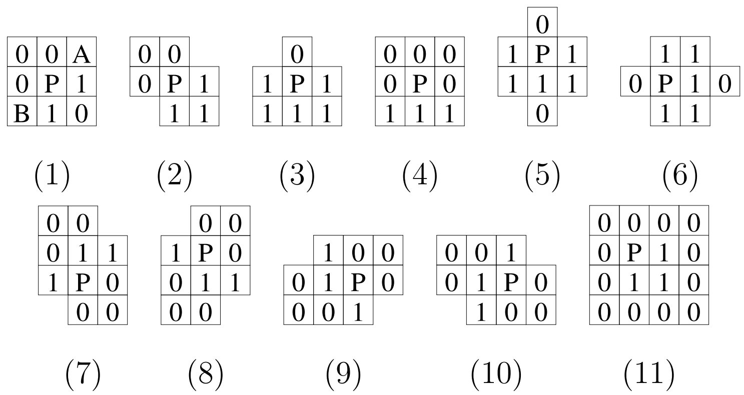


Fig. 12. Masks for the Jang and Chin's JC93 algorithm. (1, 2, 3, 4): thinning masks (with all their $\pi/2$ rotations). (5, 6, 7, 8, 9, 10, 11): restoring masks. For mask 1, at least one of the pixels A, B must be in X.

Let $X \subset \mathbb{G}^2$, let $x \in X$, let $r \in \mathbb{N}$. We say that the ball $B_4(x, r)$ [see Sec. 4] is *maximal for* X if $B_4(x, r) \subseteq X$ and if there is no other ball included in X which contains $B_4(x, r)$.

The *medial axis of* X is the set of the centers of all the maximal balls for X.

Algorithm JC93 (**Input /Output** : set X)
$A \leftarrow$ medial axis of X
Repeat
  $Y \leftarrow$ set of pixels in X which match anyone of the thinning masks
    but no restoring mask of Fig. 12
  $Y \leftarrow Y \setminus A$
  $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Property 15. *For any subset X of $\mathbb{G}^2$, the result of JC93 after one step of execution is a crucial retraction of X.*

13 Eckhardt and Maderlechner 1993 [10]

A pixel in X having all its four strong neighbors in X is an *interior pixel*, a pixel in X which is not interior is a *boundary pixel*. A boundary pixel which has an interior pixel as strong neighbor is called an *inner boundary pixel*. A pixel P in X is termed *simple* if it is a boundary pixel and if there exists exactly one strong connected component of pixels of X in the neighborhood of P which is strongly connected to P. An inner boundary pixel P is termed *perfect* if there exists a strong neighbor $\Gamma_i(P)$ of P which is interior and such that $\Gamma_j(P) \notin X$, with $j = (i+4) \bmod 8$.

Algorithm EM93 (**Input** /**Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which are both simple and perfect
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Property 16. *For any subset X of $\mathbb{G}^2$, the result of EM93 after one step of execution is a crucial retraction of X.*

14 Choy, Choy and Siu 1995 [9]

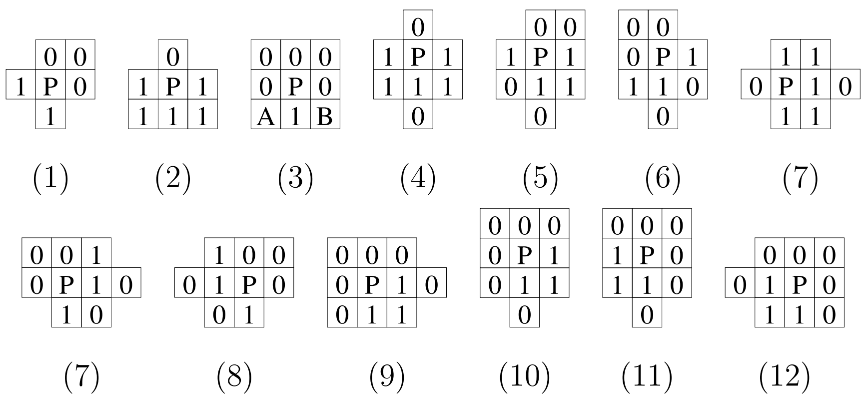


Fig. 13. Masks for the Choy, Choy and Siu's algorithm. (1, 2, 3): thinning masks (with all their $\pi/2$ rotations). (4, 5, 6, 7, 8, 9, 10, 11, 12): restoring masks. For mask 3, at least one of the pixels A, B must be in X.

Algorithm CCS95 (**Input** /**Output** : set X)
Repeat
    $Y \leftarrow$ set of pixels in X which match anyone of the thinning masks
        but no restoring mask of Fig. 13
    $X \leftarrow X \setminus Y$
Until $Y = \emptyset$

Remark 17. *Algorithm CCS95 does not preserve topology, as shown by the following counter-example.*

In the configuration of Fig. 14, each one of the three object pixels matches some rotation of mask (1) or (3) (see Fig. 13), and it does not match any restoring mask since any mask in this set has at least four object pixels. Thus, this entire connected component is deleted by the algorithm.

0	0	0	0
0	1	0	0
0	1	1	0
0	0	0	0

Fig. 14. Counter-example for algorithm CCS95.

15 Bernard and Manzanera, 1999 [2]

(1)

		1	
0	P	1	1
		1	

(2)

0	0		
0	P	1	
	1	1	1
		1	

(3)

P	0
0	1

Fig. 15. Masks for the Bernard and Manzanera's algorithm. (1, 2): thinning masks (with all their $\pi/2$ rotations). (3): restoring mask (with all its $\pi/2$ rotations).

Algorithm BM99 (**Input /Output** : set X)

```
Repeat
    Y ← set of pixels in X which match anyone of the thinning masks
        but no restoring mask of Fig. 15
    X ← X \ Y
Until Y = ∅
```

Property 18. *For any subset X of $\mathbb{G}^2$, the result of BM99 after one step of execution is a crucial retraction of X.*

16 Summary of results and discussion

To summarize, the algorithms proposed by T. Pavlidis in 1981 [27,28], by R.T. Chin, H.K. Wan, D.L. Stover and R.D. Iverson in 1987 [8], by C.M. Holt, A. Stewart, M. Clint and R.D. Perrott in 1987 [16], by R.W. Hall in 1989 [14], by Z. Guo and R.W. Hall in 1992 [13] (3 variants), by B.K. Jang and R.T. Chin in 1993 [18], by U. Eckhardt and G. Maderlechner in 1993 [10], and by T. Bernard and A. Manzanera in 1999 [2] all produce a crucial retraction after a single step of execution. Consequently, they all "preserve topology".

On the other hand, the algorithms proposed by D. Rutovitz in 1966 [31], by Y.Y. Zhang and P.S.P. Wang in 1988 [33], by R.Y. Wu and W.H. Tsai in 1992 [32], by B.K. Jang and R.T. Chin in 1992 [17], and by S.S.O. Choy, C.S.T. Choy and W.C. Siu in 1995 [9] may produce a result which has not the same topology as the input.

Fig. 16 shows the results of some of these algorithms on a simple shape. To make a more detailed comparison, we consider also two other shapes (see

Fig. 17). In particular, we mention the number of pixels, as well as the number of medial axis pixels (see Sec. 12) in the skeletons.

First, notice that some of these algorithms are clearly not aimed at containing all medial axis pixels. It is the case of [Chin, Wan et al. 1987], [Holt et al. 1987], [Hall 1989] and [Guo and Hall 1992] which are not symmetrical and thus produce thinner skeletons than symmetrical algorithms. Nevertheless, it is interesting to observe that the three last algorithms preserve many more medial axis pixels than the first one.

The algorithm [Pavlidis 1981] is the "reconstructing" variant proposed in [27,28]. We can see that it indeed preserves all medial axis pixels for shapes (1,2) but not for shape (3) (see also Sec. 4). Nevertheless, very few medial axis pixels are missing.

The algorithm [Eckhardt and Maderlechner 1993] does also preserve almost all medial axis pixels in these three shapes.

The algorithm [Jang and Chin 1993] does preserve the medial axis in all cases. This is not a surprise since, in this algorithm, the medial axis is computed beforehand and used as a constraint set during the thinning.

Some variants of [Bernard and Manzanera 1999] are studied in [26], with respect to certain metrical properties. In particular, the role of mask (2) of Fig. 15 is to eliminate "corner" configurations, with the aim of enhancing rotation invariance. Indeed with this set of masks, 8-balls and 4-balls, as well as a more general class of balls called *fuzzy balls* in [26], are reduced to one pixel by the thinning algorithm. It may be seen that the algorithm using only masks (1) and (3) is precisely the algorithm [Eckhardt and Maderlechner 1993], as noted in [26].

References

[1] C. Arcelli, L.P. Cordella, and S. Levialdi. Parallel thinning of binary pictures. *Electronic Letters*, 11(7):148–149, 1975.

[2] T.M. Bernard and A. Manzanera. Improved low complexity fully parallel thinning algorithm. In *Proceedings 10th International Conference on Image Analysis and Processing (ICIAP'99)*, 1999.

[3] G. Bertrand. On p-simple points. *Comptes Rendus de l'Académie des Sciences, Série Math.*, I(321):1077–1084, 1995.

[4] G. Bertrand. Sufficient conditions for 3d parallel thinning algorithms. In *SPIE Vision Geometry IV*, volume 2573, pages 52–60, 1995.

[5] G. Bertrand. On critical kernels. Technical Report IGM2005-5, Institut Gaspard-Monge, Université de Marne-la-Vallée, 2005.

[6] G. Bertrand and M. Couprie. Two-dimensional parallel thinning algorithms based on critical kernels. Technical Report IGM2006-2, Institut Gaspard-Monge, Université de Marne-la-Vallée, 2006.

[7] J. Burguet and R. Malgouyres. Strong thinning and polyhedic approximation of the surface of a voxel object. *Discrete applied mathematics*, 125:93–114, 2003.

[8] R.T. Chin, H.K. Wan, D.L. Stover, and R.D. Iverson. A one-pass thinning algorithm and its parallel implementation. *CVGIP*, 40(1):30–40, October 1987.

[9] S.S.O. Choy, C.S.T. Choy, and W.C. Siu. New single-pass algorithm for parallel thinning. *CVIU*, 62(1):69–77, July 1995.

[10] U. Eckhardt and G. Maderlechner. Invariant thinning. *PRAI*, 7:1115–1144, 1993.

[11] C-J. Gau and T.Y. Kong. Minimal non-simple sets in 4d binary pictures. *Graphical Models*, 65:112–130, 2003.

[12] P. Giblin. *Graphs, surfaces and homology.* Chapman and Hall, 1981.

[13] Z. Guo and R.W. Hall. Fast fully parallel thinning algorithms. *CVGIP*, 55(3):317–328, May 1992.

[14] R.W. Hall. Fast parallel thinning algorithms: Parallel speed and connectivity preservation. *CACM*, 32(1):124–131, January 1989.

[15] R.W. Hall. Tests for connectivity preservation for parallel reduction operators. *Topology and its Applications*, 46(3):199–217, 1992.

[16] C.M. Holt, A. Stewart, M. Clint, and R.H. Perrott. An improved parallel thinning algorithm. *CACM*, 30(2):156–160, February 1987.

[17] B.K. Jang and R.T. Chin. One-pass parallel thinning: Analysis, properties, and quantitative evaluation. *PAMI*, 14(11):1129–1140, November 1992.

[18] B.K. Jang and R.T. Chin. Reconstructable parallel thinning. *PRAI*, 7:1145–1181, 1993.

[19] T. Y. Kong. On the problem of determining whether a parallel reduction operator for n-dimentional binary images always preserves topology. In *Procs. SPIE Vision Geometry 2*, volume 2060, pages 69–77, 1993.

[20] T. Y. Kong. On topology preservation in 2-d and 3-d thinning. *International Journal on Pattern Recognition and Artificial Intelligence*, 9:813–844, 1995.

[21] T. Y. Kong and C-J. Gau. Minimal non-simple sets in 4-dimensional binary images with (8-80)-adjacency. In *Procs. Int. Workshop on Combinatorial Image Analysis*, pages 318–333, 2004.

[22] C. Lohou and G. Bertrand. A 3d 12-subiteration thinning algorithm based on p-simple points. *Discrete applied mathematics*, 139:171–195, 2004.

[23] C. Lohou and G. Bertrand. A 3d 6-subiteration curve thinning algorithm based on p-simple points. *Discrete applied mathematics*, 151:198–228, 2005.

[24] C-M. Ma. *Connectivity preserving transformation of digital images: theory and applications.* PhD thesis, City University of New York, 1994.

[25] C.M. Ma. On topology preservation in 3d thinning. *CVGIP*, 59(3):328–339, May 1994.

[26] A. Manzanera and T.M. Bernard. Metrical properties of a collection of 2d parallel thinning algorithms. In *Proceedings International Workshop on Combinatorial Image Analysis*, volume 12 of *Electronic Notes on Discrete Mathematics*. Elsevier Science, 2003.

[27] T. Pavlidis. A flexible parallel thinning algorithm. In *Proc. IEEE Comput. Soc. Conf. Pattern Recognition, Image Processing*, pages 162–167, 1981.

[28] T. Pavlidis. An asynchronous thinning algorithm. *CGIP*, 20(2):133–157, October 1982.

[29] C. Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Discrete Applied Mathematics*, 21(1):67–79, 1988.

[30] A. Rosenfeld. Connectivity in digital pictures. *Journal of the Association for Computer Machinery*, 17:146–160, 1970.

[31] D. Rutovitz. Pattern recognition. *Journal of the Royal Statistical Society*, 129:504–530, 1966.

[32] R.Y. Wu and W.H. Tsai. A new one-pass parallel thinning algorithm for binary images. *PRL*, 13:715–723, 1992.

[33] Y.Y. Zhang and P.S.P. Wang. A modified parallel thinning algorithm. In *ICPR88*, pages 1023–1025, 1988.

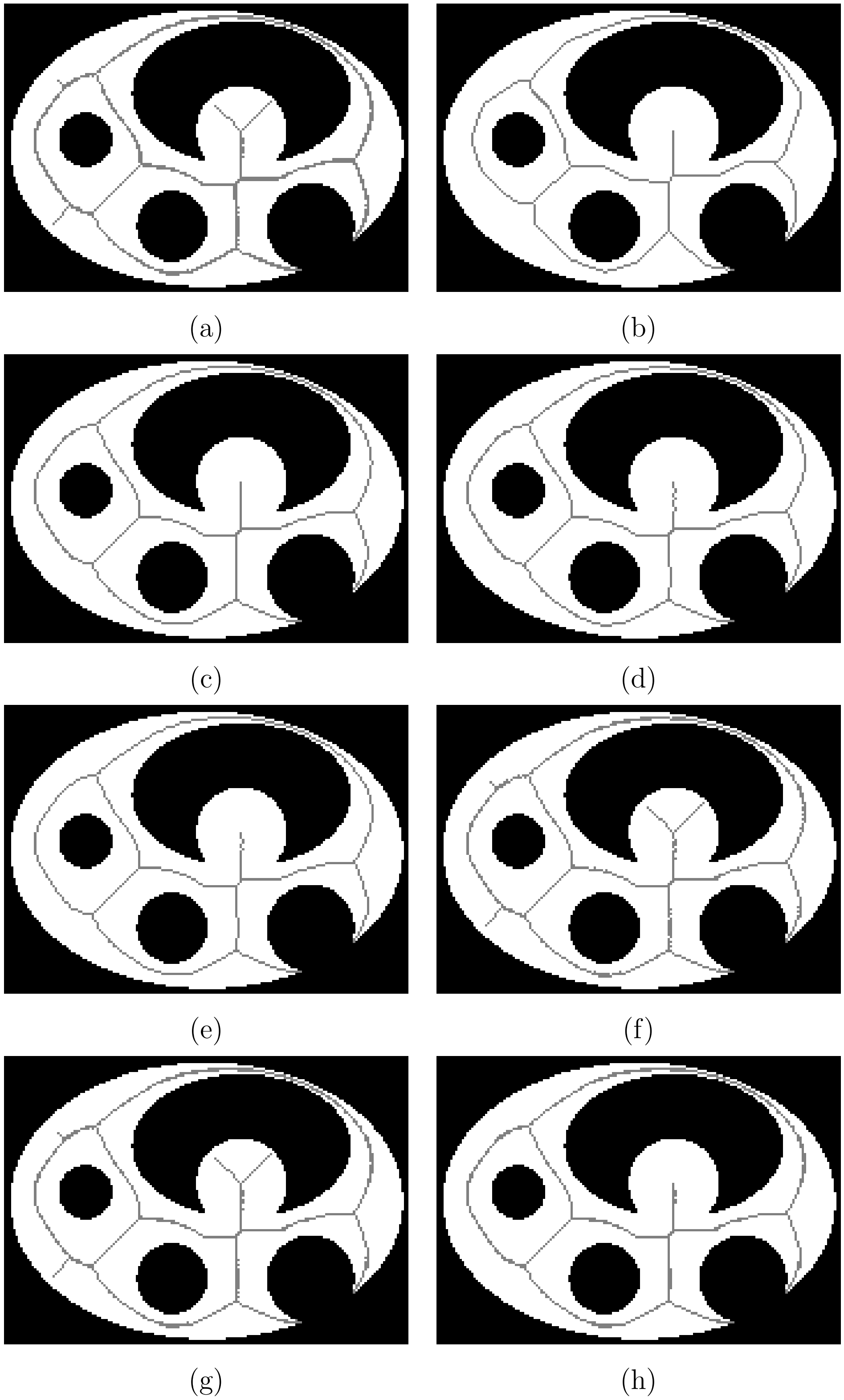
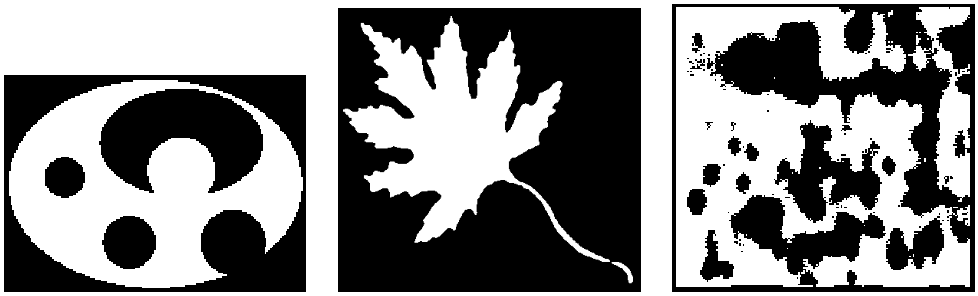


Fig. 16. (a): Pavlidis 1981. (b): Chin, Wan et al. 1987. (c): Holt et al. 1987. (d): Hall 1989. (e): Guo and Hall (3) 1992. (f): Jang and Chin 1993. (g): Eckhardt and Maderlechner 1993. (h): Bernard and Manzanera 1999. The results of Guo and Hall (1,2) on this shape are visually very close to (e) and are thus not displayed here.



(1) (2) (3)

Fig. 17. Three shapes for the comparison of thinning algorithms.

Algorithm	Sym.	N_1	A_1	N_2	A_2	N_3	A_3
Medial axis (reference)	Yes		564		1359		2178
Pavlidis 1981	Yes	847	564	2829	1359	4241	2172
Chin et al. 1987	No	544	153	1572	334	3057	778
Holt et al. 1987	No	590	466	1713	1079	2780	1444
Hall 1989	No	591	467	1773	1103	3060	1557
Guo, Hall 1992 (a)	Yes	658	484	1993	1122	3508	1903
Guo, Hall 1992 (b)	No	591	468	1775	1104	3264	1863
Guo, Hall 1992 (c)	No	560	437	1664	993	3149	1750
Jang, Chin 1993	No	704	564	2394	1359	3787	2178
Eckhardt, Maderlechner 1993	Yes	724	564	2434	1359	3895	2171
Bernard, Manzanera 1999	Yes	678	534	1929	1219	3528	2018

Fig. 18. Comparison of thinning algorithms. The column "Sym." indicates the symmetrical algorithms. N_i: number of pixels in the skeleton. A_i: number of pixels of the skeleton which belong to the medial axis. The index i refers to the shape number in Fig. 17.