

MB: a coherent collection of 2D parallel thinning algorithms

Antoine Manzanera, Thierry Bernard

▶ **To cite this version:**

Antoine Manzanera, Thierry Bernard. MB: a coherent collection of 2D parallel thinning algorithms. [Research Report] ENSTA ParisTech. 2002. hal-01222700

HAL Id: hal-01222700
https://hal.science/hal-01222700

Submitted on 30 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ENSTA/LEI - Technical Report

LEI/AVA-02-002

MB: a coherent collection of 2D parallel thinning algorithms.

Antoine Manzanera, Thierry M. Bernard

Abstract

We present new thinning algorithms to compute the digital skeletons of binary 2D images. Starting from a physical characterization of the thinning process, we show that the mere incorporation of discrete topology preservation conditions leads to the Boolean definition of different parallel thinning algorithms, depending on the topological and metrical properties that can be basically desired in the square grid. The complete Boolean definition of every algorithm is provided through a coherent set of removing and non-removing conditions. For every algorithm, logic minimization is addressed, and the implementation on a Boolean cellular automata grid is proposed. Beyond the implementation, we stress the advantages of Boolean concision in terms of complexity reduction. Some results are shown and discussed.

1 Introduction

A *thinning algorithm* is a process that peels iteratively a binary shape while preserving its geometrical and topological features. The relaxation of such algorithm leads to a *skeleton*, which is a classical shape representation scheme.
The popularity of thinning algorithms comes mainly from the regularity and localness of the operations, that make them easy to implement, cheap in terms of computational resources, and well suited to data parallel implementations. In compliance with the digital geometry framework, they make the early assumption of the discrete nature of computers.
Nevertheless, thinning approaches have considerable drawbacks:

- their ill-posedness rules out a sheer definition of *the* digital skeleton. Algorithmic definitions are proposed instead, but they can be very numerous, as witnessed by a huge bibliography [6], [7]. Thinning algorithms are defined by a set of rules which are often not justified, and whose properties are often not studied but experimentally.

- the metrics of the resulting skeleton depends on the precise nature of the iterations over the discrete mesh. As a consequence, some desirable properties like invariance to rotation and homothety are very difficult - if not impossible - to obtain.

In contrast, here are the features of the collection of thinning algorithms proposed in this paper:

- they are grounded on a physical basis, as they are defined from the discretization of the evolution equation of a monotonous propagating front. The different algorithms are obtained by logic minimization of the topology preservation constraints, in the different connectivity models of the square grid, and for different parallelization schemes (Section 3).
- the computational costs of the algorithms have been precisely quantified (Section 4).
- their metrical properties have been studied, and for some algorithms, the very nature of the median axis has been identified [10].

The paper is organized as follows: In a self-contained fashion, Section 2 recalls the main existing theoretical results about thinning algorithms in two dimensions. Section 3 presents the construction of the family of algorithms, referred to as MB. The proof of validity of every algorithm is provided. Section 4 provides the complete cellular implementation of the algorithms and addresses the problem of their computational complexity. Results and preliminary discussions about a further study of metrical and combinatorial properties in a companion paper [10] are presented in Section 5.

2 Topological issues in 2D thinning

In the continuous framework, i.e. for a subset X of $\mathbb{R}^2$, the skeleton $S(X)$ is a purely metrical notion, coinciding with the *median axis*, which is the locus of the maximal ball centers:

$$S(X) = \bigcup_{r \in \mathbb{R}} \{x \in X; \forall (y, r') \in \mathbb{R}^2 \times \mathbb{R}, B(x,r) \subset B(y,r') \subset X \Rightarrow (x,r) = (y,r')\}$$

Where $B(x,r)$ denotes the ball of center x and radius r. When the metrics is the Euclidean distance δ_e, the skeleton has the following properties: (1) It is invariant with respect to rotation and homothety i.e. for any similitude σ of $\mathbb{R}^2$, $S(\sigma(X)) = \sigma(S(X))$, (2) Its interior is empty: $\overbrace{S(X)}^{\circ} = \emptyset$, (3) If X is an open set, X and $S(X)$ have the same topology (i.e. same number of connected components and same number of holes), and (4) X is reconstructible from $S(X)$ weighted with the distance to the border: $X = \bigcup_{x \in S(X)} B(x, \delta_e(x, X^c))$.

It is straightforward to adapt the notion of median axis in a digital mesh for any discrete distance δ. Furthermore, the discrete median axis is then the local

maxima of the distance to the border:

$$S(X) = \bigcup\{x \in X; (y \in X \delta(x,y) = 1) \Rightarrow \delta(x, X^c) \geq \delta(y, X^c)\}.$$

Nevertheless, the discrete median axis lacks many of the nice properties of its continuous counterpart and, in particular, it does not preserve the topology of the initial shape. For this reason, topological issues must be addressed in the discrete framework.

From now on, we consider the discrete plane $\mathbb{Z}^2$, and (binary) *images* as subsets of $\mathbb{Z}^2$. A point $x \in \mathbb{Z}^2$ is a discrete cell or *pixel*. Pixels are spatially distributed over a *square grid*, where two distinct topologies can be defined: (1) The *4-connectivity*, where a pixel has 4 neighbors, corresponding to the four closest - or direct - neighbors in the cardinal directions {N,E,W,S}. (1) The *8-connectivity*, where a pixel has, in addition to the 4 direct neighbors, 4 other neighbors corresponding to the four diagonal directions {NW,NE,SE,SW}. Considering that two neighbors are distant of 1 unity, these two topologies induce two different metrics, respectively the *4-distance* d_4, and the *8-distance* d_8. If $x = (x_1, x_2)$ and $y = (y_1, y_2)$, then $d_4(x,y) = |x_1 - y_1| + |x_2 - y_2|$ and $d_8(x,y) = \max(|x_1 - y_1|, |x_2 - y_2|)$.

For a given topology and a set X, the relation "is a neighbor of" between pixels in X is a reflexive and symmetric relation called *adjacency*. The transitive closure of adjacency corresponds to the existence of a connected path between two pixels in X ; it is an equivalence relation whose classes are called *connected components* (cc).

The topology preservation issue for the thinning algorithm is the problem of transforming a set X in a set $Y \subset X$ without separating one cc of X in two or more ccs in Y or making vanish a cc of X, and without merging two or more ccs of X^c (the *holes*) in one single cc of Y^c, or making appear a new hole in Y. A clean definition of the notion of hole implies the discrete counterpart of the Jordan's theorem, which stands that a closed single arc (function $\gamma : [0,1] \longrightarrow \mathbb{R}^2$ such that $(x < y \gamma(x) = \gamma(y)) \Leftrightarrow (x = 0 y = 1)$) separates the plane in two disconnected parts: the interior (the hole), and the exterior. To get such a property in the square grid, however, different adjacency relations have to be considered for an image X and for its complementary X^c. So, working in K-connectivity ($K = 4$ or 8) means that we consider the K-connectivity model for the image X (represented by convention with black pixels), and the $\tilde{K}$-connectivity model ($\tilde{K} = 12 - K$) for the complementary X^c (represented by convention with white pixels). We may now recall definitions regarding the discrete topology preservation:

If $A \subset B$, A is *K-homotopic* to B iff: (1) every K-cc of A is contained in one K-cc of B exactly, and (2) every $\tilde{K}$-cc of A^c contains exactly one $\tilde{K}$-cc of B^c.

Let $X \subset A$ X is *$K-$simple* in A iff $A \setminus X$ is *$K-$homotopic* to A. For a pixel $x \in A$, we will write that x is K-simple in A if $\{x\}$ is K-simple in A.

An important result is that the simpleness of a pixel can be determined locally [6]: If we denote by $N_K(x)$ the set of K-neighbors of x in $\mathbb{Z}^2$, $x \in X$ is K-simple in X iff: (1) $(N_K(x) \setminus \{x\}) \cap X$ is a K-connected set, and (2) $N_{\tilde{K}}(x) \cap X^c \neq \emptyset$. An explicit analytical characterization has been proposed through the K-connectivity numbers [14]. We present an equivalent geometrical (or pattern-

based) characterization.

Let (α) be a binary *pattern*, i.e. a couple $\{\alpha_H, \alpha_M\}$ of two disjoint sub-

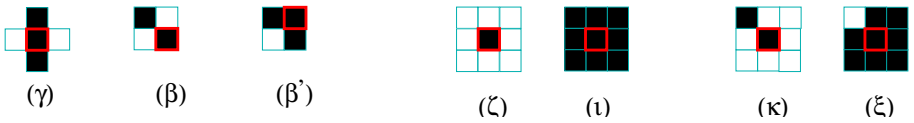


Figure 1: Non simple patterns and exceptions.

sets of $\mathbb{Z}^2$. We say that a pixel x *matches* (α) in X, or $x \in X \ltimes \alpha$, if $\bigcup_{y \in \alpha_H} \{x+y\} \subset X$ and $\bigcup_{y \in \alpha_M} \{x+y\} \subset X^c$. By convention, the elements of α_H (resp. α_M) will be represented by black (resp. white) pixels, and the origin of the pattern will be outlined in red. As the patterns will often be considered up to rotation, we will note that $x \in X \boxtimes \alpha$ when x matches (α) or one of its $\pi/2$ rotated versions, in X. The pattern-based characterization of simpleness is then the following one, using the patterns shown on Figure 1: (1) ["x is not 4-simple in X" OR $x \in X \boxtimes \xi$] *is logically equivalent to* $[x \in (X \boxtimes \gamma) \cup (X \boxtimes \beta') \cup (X \boxtimes \zeta) \cup (X \boxtimes \iota)]$. (2) ["$x$ is not 8-simple in X" OR $x \in X \boxtimes \kappa$] *is logically equivalent to* $[x \in (X \boxtimes \gamma) \cup (X \boxtimes \beta) \cup (X \boxtimes \zeta) \cup (X \boxtimes \iota)]$. This remark, essentially due to C. Arcelli [1], will henceforth be referred to as the "Arcelli's property".

When designing a thinning algorithm, it is desirable to remove a large set of points *in parallel*, either because they must be removed *simultaneously* through a computation on a parallel machine, or because they must be removed *independently* of the scanning order, for the sake of geometry preservation. But a union of simple points is not always a simple set, so the above results are not sufficient to prove the validity of a parallel thinning algorithm.

If a set $P \subset X$ can be ordered in a sequence, called simpleness chain $\{x_0, \ldots, x_n\}$ such that $\forall i \in [0, \ldots, n], x_i$ is simple in $X \setminus \{x_0, \ldots, x_{i-1}\}$, then P is simple. But this property is too weak, because, as proved by C. Ronse [11], any simple set can be ordered in a simpleness chain. A much stronger condition is to impose that *any ordering* of the set P is a simpleness chain. G. Bertrand [2] has shown that such property can be formally proven ; parallel thinning algorithms respecting this condition for any removed set P are called *P-simple*.

C. Ronse has proposed explicit sufficient conditions for a parallel thinning algorithm to preserve the topology, in 4 and 8-connectivity [12]. It has been shown later that these conditions were equivalent to the P-simpleness. We now present these conditions: a parallel reduction algorithm removing in parallel a set of points P from a binary image X preserves the K-connectivity if, for any X and P, the 3 following conditions are met: (1) every pixel $x \in P$ is K-simple, (2) every pair $\{x_1, x_2\} \in P$ of $\tilde{K}$-adjacent pixels (and not 4-adjacent if $K = 4$) is K-simple, and (3) if $K = 8$, P does not contain any cc of X included in a 2×2 block.

We can now present and valid the family of MB thinning algorithms.

3 The MB thinning algorithms

Physically, a thinning algorithm is expected to simulate the behavior of a monotonous propagating front or interface, as in the grass fire analogy [3]. In the continuous framework, the evolution of such interface is usually described by an equation of the form: $\frac{\partial C}{\partial t} = \mathcal{F}\vec{N}$, with $C = (x(t), y(t))$ the parametric form of a closed curve, $\vec{N}$ is the normal to the curve oriented toward the interior, $\mathcal{F}$ is the propagation velocity. $\mathcal{F}$ is a function of (x, y, t) depending on the modeling, with the only constraint $\mathcal{F} > 0$ imposed by the monotonicity.

Of course, the object to be thinned must have some thickness to allow front

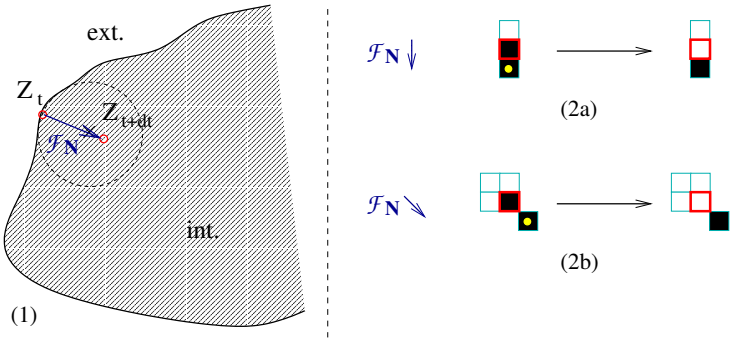


Figure 2: The evolution of a monotonous propagating front (1) and the corresponding discrete evolution rules in the 4-neighborhood (2a), and in the 8-neighborhood (2a and 2b).

propagation toward its interior. Then, the most naive discretization of such evolving interface in the 4-neighborhood is to allow four elementary displacements of the frontier, corresponding to the four cardinal directions of the 4-connectivity, generated by the evolution rule shown on Figure 2(2a) and its $\pi/2$ rotated versions. In the 8-neighborhood four additional elementary displacements are allowed, corresponding to the four diagonal directions of the 8-connectivity, and generated by the evolution rule of Figure 2(2b) with its $\pi/2$ rotated versions.

In Figure 2(2a) and (2b), the black pixels with a yellow dot correspond to points that would be an interior point in the continuous framework. For this reason, we henceforth call them "interior" (within quotes) though it will become clear only gradually what it means in the discrete framework. Actually, we shall now determine the smallest pattern(s) surrounding the "interior" point, such that the above evolution rules become sensible enough to yield adequate thinning algorithms. We shall consider 3 independent thinning options: (1) The front

propagation may follow either the 4 (N,E,W,S) or the 8 (N,NE,E,SE,S,SW,W, NW) cardinal directions. Accordingly, the algorithms will be denoted by suffices "-1" or "-2". (2) Either 4- or 8-connectivity may be considered for the topology preservation. The corresponding algorithms will be denoted respectively "-4" or "-8". (3) Finally an algorithm may be either directional or fully parallel, depending on whether the pixels are removed only from the border of one cardinal direction, or from the whole border in all directions at the same time. This choice is based on metrical considerations as we will see later. The corresponding algorithms will be denoted respectively "-dir" or "-fp". Those three binary choices lead to 8 different algorithms that are presented on Figure 9. We now detail their construction.

We first note that, for any of the conditions above, the so-called "interior" point must be surrounded by at least one black pixel in the 8-neighborhood of the candidate to deletion ; otherwise it would allow the deletion of end-points. For the sake of symmetry, we eventually impose, in the two deletion conditions corresponding to the two evolution laws, that at least one of the two pixels 4-adjacent to the "interior" point in the neighborhood of the candidate to deletion, is black (see Figure 3, to be considered with all its $\pi/2$ rotated versions).

It turns out that this single condition is almost sufficient to provide directional



(1) (2)

Figure 3: In order to prevent the removal of end points, at least one of the two crossed pixels must be black in each condition.

algorithms. Indeed, as guaranteed by the Arcelli property, if a north-border pixel (black pixel whose north neighbor is white) matches pattern (1) of Figure 3, then it is 8-simple (resp. 4-simple), unless it matches pattern (β) (resp. (β')) of Figure 1, or its mirrored version. It is also true for a north-border pixel matching pattern (1) or (2) (or its mirrored version) of Figure 3. The same property is true *mutatis mutandis* considering east, south or west border pixels. The other conditions of the Ronse's theorem are obvious in the case of directional thinning, since the deletion of non-end simple border points in one single direction always preserve the topology [6]. We have thus defined 4 directional MB algorithms, corresponding to the 4 first lines of the table of Figure 9, respectively MBdir1-8, MBdir2-8, MBdir1-4 and MBdir2-4. The table shows a sub-iteration corresponding to the deletion of WEST border points: each algorithm removes the pixels matching one of the patterns contained in the column "removing condition", provided that they do not match one of the patterns contained in the column "non removing condition". The corresponding thinning algorithms are the relaxation of the thinning operators, alternating periodically the four cardinal directions.

In the case of a fully parallel algorithm, some constraints must be added compared with directional patterns. Theses constraints will show up through the different configurations corresponding to conditions (2) and (3) of Ronse's theorem. We are going to show that, in the case of MBfp1-8, all the pixels 4-adjacent to the "interior" point must be black. We will then prove that this characterization of "interior" points as 4-interior pixels can be used for all the fully parallel MB algorithms.

As our purpose is to determine the smallest set around the "interior" point,

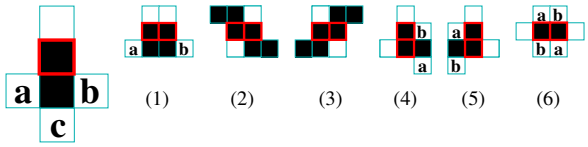


Figure 4: Construction of MBfp1-8: in the case of a fully parallel removal, the three pixels denoted a, b and c must be black. The role of a (resp. b) is to avoid configurations (2) and (4) (resp. (3) and (5)) ; The role of c is to avoid configurations (6).

we begin by considering the 3 closest pixels of the "interior" point (except the candidate to deletion). These are pixels a, b and c of Figure 4. From the construction of MBdir1-8, we already know that the candidate to deletion must not match pattern (β), and that a or b must be black. Figure 4(1)-(6) represents the different configurations of pair of 4-adjacent pixels matching the condition shown on the left (or one of its $\pi/2$ rotated versions), where a or b is black. We see that in the two cases (2) and (3), the pair is not simple. These configurations are made impossible if *both* a and b are black. This stronger condition also avoids configurations (4) and (5), where the deletion of the pair could lead to topology changes. Finally, considering configuration (6), it can be seen that, for any values of a and b, the pair is not simple if no other values are imposed outside the support of the pattern shown on the left. So the only way to address that case without increasing the support is to make this configuration impossible by imposing that pixel c is black too. Finally, if the three pixels a, b and c are black, condition (2) of Ronse's theorem is fulfilled. Condition (3) of the same theorem is then obvious, since a cc contained in a 2×2 block cannot contain a 4-interior point. So the construction of MBfp1-8 is achieved: see line 5 of the table of Figure 9. Note that MBfp1-8 is equivalent to the algorithm proposed by U. Eckhardt et al. in [4]. For this algorithm, and the other fully parallel MBs, the patterns are to be considered with all their $\pi/2$ rotated versions.

The MBfp2-8 algorithm is deduced by taking the same characterization of "interiority" for the patterns corresponding to diagonal propagation: so the deletion condition now corresponds to the two patterns (α_1) and (α_2) of Figure 5, whereas the non-deletion condition is the same as MBfp1-8. We now prove the validity of this algorithm. Condition (1) of Ronse's theorem has already been

proved, since this condition is stronger than the directional counterpart. Condition (3) is obvious. To meet condition (2), we have to consider the different configurations of 4-adjacent pairs that are removed by MBfp2-8. Up to the rotation, there are three possible cases, shown in Figure 6. The first case (both pixels match (α_1)) has been treated already in the construction of MBfp1-8. The second case (both pixels match (α_2)) is just a particular case of the first one. In the third case (the right pixel matches (α_1), the left one (α_2)), if the left pixel is removed first, then the right pixel remain removable by the MBfp2-8 condition, so *a fortiori* the pair is 8-simple. This completes the proof of validity of MBfp2-8.

As in the directional cases, the same deletion conditions can be used to con-

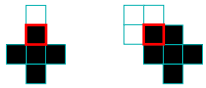


Figure 5: The deletion patterns for the fully parallel algorithms, to be considered with all the $\pi/2$ rotated version: in the directions of the 4-neighborhood (α_1), or in the directions of the 8-neighborhood (α_1 and α_2).

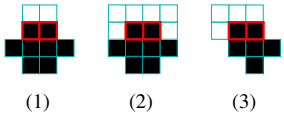


Figure 6: Proof of MBfp2-8: the three possible cases of 4-adjacent pair removed by MBfp2-8.

struct the algorithms in 4-connectivity. Thanks to the Arcelli's property, it can be seen that any pixel matching condition (α_1) or (α_2) (see Figure 5) is 4-simple. We are now going to see what constraints must be added to match condition (2) of Ronse's theorem, first for MBfp1-4, and next for MBfp2-4.

To check condition (2) in the case of MBfp1-4, we have to consider the different configurations of diagonally adjacent pairs that are removed by MBfp1-4. Up to the rotation, there are five possible cases, shown in Figure 7. Let us call the two removable pixels a and b as shown on Figure 7. It is easy to check that the pair $\{a, b\}$ is simple except in two cases: in case (4), and in case (5) when x is black. To avoid those cases, we note that the deletion of the pair $\{a, b\}$ make a β pattern appear in the two cases. The new constraint imposed is then that the deletion of pixels should not make a β pattern appear. Now a β pattern can appear if two diagonally pixels $\{a, b\}$ match (α_1), and the two other pixels in the 2×2 block containing $\{a, b\}$ are black and do not match

(α_1). As there cannot be more than two pixels matching (α_1) in a 2×2 block, it is sufficient to check that the two other pixels are black. That leads to the definition of MBfp1-4, see line 7 in the table of Figure 9. In the non deletion condition, the white pixels with a red dot represent pixels matching the deletion condition (pattern (α_1)). This algorithm is equivalent to the thinning algorithm proposed by L.J. Latecki et al in [8]. To get a thinner result, the authors of [8] proposed to remove one of the two pixels in the critical cases. This correspond to look for appearing β pattern in two directions only, instead of considering all the $\pi/2$ rotated version. We will follow this slight distorsion to isotropy in the description of the algorithm.

Meeting condition (2) of Ronse's theorem for MBfp2-4 brings 3 other con-

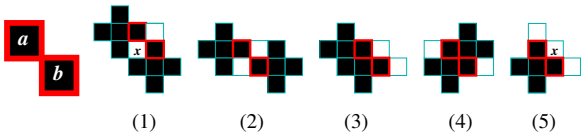


Figure 7: Proof of MBfp1-4: the five possible cases of diagonally adjacent pair removed by MBfp1-4.

figurations, appearing in Figure 8. In the three cases, it is easy to check that the pair $\{a, b\}$ is simple. But there is a notable difference with the previous algorithm, because, unlike MBfp1-4, there can be 3 pixels matching (α_1) or (α_2) in a 2×2 block. In particular, in configuration (5) of Figure 7, x can be a black pixel matching pattern (α_2). In that case, the deletion of the pair $\{a, b\}$ does not modify the topology, if x is removed as well. The definition of MBfp2-4 appears on line 8 of Figure 9. In the non deletion condition, the white pixels with a red dot represent pixels matching the deletion condition, whereas the black pixels with a green square represent black pixels *not* matching the deletion condition. Note that MBfp2-4 does not respect the condition (2) of Ronse's theorem, because in the case of Figure 7(5) with x black, the pair $\{a, b\}$ *is not* simple. Nevertheless, it is contained in a triplet $\{a, b, x\}$ that *is* simple, and then the algorithm preserves the topology. Note that MBfp2-4 is *not a P-simple algorithm*, because the sequences $< a, b, x >$ and $< b, a, x >$ are not simpleness chains.

The table of Figure 9 summarizes the 8 different algorithms that have been constructed ; the different columns give an overview of the different properties that have been detailed or that will be detailed in Section 4. It can be noticed that in the removing conditions corresponding to diagonal propagation, the white pixel diagonally opposed to the "interior" pixel does not appear in the patterns. In fact this pixel is useless in the 8-connected algorithms, because it is redundant with the non-removing condition. In the case of 4-connected skeleton, we follow Latecki et al. by imposing the *well-shapedness* of images, i.e. the input image must have the same topology in 4- or 8-connectivity, and then must not contain

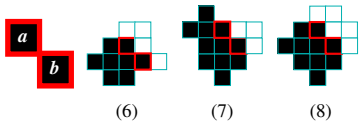


Figure 8: Proof of MBfp2-4: the three other possible cases of diagonally adjacent pair removed by MBfp2-4.

pattern (β). By construction of the MB 4-connected algorithms, pattern (β) cannot appear during the thinning process, and then computation of the angle white pixel of pattern (α_2) becomes pointless.

The physical characterization,together with fundamental constraints had lead

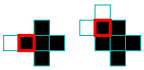
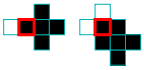
ALGORITHM	*Removing condition*	*Non–removing condition*	*Parallelism*	*Topology preservation*	*Isotropy*	*1–pixel thickness*	*Boolean complexity*	*Support and size*	*P–simpleness*
MBdir1–8			*DIR*	*8*	*NO*	*YES*	28 r	 8 (8)	*YES*
MBdir2–8			*DIR*	*8*	*NO*	*YES*	76 r	 8 (8)	*YES*
MBdir1–4			*DIR*	*4*	*NO*	*YES*	28 r	 7 (8)	*YES*
MBdir2–4			*DIR*	*4*	*NO*	*YES*	60 r	 7 (8)	*YES*
MBfp1–8 *[Eckhardt et al. 93]*			*FP*	*8*	*YES*	*NO*	18 ρ	 13	*YES*
MBfp2–8			*FP*	*8*	*YES*	*NO*	28 ρ	 21	*YES*
MBfp1–4 *[Latecki et al. 95]*			*FP*	*4*	*NO*	*NO*	16 ρ	 23	*YES*
MBfp2–4			*FP*	*4*	*NO*	*NO*	26 ρ	 38	*NO*

Figure 9: The MB family parallel thinning algorithms.

to a family of well defined 2-dimensional skeletons. Now, what are the consequences of the different properties that we have chosen to impose or relax ? What is the relevance of the different alternatives in the MB family ? These questions are formally addressed in [10]. The patterns appearing in Figure 9 show up the conceptual concision of the MB algorithms, which is the fruit of the derivation principle. If this property implies naturally the simplicity of implementation, the advantage in terms of computational complexity remains to

prove. This is done in the following section, where we present the complete cellular implementations of the MB algorithms and discuss their combinatorial properties.

4 Combinatorial properties of MB

This section is dedicated to the presentation of the complete implementations of the different algorithms presented above. The computational model used is a *cellular Boolean massively parallel machine.* The choice of this model is based on the fact that the computation time of the cellular program is exactly *the number of Boolean binary operations performed per pixel.* Although the computation time depends on the implementation, this number is a fundamental area-time complexity measure, because in any data parallel implementation, the Boolean binary operation represents the energy quantum, consumed either at the software or at the hardware level.
It is worth noting that the Boolean minimization in the cellular implementation is not related to the support size of the Boolean function computed [5], as we are going to see some functions with huge supports, computed with a compact cellular program. This is due to the parallel decomposition of functions, which is valuable not only in the cellular model, but also in other computation frameworks, as it will be shown.
We describe now the computation model and the cellular program, that will be given for readability purposes as graphical representations. The cellular machine is a square grid of Boolean processors. Every processor has n bits of digital memory $\{b_0, \ldots, b_{n-1}\}$, and has access to the memory of its 8 closest neighbors, denoted in exponent by the cardinal directions. As elementary operation, every processor can perform one binary Boolean operation between two bits of its own memory or of the memory of one of its closest neighbors: $b_i = op(b_j^{d1}, b_k^{d2})$. Operator op can be the logical AND, OR, XOR, or AND NOT, respectively denoted $\wedge$, $\vee$, $\oplus$ and $\setminus$. Directions $d1$ and $d2$ can be the 8 cardinal directions denoted NW, N, NE, E, SE, S, SW, W, or the ego position denoted "$\cdot$". The cellular programs are graphically represented as shown on Figure 10. The nodes with two entries represent the elementary operations, labeled with the corresponding operator. The rectangles with different inputs labeled with cardinal directions represent the values within a certain neighborhood for the same bit of information. As an example, we are going to detail the program corresponding to MBdir1-4, at the upper left of Figure 10, which is a WEST iteration of the thinning algorithm appearing at line 3 of Figure 9:

$\setminus * \; b_0$: input data $* \setminus$
$b_1 = (b_0^E \setminus b_0^N) \wedge (b_0^{NE} \vee b_0^{SE})$;
$\setminus * \; b_1$: points matching the removing condition $* \setminus$
$b_2 = b_0^{\cdot} \setminus b_0^E$;
$b_3 = b_2^N \vee b_2^S$;
$\setminus * \; b_3$: points matching the non removing condition $* \setminus$

$b_0 = b_0^{\cdot} \setminus (b_1^{\cdot} \setminus b_3^{\cdot})$;
* deletion of the points to be removed *\

All the working processors (which can be the whole grid, or a subset of active processors) compute simultaneously the same sequence corresponding to the above program. The computation time, or Boolean complexity of this iteration, is then the number of elementary operations, i.e. the number of nodes of the corresponding graph. As every iteration examines the 4-contour pixels in one cardinal direction, 4 directional iterations will examine the 8-contour pixels in all directions. The number of iterations of the directional algorithms is then 4 times the radius of the biggest ball of distance d_8 contained in the binary input image. Let us call r this radius ; the Boolean complexity of algorithm MBdir1-4 is then $4 \times 7 \times r = 28r$.
The cellular implementation of the other directional algorithms are shown on Figure 10. The Boolean complexity of every algorithm appears in the table of Figure 9. The support of the operator is also shown in the table ; for the directional algorithms, the size of the support appearing between parentheses is the support of the whole set of 4 directional iterations.
The implementations of the 8-connected fully parallel thinning algorithms are

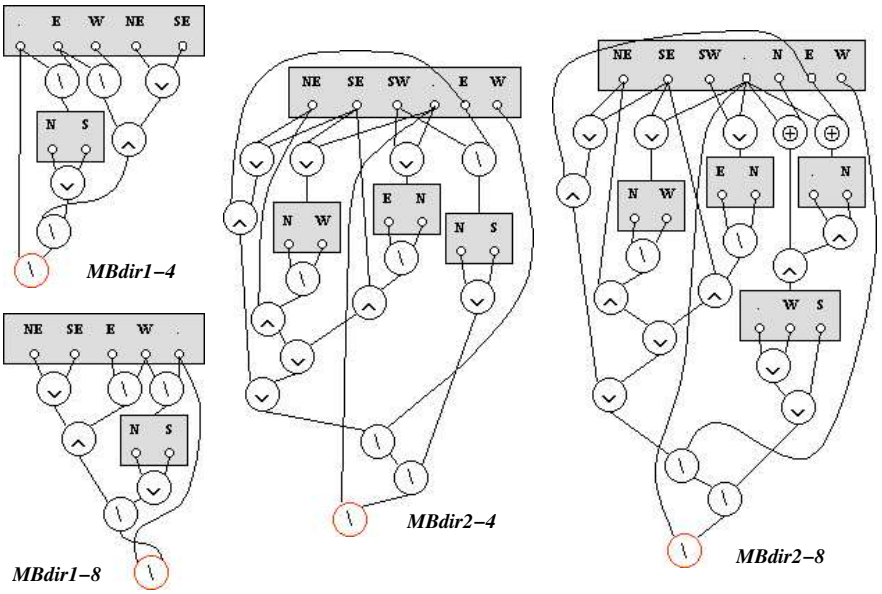


Figure 10: Cellular implementations of the directional MB algorithms (WEST iteration).

detailed in Figure 11. The four operators shown on top are the sub-operators of

the two algorithms, appearing at the bottom. From left to right: (1) computation of the 4-interior points (erosion by a d_4 ball of radius 1), (2) computation of points matching pattern (α_1), i.e. the points having a 4-neighbor 4-interior, and a 4-neighbor white in the opposite direction, (3) computation of points matching pattern (α_2), i.e. the points having a diagonal neighbor 4-interior, and the two 4-neighbors white in the opposite directions. (2) computation of points matching pattern (β) i.e. the non removing condition. The cellular programs of MBfp1-8 and MBfp2-8, shown at the bottom, are then easy to understand. Note that in the sub-operator denoted $\alpha_1 \vee \alpha_2$, the operator E_4 is only computed once. As every iteration examines the 4-contour pixels in all cardinal directions, the number of iterations of the fully parallel algorithms is then the radius of the biggest ball of distance d_4 contained in the binary input image. Let us call ρ this radius ; the Boolean complexity of algorithm MBdir1-8 (resp. MBdir2-8) is then 18ρ (resp. 28ρ). The shape and size of the supports are shown on Figure 9. According to [5], MBdir1-8 has the smallest support with central symmetry for a fully parallel thinning algorithm.

The implementations of the 4-connected fully parallel thinning algorithms are

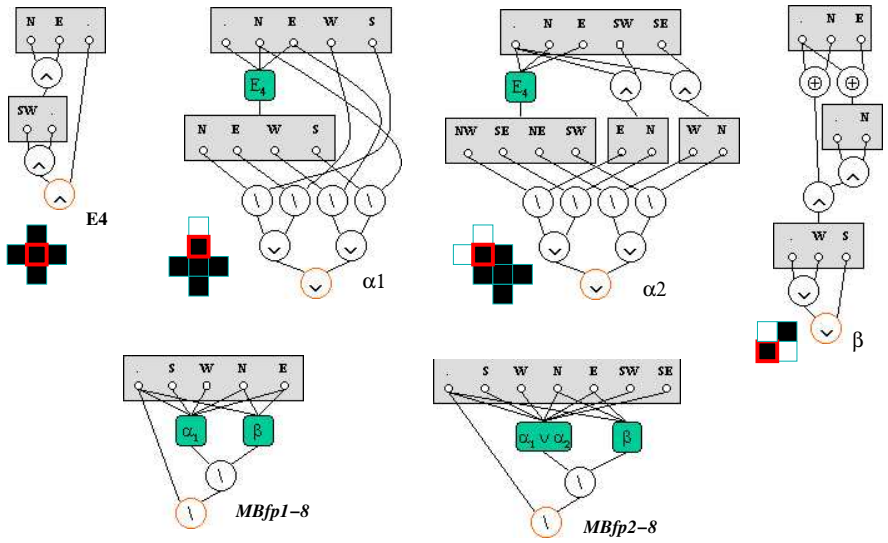


Figure 11: Cellular implementations of the fully parallel 8-connected MB algorithms.

detailed in Figure 12. They use the operators that compute (α_1) and (α_2), (see Figure 11), and a restricted version of (β) (only two rotations of pattern (β) are considered, see [10]). The important difference with MBfp in 8-connectivity is that MBfpx-8 removes pixels matching (α) and not (β), whereas MBfpx-4 removes pixels matching (α) if their deletion does not create a (β) pattern. So the two sub-operators are composed sequentially ; this appears in the shape of

the graphical representations, where (α) and (β) are disposed vertically with a sequential dependence, whereas they were disposed horizontally without dependence in the case of MBfpx-8 (see Figure 11). That decomposition explains the large size of the support (see Figure 9). But in spite of this, the Boolean complexities of MBfp1-4 and MBfp2-4 are slightly inferior than the one of their 8-connected counterparts: 16ρ and 26ρ respectively.

Is this advantage of decomposition meaningful independently on the data

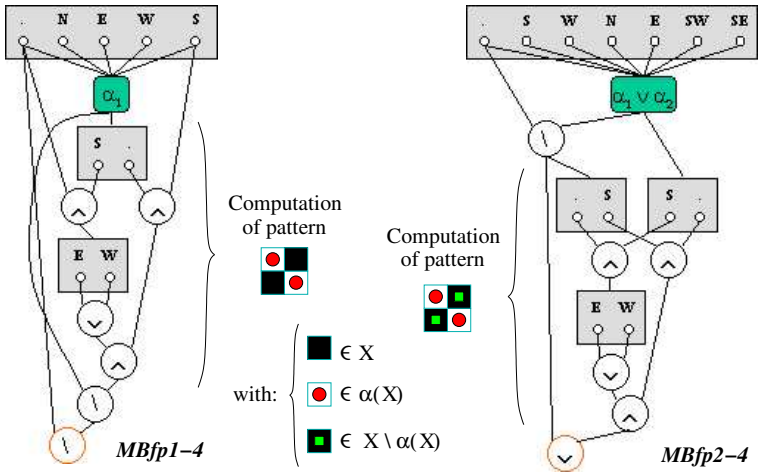


Figure 12: Cellular implementations of the fully parallel 4-connected MB algorithms.

parallelism of the cellular array ? The answer is yes, since a similar decomposition can be performed in other paradigms (sequential, pipe-line, ...). Simple instances of this property can be found in the separable computation of associative Boolean operators (e.g. morphological erosion and dilation), or in the recursive computation of large impulse response filters. We give hereunder a less regular example through the computation of MBfp2-4.

From a geometrical point of view, MBfp2-4 is complex, as it is a *not P-simple* thinning algorithm [2], [10]. From a combinatorial point of view, MBfp2-4 is a very complex operator because of its huge support (38 points). A straightforward implementation with a whole examen of the neighborhood is not realizable, because the Shannon theorem [13] predicts that the corresponding Boolean function has certainly more than $2^{38}/38 > 7 \times 10^9$ binary Boolean operators, and because the size of its *Look-Up-Table* (or LUT, the list of the 38-tuples of bits matching the function) would be approximately of $38 \times 2^{38}/2$ bits, that is more than 600 gigabytes... Nevertheless, the very compact cellular implementation of MBfp2-4 suggest the decomposed implementation shown on Figure 13: first, the erosion is computed (support size: 5), then the (α) function (support size: 16), and finally the restricted (β) function (support size: 8). In the typical case of a

sequential LUT based implementation, the computation time depends roughly linearly of the size of the LUT, as long as the LUT fits into the microprocessor closest memory. It turns out that for such implementation on a processor with 64 kilobytes of internal cache, the decomposed computation of MBfp2-4 shown in Figure 13 is significantly faster than the direct LUT implementation of an average Boolean function of 17 variables.

To conclude this Section, it is important to stress that a large comparative

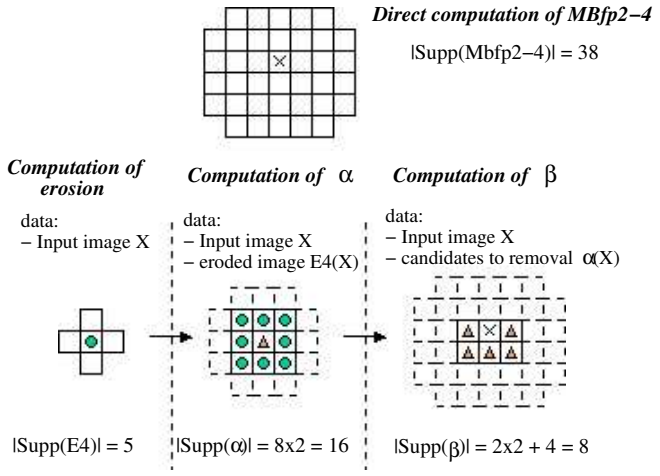


Figure 13: Decomposed implementation of MBfp2-4 inspired by the cellular program.

study has been done on cellular implementations of other thinning algorithms [9], which put the MB algorithms among the most compact.

5 Results and discussion

The compared results of the different algorithms are presented on Figure 14.
The first remark that can be made is that the choice of the topology does not seem to affect fundamentally the geometry of the skeletons, as every -4 skeleton is globally very similar to its -8 counterpart. In fact, this is the case for every well-shaped image [8]. The main difference between the -4 and -8 fully parallel algorithms is that the latter are perfectly symmetric with respect to the 4 cardinal directions (we refer to this property as "isotropy"), whereas, following Latecki et al [8], a small anisotropy is introduced in the former algorithms, corresponding to look for appearing β pattern in two directions only, in order to get a thinner skeleton.
Anyway, no fully parallel MBs lead to one pixel thick skeleton, this is a natural outcome of the fundamental constraints we have chosen. The directional MBs

are anisotropic by construction, as the ordering in the directions of deletion is arbitrary, but they produce a one pixel thick skeleton.
But the more fundamental difference between the directional and fully parallel MBs is geometrical: the MBdir skeletons show more horizontal and vertical lines, whereas the MBfp skeletons show more diagonal lines. This is an important metrical properties that is presented more formally in [10].
Finally, the MB-2 algorithms present less branches than their corresponding MB-1, and in particular, they do not distinguish the squares from the disks, and they generate the same skeleton for the two rotated versions of the square (see Figure 14). This important features of the MB-2 algorithms lead to different properties with respect to noise immunity, rotation invariance and reconstructibility, all of which being detailed in [10].

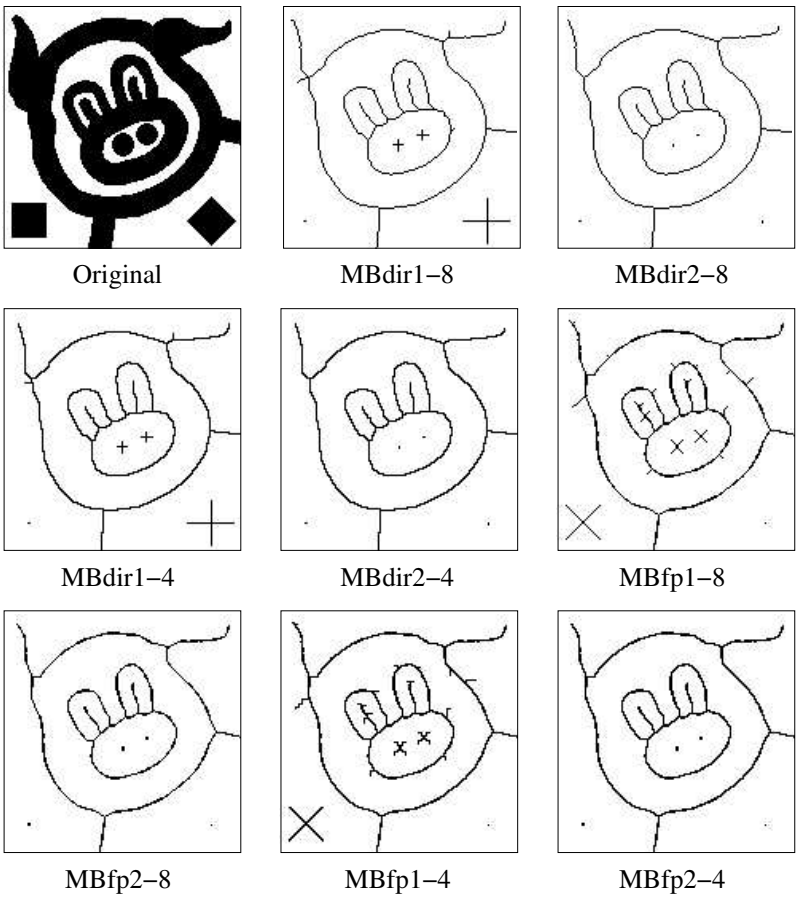


Figure 14: Compared results of the MB thinning algorithms.

References

[1] C. Arcelli. A condition for digital points removal. *Signal Processing*, 1-4:283–285, 1979.

[2] G. Bertrand. On P-simple points. *Comptes Rendus à l'Académie des Sciences*, 321-1:1077–1084, 1995.

[3] H. Blum. A transformation for extracting new descriptors of shape. *Symp. mod. for the perc. of speech and visual form*, 362–380. M.I.T. Press, 1967.

[4] U. Eckhardt and G. Maderlechner. Invariant thinning. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 7-5:1115–1144, 1993.

[5] R.W. Hall. Optimally Small Operator Supports for Fully Parallel Thinning algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 15-8:828–833, 1993.

[6] T.Y. Kong and A. Rosenfeld. Digital topology: introduction and survey. *Computer Vision, Graphics and Image Processing*, 48:357–393, 1989.

[7] L. Lam, S.W. Lee, and C.Y. Suen. Thinning methodologies: a comprehensive survey. *IEEE trans. on patt. analysis and mach. intelligence*, 14:869–885, 1992.

[8] L.J. Latecki, U. Eckhardt, and A. Rosenfeld. Well-composed sets. *Computer Vision and Image Understanding*, 61-1:70–83, 1995.

[9] A. Manzanera, T.M. Bernard, F. Prêteux, and B. Longuet. nD skeletonization: a unified mathematical framework. *Journal of Electronic Imaging*, vol. 11(1), 25–37, 2002.

[10] A. Manzanera and T.M. Bernard. Metrical properties of a collection of 2D parallel thinning algorithms. *Int. Work. on Combinatorial Image Analysis, Palermo, May 2003* Electronic Lecture Notes on Discrete Applied Mathematics, Elsevier Science, 2003

[11] C. Ronse. A topological characterization of thinning. *Theoretical Computer Science*, 43:31–41, 1986.

[12] C. Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Disc. appl. math.*, 21:67–79, 1988.

[13] C.E. Shannon. *The synthesis of two-terminal switching circuits.* Bell Systems Tech. J., 1949.

[14] S. Yokoi, J.I. Toriwaki, and T. Fukumura. Topological properties in digitized binary pictures. *Systems, Computers, Controls*, 4-6:32–39, 1973.