Improved Low Complexity Fully Parallel Thinning Algorithm

Thierry M.Bernard
ENSTA/LEI
32 Bd Victor, F75015 Paris, France
tbernard@ensta.fr

Antoine Manzanera
Aérospatiale
2, rue Béranger, F92320 Chatillon, France
manzaner@etca.fr

Abstract

A fully parallel iterative thinning algorithm called MB2 is presented. It favourably competes with the best known algorithms regarding homotopy, mediality; thickness, rotation invariance and noise immunity, while featuring a speed improvement by a factor two or more owing to a smaller number of operations to perform. MB2 is grounded on a simple physics-based thinning principle that conveys both quality, efficiency and conceptual clarity. It is particularly suited to data parallel execution.

1 - Introduction

Machine vision has often to deal with elongated shapes, of which the length is much larger than the thickness. Exemplary applications are found in character recognition or medical imaging. It is often worth representing these elongated objects by simple lines, i.e. one-pixel-thick connected sets of pixels. This allows a lighter representation easing further pattern analysis. Hence, the notion of skeleton has been introduced, which most usually applies to binary images.

It is often considered that everything has been said about this subject on which hundreds of papers have been published over the last thirty years. Still, skeletonisation algorithms are usually exploited as recipes with an imprecise knowledge of their characteristics and properties. Actually, the latter are often appreciated on a pure experimental basis [1]. This is all the more surprising that there are only a few readily understandable requirements that a skeleton should fulfil. Then one could expect a rigorous characterisation of any skeletonisation algorithm with respect to them. Here are the usual ones:

- Homotopy: the skeleton must preserve the topology of the original image (any hole must remain).
- One-pixel-thickness: the skeleton should be made of curves, i.e. one-pixel-thick objects.
- Mediality: the skeleton should lie in the middle of the shape, with every point of the skeleton the same distance from the two closest borders of the object.
- Rotation invariance: skeletonisation and object rotation should commute. Due to spatial discretisation, this can only be true for rotation angles multiples of $\pi/2$. But this should also be approximately satisfied for arbitrary rotation angles.
- Noise immunity: the skeleton should be fairly insensitive to noise on the image (boundary pixels added or removed).

It is well-known that some of these requirements are contradictory. But this must not be an excuse for lack of clarity and precision in the characterisation of a skeletonisation algorithm. Furthermore, these antagonisms are not that strong. We will actually explain in section 2 how to get away with them.

Beside intrinsic quality and characterisation clarity, computational speed is another important matter. As a low level image processing operator, skeletonisation can take advantage of parallelism. The most common way to do so is to implement skeletonisation as a thinning procedure that iteratively peels objects. Then massive data parallelism [2] proves particularly suitable due to the locality of the thinning procedure: only the values of the closest neighbours are necessary to decide whether a pixel can be removed or not. The number of iterations is determined by the maximal object thickness (the radius of the largest ball contained in the image). Another key advantage of data parallelism is that it allows an intrinsic isotropy of the thinning procedure, which is most important to ensure mediality. This is a point where many sequential algorithms fail.

As will be detailed in section 3.6, we measure the speed of a parallel iterative algorithm (such as thinning) by the total number of Boolean operations performed per pixel before convergence. We call it the Shannon measure due to its obvious relationship to circuit complexity. Its main advantage is its universality.

In this paper, we present a parallel iterative thinning algorithm called MB2 which favourably competes with the best known algorithms regarding the five above criteria, while featuring a speed improvement by a factor two or more. Like its predecessor MB [3] which was even faster but with limited rotation invariance, MB2 is grounded on a unified thinning principle that conveys both efficiency and clarity. Section 2 recalls some preliminaries before exposing our strategy to get a rational trade-off between the different skeletonisation criteria. The MB2 algorithm is presented in section 3 under both pattern matching and procedural forms. Then it is analysed formally and experimentally.

2 - Which skeleton ?

Several properties are desirable for a skeleton, which have been previously listed. The present section recalls definitions and results related to them before discussing their compatibility. This discussion leads to strategic choices that allow to best satisfy all criteria.

2.1 - Basics

Let us first recall that binary images can be considered as subsets of Z^2 (Z is the set of integers). These subsets are always supposed finite (this formally corresponds to images that have pixels with value 1 only in a finite window). A digital topology on Z^2 is defined by adjacency relationships between pixels. For k=4 or 8, the "k-topology" designates the topology for which a pixel is adjacent to its k closest neighbours (for the Euclidean distance). Expressions like k-adjacent or k-neighbour directly follow from this definition. The k-topology induces a distance δ_k: δ_4 is called the square distance on Z^2 and δ_8 the diamond distance, owing to the ball shape they induce (cf. Def.2 in §2.3). δ_4 is also known as the Manhattan distance. Adjacency relationships are readily extended from points (pixels) to subsets of Z^2 (images). $A\subset Z^2$ and $B\subset Z^2$ are said k-adjacent if there exist $a\in A$ and $b\in B$ such that a and b are k-adjacent. $X\subset Z^2$ is said to be a k-CC (k-connected component) if there does not exist any partition of X into two non k-adjacent subsets. $x\in X\subset Z^2$ is said k-interior to X if all its k-neighbours belong to X.

2.2 - Homotopy (connectivity preservation)

Homotopy is a concept which is more naturally defined in the continuous real plane R^2 than in the discrete integer plane Z^2. In particular, in R^2, Jordan's theorem tells that a simple closed arc partitions the plane into two unconnected regions: the interior and the exterior. In Z^2, this important result no longer holds. To overcome this difficulty and most of its harmful consequences, the most appropriate and common solution is to use the k-topology (k=4 or 8) for the foreground of the image (the objects) and the k'-topology for the background, with k'=12-k. *In the rest of this paper, and in particular for the MB2 algorithm, the choice is k=8 and k'=4*. The 8/4 notation will recall this fact.

Two (finite) subsets of Z^2 are homotopic if there exists a one-to-one mapping F between their respective k-CCs and another one B between the respective k'-CCs of the background, such that the tuple (F,B) preserves the adjacency relationships between connected components. Calling C_∞ the only k'-CC that extends to infinity in the background of any finite subset, a supplementary condition is $B(C_\infty)=C_\infty$.

An image and its skeleton must be homotopic. Since skeletonisation only removes points from the foreground, this implies inclusion relationships which actually make homotopy simpler to express: *when removing points, k-CCs of the foreground must neither split nor be erased and k'-CCs of the background must neither merge nor be created*. A point (respectively a set of points) which can be removed while meeting these criteria is called a *simple point* (respectively a *simple set*). A parallel algorithm will preserve connectivity iff (if and only if) the several points it removes simultaneously at each iteration form a simple set. A well-known difficulty is that reunions of simple points are generally not simple sets. Here are some formal characterisations of simple points and sets.

Notations: Logical and arithmetic operators are represented using the C-language syntax. Signs &, |, ^ respectively stand for the logical AND, OR and XOR. Sign + designates standard arithmetic addition. Following a usual set theory notation, sign \ stands for ANDNOT. Given a binary image X, x stands for the binary value of an arbitrary pixel and x_N, x_{NE}, x_E, x_{SE}, x_S, x_{SW}, x_W, x_{NW} stand for the values of its eight closest neighbours (designated using the cardinal points).

Definition 1: Given an image X, the connectivity number (for k=8 and k'=4) of a pixel is $C_{8/4}(x) = (x_N|x_{NE})\backslash x_E+(x_E|x_{SE})\backslash x_S+ (x_S|x_{SW})\backslash x_W + (x_W|x_{NW})\backslash x_N$

Theorem 1 [4]: A point x is simple iff $C_{8/4}(x) = 1$.

But how to guarantee that a thinning algorithm removes only simple sets ? Here is an answer.

Theorem 2 (adapted from [5]): For a parallel thinning algorithm to preserve 8/4 connectivity, it is sufficient to simultaneously respect the three following conditions:

(a) all pixels removed are simple;

(b) an 8-CC contained in a 2×2 square cannot be completely deleted;

(c) any pair of 4-adjacent pixels removed simultaneously would also be removed sequentially.

2.3 - Mediality

Mediality is a geometrical notion which can only be based on the choice of a distance δ. For the sake of rotation invariance, the Euclidean distance δ_e would be the best choice. However the square and diamond distances δ_4 and δ_8 come out much more naturally within the frame of iterative thinning procedures. Here are now a few classical definitions and results that precise the notion of mediality.

Definition 2: Let δ a distance. The δ-ball B(x,r) of centre x and radius r is the set $\{ y\in Z^2 ; \delta(x,y)\leq r \}$.

For $\delta= \delta_4$ (respectively δ_8) we will simply note 4-ball (respectively 8-ball). Now, let $X\subset Z^2$ a binary image.

Definition 3: Let δ a distance. A maximal δ-ball in X is a δ-ball $B\subset X$ with no other δ-ball B' (but itself) such that $B\subset B'\subset X$.

The centres of maximal δ-balls are the points that lie in the middle of the shape from the viewpoint of distance δ. The following results tell how to find them.

Definition 4: Let δ a distance. The distance map associated with X is the function Φ_δ from Z^2 to N (or R^+) such that $\Phi_\delta(x) = \delta(x,X^c) = \min\{ \delta(x,y) ; y\notin X \}$.

For $\delta= \delta_4$ (respectively δ_8), we will simply note Φ_4 (respectively Φ_8).

Theorem 3: The centres of the maximal k-balls (k=4 or 8) in X are the local maxima of Φ_k over their k-neighborhood, i.e. the set $\{ x\in X / \forall y\in B_k(x,1)\ \Phi_k(x) \geq \Phi_k(y) \}$

2.4 - Homotopy versus one-pixel-thickness and mediality

Figure 1a shows a binary image (each square is a pixel) from which no black pixel can be deleted without creating a white 4-CC or merging two. It is an exemplary case which shows that homotopy may imply arbitrary large thickness. Figure 1b presents another case where thinning could produce a one-pixel-thick skeleton but would violate mediality. The skeleton is actually meaningless for such images. Fortunately, these unwanted pathological situations can be characterised and even detected. But this goes beyond the scope of the present paper, in which we will simply not apply our thinning algorithm on such pathological images.

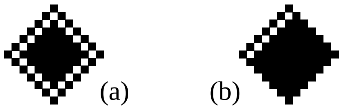


Figure 1: Pathological images.

2.5 - Mediality and rotation invariance versus one-pixel-thickness

Even in the absence of pathological configurations, mediality is in itself an obstacle to one-pixel-thickness. Considering a square-shaped object of size 2×2, each of the 4 pixels are local maxima. Therefore, none should be removed, as is also dictated by rotation invariance. Actually, mediality and rotation invariance only allow two-pixel-thickness. Requiring one-pixel-thickness everywhere would amount to force the thinning algorithm to make arbitrary choices. We think that a more rational approach is to accept the two-pixel-thickness limitation as an inherent characteristics of a skeleton. By the way, this is perfectly consistent with the implicit isotropy that only a data parallel thinning algorithm may feature, like MB2. Passing from two-pixel-thickness to one-pixel-thickness is then only a matter of simple post-processing (simpler than a single thinning iteration).

2.6 - Homotopy, mediality, noise immunity

Homotopy and mediality are fundamentally antagonistic with noise immunity. A single pixel added or removed may completely alter the connectivity and therefore the skeleton. Likewise, a single pixel added at the boundary of an object featuring a simple shape will often be a local maxima far from any other, therefore deeply changing a skeleton subject to the mediality constraint. Still MB2 proves experimentally (for understood reasons) to be more noise-immune than other high-quality algorithms. But for a large part, we believe that noise immunity cannot be the responsibility of a thinning algorithm. It is rather the matter of some application-dependent pre-, post- or concurrent processing.

2.7 - Putting it all together

The goal of section 2 was to determine the best possible trade-off between the different desirable properties of a skeleton. A main concern was genericity, which goes with conceptual clarity and hopefully efficiency. It turns out that homotopy, mediality and 2-or-1-pixel-thickness are the three prime compatible constraints once some peculiar and meaningless configurations are excluded. Then mediality is governed by the explicit or implicit choice of a distance, which also affects rotation invariance and, to a lesser extent, noise immunity. This strategy proves fully consistent with a data parallel framework, from which both isotropy and speed are expected.

3 - The MB2 thinning algorithm

3.1 - Principles

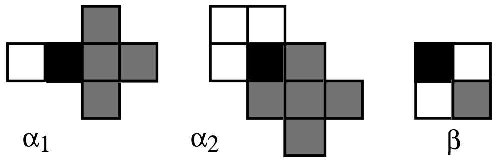


Figure 2: The MB2 patterns (which come with all their π/2 rotated versions).

The MB2 algorithm is a parallel iterative algorithm that deletes points corresponding to certain neighborhood conditions. The latter are based on patterns displayed on figure 2. In these patterns, black and grey pixels belong to the foreground and white pixels to the background. Given an image, it is said that pixel x matches some pattern if, putting that pattern on the image with the black pixel on x, there is an exact coincidence of the image with the whole pattern. It is also considered that each pattern may be rotated by any multiple of π/2.

> **Definition 5**: The *MB2 iteration* consists in removing simultaneously all pixels that match α_1 or α_2, but not β. The *MB2 algorithm* repeats the MB2 iteration until stability is reached.

Considering the α_i patterns (i=1 or 2), it appears that a point is removed by the MB2 iteration if it lies exactly between the background and a 4-interior point in the foreground to which it is n-adjacent, with n=4 for α_1 and n=8 for α_2. Actually, the α_i patterns convey the physics of the thinning action and this is the strong conceptual originality of the algorithm. We believe this is responsible for the remarkable behavior of the algorithm as the next sections will show.

Another unusual feature of the MB2 patterns is their size. As α_1 and α_2 rotate around their origin (the black point), the whole centered 5×5 neighbourhood, but its four corners, is described. This contrasts with the other algorithms which attempt to minimise the size of this working neighbourhood [6]. This is certainly crucial for computations performed using look-up tables, but meaningless otherwise. For us, the point was to choose the patterns that minimise computational complexity (cf. section 3.6).

3.2 - MB2 procedure

Let B_4=B_4(o,1) be the unity 4-ball (o is the origin of Z^2), as shown on figure 3. Note that B_4 (translated) appears in the foreground part of both α_1 and α_2. Let us also recall a basic notion of mathematical morphology:

the *erosion* of an image X by $B \subset Z^2$, denoted $X \Theta B$, is the set of all points $x \in Z^2$ such that the translated set of B by x (acting as a vector) is completely included in X.

Table 1 presents an optimised procedure that performs the MB2 iteration in a data parallel framework: any computation with the local binary variables a1, a2, b, etc., is performed on the whole corresponding images A1, A2, B, etc. The notations used are those presented in section 2.2. Variable names have been chosen to be as explicit as possible, as explained in the table caption.

$e4 = x_N \;\&\; x_E$;
$e4 = e4 \;\&\; e4_{SW} \;\&\; x$;
$a1 = (e4_W \backslash x_E) \mid (e4_E \backslash x_W) \mid (e4_N \backslash x_S) \mid (e4_S \backslash x_N)$;
$d1 = x \mid x_{SW}$;
$d2 = x \mid x_{SE}$;
$a2 = (e4_{SE} \backslash d1_N) \mid (e4_{NW} \backslash d1_E)$
$\quad \mid (e4_{SW} \backslash d2_N) \mid (e4_{NE} \backslash d2_W)$;
$b = x \wedge x_N$;
$b = b \;\&\; b_E \;\&\; (x \wedge x_E)$;
$b = b \mid b_S$;
$b = b \mid b_W$;
$x = x \backslash ((a1 \mid a2) \backslash b)$;

Table 1: A data parallel procedure that performs the MB2 iteration. Variable names correspond to the following outcomes of the procedure: E4 = $X \Theta B_4$ while $A1 \cap X$, $A2 \cap X$ and $B \cap X$ respectively contain the points of X that match α_1, α_2 and β. Those points are removed from X in the last line.

The compactness of the MB2 procedure results from different features of the MB2 patterns: (a) their small number, (b) their simplicity, (c) their regularity, which allows the use of spatial decomposition techniques to compute them. Also, some redundancy between the patterns has been removed. In particular, the upper-left white point in pattern $\alpha 2$ is useless since pattern β would reject a black point there. Finally, the number of Boolean operations required by an MB2 iteration is as low as 28. Before comparing with the cost of other algorithms, quality is first to be examined, as the next sections do. As far as the memory requirement is concerned, the MB2 procedure uses 7 binary variables per pixel for readability purposes; but some of them could have been reused, thus lowering their number to 5 including image X.

3.3 - Homotopy

MB2 removes points in parallel all around the image. It is therefore called a fully parallel (FP) algorithm. This contrasts with a large family of thinning algorithms which remove points in several (4 or 2) directional (D) sub-iterations [7, 8] (only pixels located on a particular side of the objects are removed at a time). For the latter algorithms, satisfying homotopy is a simpler issue [9]. On the other hand, MB2 needs Theorem 2 to prove that it preserves homotopy.

Condition (a) of Theorem 2 asks that every pixel removed is simple. Considering Theorem 1 and Definition 1, it can be shown that a point which is not simple necessarily matches either of the three patterns shown on figure 3 (or their rotated versions). But MB2 forbids β while $\alpha 1$ and $\alpha 2$ prevent points from matching B4 or γ.

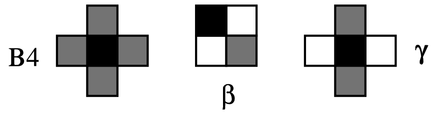


Figure 3: Non simple patterns

Condition (b) is met since the foreground parts of $\alpha 1$ and $\alpha 2$ extend beyond a 2×2 square.

Condition (c) requires three cases to be examined:

- If the two 4-adjacent pixels simultaneously removed both match $\alpha 1$ (respectively $\alpha 2$), then the only possible situation is that of figure 4a (respectively 4b) with all its rotations. It appears that if one is removed first, the other remains removable.
- If one point matches $\alpha 2$ and the other $\alpha 1$, then the only possible situation is that of figure 4c. In that case, if the point that matches $\alpha 2$ is removed first, the other remains removable.

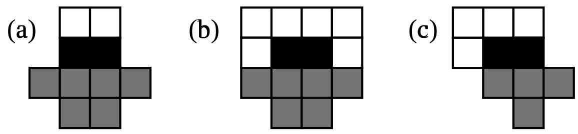


Figure 4: Simultaneous removal of a pair of 4-adjacent pixels.

This ends the proof that *MB2 is homotopic*. From the details of the proof, it turns out that much of the homotopy is inherently conveyed by patterns α_i. For us, this is yet another virtue of their physical meaning.

3.4 - Rotation invariance and mediality

The MB2 algorithm is 4-isotropic: the thinning action is exactly the same in the four cardinal directions (N, E, S, W). This results from the perfectly symmetrical expressions used in the MB2 procedure and from the full parallellism of the algorithm (no anisotropy induced through the order in which pixels are examined). Then exact rotation invariance is guaranteed for angles multiples of $\pi/2$. This 4-isotropy is also the reason why the MB2 skeleton is 2-pixel-thick in some places.

Actually, these nice properties were already featured by the predecessor of MB2, namely MB [3] of which sample outputs are presented on figure 6b and 7b. MB is only based on patterns $\alpha 1$ and β. As illustrated on figure 6, MB preserves all centres of maximal 4-balls. Thus the skeleton produced by MB is medial from the viewpoint of distance $\delta 4$. However, $\delta 4$ is not the ideal distance. Neither is $\delta 8$, on which the JC algorithm [7] (cf. figure 7a) is based. In order to improve rotation invariance, it is desirable to be less biased by either of these two distances. The additional pattern $\alpha 2$ used concurrently with $\alpha 1$ by MB2 was introduced for this purpose. As a first outcome, MB2 — unlike MB — reduces 4-balls as well as 8-balls to their centres. Actually, all shapes shown on figure 5 (where each black square is a pixel) are reduced to a single point (superimposed in white). We call them fuzzy balls. Here are some more precise definitions and results.

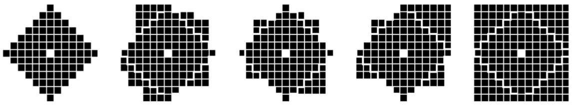


Figure 5: Fuzzy balls.

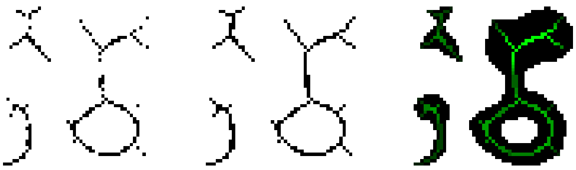
Notation: Given two points a and b of Z^2, let BB(a,b) be their bounding box, i.e. the smallest rectangular area (with vertical and horizontal edges) including a and b.

Definition 6: X is a fuzzy ball iff $(\exists x \in Z^2)(\exists r \in N)$ $B_4(x,r) \subset X \subset B_8(x,r)$ and $(\forall y \in X)$ $BB(x,y) \subset X$. If so, point x is called its centre, and r its radius.

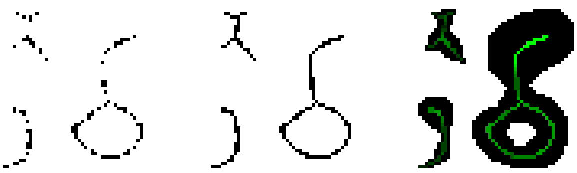
Then the notion of maximal ball (cf. definition 3) may be straightforwardly extended to fuzzy balls with direct application to the characterization of the MB2 algorithm.

Theorem 4: The centres of maximal fuzzy balls in X is the set $\{ x \in X \,/\, \forall y \in B_8(x,1)\ \Phi_4(x) \geq \Phi_4(y) \}$.

Theorem 5 : MB2 reduces any fuzzy ball to its centre.



(a) centres of maximal 4-balls (b) MB skeleton (c) exact reconstruction



(d) centres of maximal fuzzy balls (e) MB2 skeleton (f) approximate reconstruction

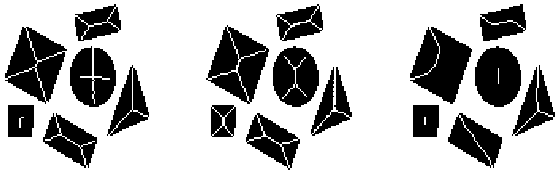
Figure 6: MB2 versus MB.

Theorem 4 (note the difference with theorem 3) and theorem 5 explain why MB2 does not preserve all centres of maximal 4-balls, but only those of maximal fuzzy balls (cf. figure 6abde). However, this results into a better isotropy, though the influence of δ_4 is still predominant (the MB2 skeleton is included in the MB skeleton).

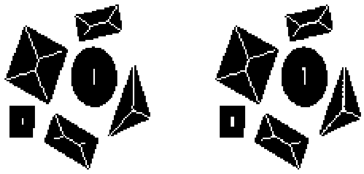
Figure 7 allows to appreciate the different characteristics of the MB2 algorithm versus some renowned thinning algorithms proposed within the last ten years: "JC" stands for the Jang & Chin algorithm [7], "St" for the Stewart algorithm [10] and "AFP3" is the best of 3 algorithms jointly presented in [11].

3.5 - Noise immunity and reconstructibility

Thanks to theorem 5, MB2 features significant noise immunity. Figure 8 shows the behavior of several algorithms on an elementary shape which is noise corrupted (right) or not (left). MB2 proves to feature levels of immunity comparable to that of AFP3. The main difference between them lies in the thickness, which is arbitrarily forced to 1 in the case of AFP3.



(a) JC (b) MB (c) St



(d) AFP3 (e) MB2

Figure 7: Comparison of 5 thinning algorithms on simple rotated shapes.

However noise immunity implies information loss. Now, for some applications, the skeleton is expected to allow the reconstruction of the shape. For this purpose, every point in the skeleton is weighted with the iteration number at which it has been reached by the thinning process. For a point which is a local maximum, this gives the radius of the maximal ball it is the centre of. On figure 6cf, the radius of a skeleton point is represented by its grey level. In the case of MB2, it is not possible however to know if the ball to be regenerated from a skeleton point is for the δ_4 or δ_8 distance. A solution is to use octagonal balls as intermediate shapes between squares and diamonds. This provides an approximate reconstruction shown on figure 6f whereas the original shape appears on figure 6c. This allows to better understand the very nature of the noise immunity featured by MB2 or AFP3.

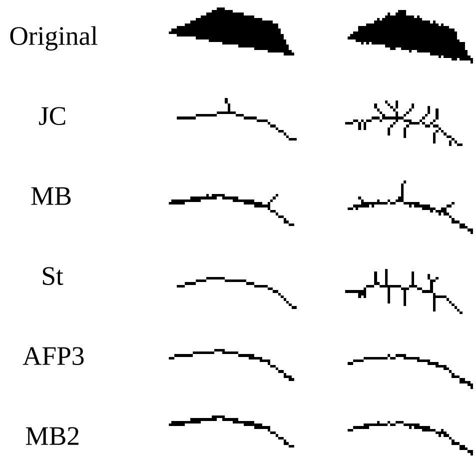


Figure 8: Comparative noise immunity.

3.6 - Speed

As mentioned in section 3.2, an MB2 iteration needs 28 Boolean operations with two binary operands. In practice, we have just counted the number of symbols &, |, ^ and \ that appear in table 1. This measure of computational complexity represents the number of two-input logic components that are needed to perform the computation. It has actually been proposed by Shannon [12] in the early times of logic synthesis to define the complexity of Boolean functions. It is still the reference in the field of Boolean complexity. The Shannon complexity of a data parallel procedure represents its minimal execution time on a basic cellular automata machine, such as the one developed in cooperation with the authors, under the form of a Programmable Artificial Retina [13]. However, the Shannon measure makes sense for any software or hardware implementation as it is related to Area-Time complexity. It also represents the minimal amount of energy required to perform the computation of a digital function, as every operation between two bits of information physically consumes some energy quantum, whatever the computation medium.

Table 2 compares the complexity of several parallel algorithms. However, some of them are directional (D) and it takes them 2 to 4 passes to do the same work as fully parallel (FP) algorithms. This is often a source of confusion when comparing algorithm speed. To avoid that, table 2 gives the Shannon complexity per pixel of the overall skeletonisation process as a function of R, i.e. the radius of the biggest ball contained in the image. The MB and MB2 algorithms feature the best computational efficiency. In fact they are the first fully parallel algorithms to beat directional ones.

Algorithm	Reference	Type	Neighb. Size	Shannon complexity
JC	[7]	D	7	32×R
—	[8]	D	7	40×R
St	[10]	FP	19	60×R
—	[14]	FP	11	60×R
AFP3	[11]	FP	11	80×R
MB	[3]	FP	13	18×R
MB2	—	FP	21	28×R

Table 2: Computational complexity for several parallel thinning algorithms. Some complexities are only estimations as the quoted papers did not always provide a minimal Boolean expression of their algorithm.

4 - Conclusion

We have described a fully parallel thinning algorithm which features the quality of AFP3 [11], one of the best algorithms, at the speed of JC [7], the fastest one we were aware of before MB [3]. Furthermore, MB2 is based on a clear thinning principle that follows a strategy defined at the beginning of the paper. Thanks to the algorithm genericity, recent work by the authors has allowed to extend it to 3D [15] and even to nD [16]. Ongoing work should lead to a formal characterisation of the algorithm.

5 - References

[1] L. Lam and C.Y. Chin, "An Evaluation of Parallel Thinning Algorithms for Character Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp.914-919, 1995.

[2] C. Arcelli, L. Cordella and S. Levialdi, "Parallel thinning of binary pictures," *Electronic Letters*, vol. 11, no. 7, pp. 148-149, 1975.

[3] A. Manzanera, T.M. Bernard, F. Prêteux and B. Longuet, "Ultra-fast skeleton based on isotropic fully parallel algorithm," Proc. of Discrete Geometry for Computer Imagery, 1999.

[4] S. Yokoi, J.I. Toriwaki and T. Fukumura, "Topological properties in digitised binary pictures," *Systems, Computers, Controls*, Vol. 4, no. 6, pp. 236-239, 1973.

[5] C. Ronse, "Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images," *Discrete Applied Mathematics*, Vol. 21, pp. 67-79, 1988.

[6] R.W. Hall, "Optimally Small Operator Supports for Fully Parallel Thinning algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no.8, pp.828-833, 1993.

[7] B.-K. Jang and R.T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no.6, pp. 514-551, 1990.

[8] R. Cardoner and F. Thomas, "Residuals + Directional Gaps = Skeletons," *Pattern Recognition Letters*, Vol. 18, pp. 343-353, 1997.

[9] T.Y. Kong and A. Rosenfeld, "Digital Topology : Introduction and Survey," Computer Vision, Graphics and Image Processing, Vol. 48, pp.357-393, 1989.

[10] A. Stewart, "A one-pass thinning algorithm with interference guards," *Pattern Recognition Letters*, Vol. 15, pp. 825-832, 1994.

[11] Z. Guo and R.W. Hall, "Fast Fully Parallel Thinning Algorithms," Computer Vision, Graphics and Image Processing, Vol. 55, no.3, pp.317-328, 1992.

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

[13] F. Paillet, D. Mercier and T.M. Bernard, "Making the most of 15k lambda2 silicon area for a digital retina PE," Proc. SPIE, Vol. 3410, Advanced Focal Plane Arrays and Electronic Cameras, pp. 158-167, 1998.

[14] R.-Y. Wu and W.-H. Tsai, "A new One-Pass Parallel Thinning Algorithm for binary images," *Pattern Recognition Letters*, Vol. 13, pp. 715-723, 1992.

[15] A. Manzanera, T.M. Bernard, F. Prêteux and B. Longuet, "Medial faces from a concise 3D thinning algorithm", Proc. ICCV, 1999.

[16] A. Manzanera, T.M. Bernard, F. Prêteux and B. Longuet, "A unified mathematical framework for a compact and fully parallel n-D skeletonisation procedure", Proc. SPIE, Vol. 3811, Vision Geometry VIII, 1999.