Ultra-fast skeleton based on an isotropic fully parallel algorithm

Antoine Manzanera[134], Thierry M.Bernard[2], Françoise Prêteux[4], and Bernard Longuet[1]

[1] Aerospatiale E/SCS/V, 2 Rue Béranger, 92320 Chatillon Cedex, France
`Bernard.Longuet@missiles.aerospatiale.fr`
[2] ENSTA/LEI, 32 Bd Victor, 75015 Paris, France
`Thierry.Bernard@ensta.fr`
[3] CTA/GIP, 16bis Avenue Prieur de la Côte D'Or, 94114 Arcueil Cedex, France
`Antoine.Manzanera@etca.fr`
[4] INT/Signal et Image, 9 Rue Charles Fourier, 91011 Evry Cedex, France
`Francoise.Preteux@sim.int-evry.fr`

Abstract. In this paper we introduce a new thinning algorithm, called MB, which is optimized with respect to the total number of elementary Boolean operators needed to perform it. We first emphasize the sound foundations of the algorithm, which is built by expressing into the Boolean language the three following constraints: (1) homotopy, (2) median axis and (3) isotropy. The MB algorithm benefits from both novel algorithmic ideas and systematic logic minimization. By hunting down any redundancy in the expressions of topological/geometrical features, we achieve a procedure that is: firstly, dramatically low-cost, as it is completely computed in 18 Boolean binary operators per iteration, and secondly, fully parallel, or one-single-pass, which guarantees that the number of iterations equals half the biggest object thickness.
Keywords: Thinning - Skeleton - Fully Parallel Algorithm - Isotropy - Median Axis - Discrete Topology - Logic Minimization - Boolean Complexity.

1 Introduction

The process that consists in peeling iteratively a black object on a white background while preserving its topological properties has given rise to intensive international research activities for the last forty years with hundreds of published papers. The importance of such an investigation is due to the interest in digital skeletons (or image thinning), as a fundamental pre-processing step in pattern recognition and image analysis. Its duration is due to the fact that no definitely satisfying solution has been found.

Properties required for a skeleton can vary slightly according to the applications under study. Let us mention the most usual ones, ranked by order of importance:

– *Homotopy:* the skeleton must preserve the topology of the original image.

- *Median axis:* the skeleton must contain the centres of maximal balls.
- *Isotropy:* skeletonization and object rotation must commute, at least for rotation angles multiples of $\pi/2$.
- *One-pixel Thickness:* speaks for itself.
- *Noise Immunity:* the skeleton must be fairly unsensitive to noise on the image (boundary pixels added or removed).

It is well-known, however, that theses requirements cannot be satisfied simultaneously, as some of them are contradictory. In particular, Noise Immunity is incompatible with keeping the whole median axis in the skeleton. Likewise, One-pixel Thickness is incompatible with Isotropy, but can be obtained by a low-cost post-processing step on two-pixel-thick lines. So we decide to restrict ourselves to the three former constraints, as they seem the most fundamental ones.

Thinning algorithms which are both iterative and local may be either sequential or parallel. In this paper, we consider only the parallel framework. We propose a dramatically fast thinning parallel algorithm called *MB* (after the names of its two creators). Indeed MB proves to be faster than all other algorithms we know of, in a fundamental sense: for us, the *fastest* possible algorithm is the one that minimizes the overall computational effort required to achieve the desired image transformation. More precisely, whereas the cost of parallel algorithms is often defined as the number of iterations before convergence [5], [4], we rather define it as the number of elementary operations performed per pixel during the complete execution of the algorithm.

The paper is organized as follows. In the next section, we simply present the MB algorithm as a recipe, and show some experimental comparative results. Section 3 introduces the complexity measure we use to quantify the computational cost of the MB algorithm and its competitors. Section 4 details the different issues of parallel thinning (according to the previously chosen criteria) and explains how the MB algorithm deals with them.

2 The MB algorithm

Let us first introduce the formalism and related notations used in this paper. Here, $\mathbb{Z}^2$ is the *discrete plane*, represented by a square mesh. A *binary image* X is a subset of $\mathbb{Z}^2$. Considering X, a point p of $\mathbb{Z}^2$ or *pixel* is characterized by its *binary value* $x = \mathbf{1}_X(p)$, where $\mathbf{1}_X$ is the characteristic function of the set X. The convention used to designate the value of an 8-neighbor is shown on Figure 1. The MB procedure appears in Table 1, including some hint comments to get the spirit of the algorithm at once (before reading Section 4). Computations therein only involve *Boolean operators* (i.e. Boolean functions of two binary variables). Four of them are actually used, with the following notations: symbol $\wedge$ stands for the *conjunction* (AND), symbol $\vee$ for the *disjunction* (OR), symbol $\setminus$ for the *exception* (AND NOT) and symbol $\oplus$ for the *inequality* (XOR). The procedure uses intermediate pixel binary variables a, b and c that actually correspond to

X_{NW}	X_N	X_{NE}
X_W	X	X_E
X_{SW}	X_S	X_{SE}

Fig. 1. Pixels values in the 8-neighborhood.

Table 1. The MB procedure.

The following sequence is repeated for all pixels within a parallel framework, until stability:

(1) $a = x_N \wedge x_E$
(2) $a = a \wedge a_{SW} \wedge x$
// A = eroded set of X
(3) $b = (a_S \setminus x_N) \vee (a_N \setminus x_S) \vee (a_E \setminus x_W) \vee (a_W \setminus x_E)$
// B = {pixels between a point in A and a point in $\bar{X}$ }
(4) $c = x \oplus x_E$
(5) $c = c \wedge c_N \wedge (x \oplus x_N)$
(6) $c = c \vee c_W$
(7) $c = c \vee c_S$
// C = {pixels that belong to pattern (β) of section 4.3 }
(8) $x = x \setminus (b \setminus c)$
// removed pixels are those that belong to B but not to C

binary images A, B and C. Figure 2 shows some results of the MB algorithm. On these various examples, the three criteria we have selected in the introduction are met. The two-pixel-thick segments that appear in the produced skeleton are a natural outcome of the isotropy of the MB procedure. Anyway, there exist arbitrary thick objects that must not be thinned, such as the one shown in the top right corner of the image, which is indeed preserved by the MB algorithm. Table 2 outlines the main characteristics of the MB algorithm, to be detailed in the next sections.

Table 2. Main characteristics of the *MB algorithm*

- *Fully Parallel* thinning algorithm.
- Number of iterations before convergence equals half the maximal thickness.
- 18 elementary Boolean operations per iteration.

3 Complexity measure

Before going further into the MB algorithm, let us consider its complexity. Counting the number of symbols $\wedge$, $\vee$, $\oplus$ and $\setminus$ in the MB procedure (cf. Table 1), leads to assign a *computational cost of 18* to each iteration. This measure represents the number of two-input logic components that are needed for the computation, no matter if it is at the software or hardware level. This measure has actually been used since the early times of logic synthesis to define the complexity of Boolean functions, as recalled hereunder [17]:

Definition 1. *Let $f : \{0,1\}^n \longrightarrow \{0,1\}$ be a Boolean function of n variables. Let B_2 be the set of the Boolean functions of two variables (*and*,* or*,* xor *...). A circuit γ over B_2 is a connected, acyclic oriented graph with nodes either having in-degree 2 and labeled by a function of B_2 (these nodes are called* gates*), or having in-degree 0 and labeled by variables (these nodes are called* leaves*). Only one node has out-degree 0 and is called* output*. A node computes a value by applying its labeling function to the values computed by its preceeding nodes. Thus a circuit γ with n distinct variables labeling its leaves computes a Boolean function f of n variables in its output. γ is called* a *syntax of f, whereas f is called* the *semantic of γ.*

Definition 2. *The Shannon complexity of a Boolean function is equal to the number of gates of its most compact syntax.*

Here, due to the parallel expression of the MB algorithm, the Shannon measure represents the minimal execution time on a basic Cellular Automata machine, just as the one developed in our team, 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 given Boolean function, as every operation between two bits of information physically consumes some energy quantum, whatever the computation medium.

4 Computing digital skeletons

We now move to the algorithmic principles. Here, the MB algorithm is presented following a constructive approach. Using fully isotropic procedures, it is shown first how to keep the medial axis in the skeleton, then how to preserve homotopy, even when pixels are removed in parallel. For each of these issues, some basic definitions or results are recalled before a precise explanation of the MB algorithm operation is provided.

4.1 Distance map - Local maxima

In the discrete plane represented by the square mesh, two topologies can be defined, depending on the neighborhood relation: either the 4-connectivity, where a pixel has one neighbor in each cardinal direction (N, E, W and S), or the 8-connectivity, where a pixel has 4 more neighbors (NE, NW, SW, SE). This induces two basic distances in $\mathbb{Z}^2$: δ_4, or diamond distance, and δ_8, or square distance.

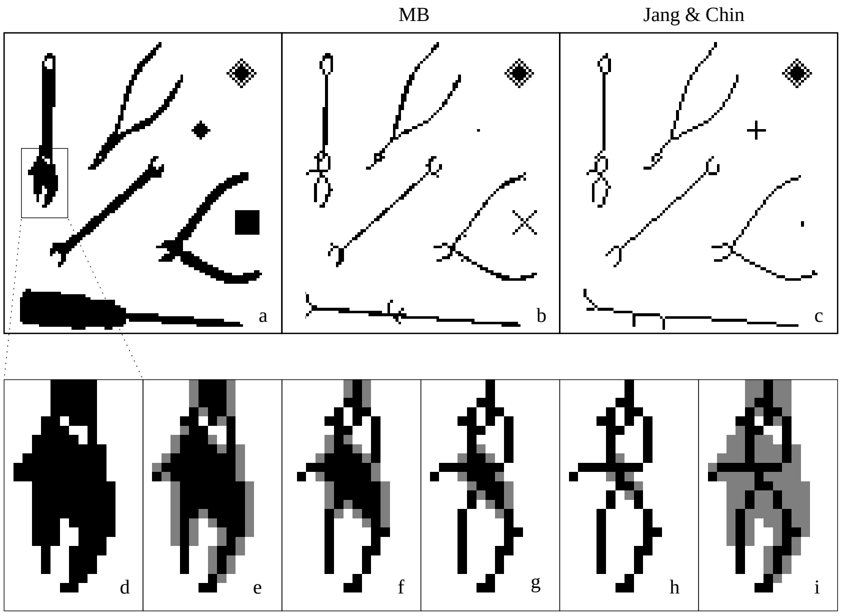

Fig. 2. Experimental results. Image (a) is the original one, whereas image (b) is the result of the convergence of the MB algorithm, reached after 6 iterations. Images (d) to (h) show a detail for the 4 first iterations (the grey pixels are the removed ones) and image (i) represents the overall process. For comparison purposes, image (c) shows the result obtained by the Jang and Chin algorithm [8] on the same image.

Definition 3. *Let δ be a distance in $\mathbb{Z}^2$. Let $\mathcal{B}_n^x = \{y \in \mathbb{Z}^2; \delta(x,y) \leq n\}$ be the ball of centre $x \in \mathbb{Z}^2$ and radius $n \in \mathbb{N}$. Let $X \subset \mathbb{Z}^2$. The collection of the centres of maximal balls is the set:*

$$\mathcal{S}(X) = \bigcup_{n \in \mathbb{N}} \{x \in \mathbb{Z}^2; \mathcal{B}_n^x \subset X \textit{ and } \forall y \in \mathbb{Z}^2, \forall m \in \mathbb{N}, \mathcal{B}_m^y \subset X \Rightarrow \mathcal{B}_n^x \not\subset \mathcal{B}_m^y\}$$

Definition 4. *Let $X \subset \mathbb{Z}^2$. The* distance map *associated with X is the function:*

$$\begin{aligned} \Phi : \mathbb{Z}^2 &\longrightarrow \mathbb{N} \\ x &\longmapsto \Phi(x) = \delta(x, X^c) \end{aligned}$$

Property 1. $\mathcal{S}(X) = \bigcup\{x \in X; \forall y \in \mathcal{B}_1^x, \Phi(y) \leq \Phi(x)\}$. In other words, the collection of the centres of maximal balls corresponds to the set of local maxima of the distance map.

This set is related to the notion of *median axis*. Historically, it has been the first (and last !) sheer definition of the skeleton [2]. However, it is not a satisfying shape descriptor, since it does not preserve topology.
Anyway, we must guarantee that the set we are building by iterated passes of thinning *contains* the median axis. A natural idea is *to compute explicitely* the set of local maxima in order to impose its presence in the result [20], [3]. The simplest iterative parallel procedure that provides the local maxima for distance δ_4 is shown on Table 3. Lines 1 and 2 in Table 3 compute the *eroded set* of

Table 3. Local maxima procedure.

(1) $a = x_N \wedge x_E$
(2) $a = a \wedge a_{SW} \wedge x$
(3) $b = a_N \vee a_E$
(4) $b = b \vee b_{SW} \vee a$
(5) $x = x \setminus b$

image X, i.e. the set A of pixels of X whose 4-neighbors are all in X. Lines 3 and 4 compute the *dilated set* of A, i.e. the set B of pixels of $\mathbb{Z}^2$ for which *at least one* 4-neighbor is in A. Finally the set-difference of X and B is computed in line 5. This procedure removes every pixel x whose neighborhood matches the pattern shown on Figure 3.1, or one of its $\pi/2$ rotated configuration. The hachured pixels have value 1, and at least one of the three pixels denoted by y has value 0. The computational cost is 7. What we compute in the MB algorithm

Fig. 3. (1) Non local maximum pattern and (2) Pattern (α).

is slightly different, and actually a bit more restrictive. Refering to Table 1, lines 1-3, together with line 8 (in which we temporarily forget about c), removes every pixel x whose neighborhood matches a pattern called (α), shown on Figure 3.2, or one of its $\pi/2$ rotated configuration. The hachured pixels have value 1, the white pixel has value 0. Thus it is obvious that a super-set of the local maxima will be produced by the MB algorithm: this makes the first step of a proof. Matching pattern (α) increases the cost up to 11 operations (instead of 7), but as we will see, it is worth the extra.

4.2 Homotopy

The notion of homotopy in the real plane $\mathbb{R}^2$ is originally defined between *arcs*, i.e. continuous functions from a real interval $[a, b]$ to $\mathbb{R}^2$. Two arcs are *homotopic* if there exists a continuous function mapping one to the other. This relation being an equivalence relation, it is said that 2 open subsets of $\mathbb{R}^2$ are *topologically equivalent* if their sets of equivalence classes are isomorphic. In other words, two subsets of the plane are topologically equivalent (we may say homotopic) if there exists a one-to-one mapping between their connected components, *and another one* between the connected components of their complementary sets (the holes), both preserving the relation *"is contained inside..."*. Indeed, a connected component (or a hole) Y is said to be contained inside the connected component X if there exists a closed arc Γ in X that *surrounds* Y. This means that Y is completely contained *in the interior* of Γ. The notion of homotopy is indeed sharply based on the Jordan theorem : *a simple closed arc (or Jordan arc) divides the plane in two distinct regions : the interior and the exterior.* As mentioned earlier, two kinds of topology can be used in the discrete plane. In both cases, however, the Jordan theorem is no longer valid, and so the notion of *hole* is not well-defined. To cope with this problem, two distinct topologies must be used for the image and for its complementary. So homotopy will make sense if one always considers the 8-connectivity for the image and the 4-connectivity for its complementary (or vice-versa). We may now define the notion of *simple point.*

Definition 5. Simple point

Let $X \subset \mathbb{Z}^2$ be a binary image. Let $a \in X$. Point a is simple *iff $X \backslash \{a\}$ is homotopic to X.*

A well-known result is that the simplicity of a point can be determined by looking only at its 3x3 neighborhood. The forerunner in the search of a formula over the 3x3 neighborhood to characterize simplicity is Rutowitz [16]. The more elaborated notion of *connectivity number* was first defined by Yokoi [22], though it had been used earlier by Hilditch [7]. The 8- and 4-connectivity numbers are respectively defined as follows:

Definition 6. Connectivity numbers

$$\mathcal{N}c_8 = \sum_{i=0}^{3} (x_{2i+1} \vee x_{2i+2}) \setminus x_{2i} \ . \qquad \mathcal{N}c_4 = \sum_{i=0}^{3} x_{2i} \setminus (x_{2i+1} \wedge x_{2i+2}) \ .$$

where the x_j represent the 8-neighbors of x in the following way: $x_0 = x_E$, and so on in a counter-clockwise sense until $x_7 = x_{SE}$. Sign + within indices stands for the addition modulo 8. The N-connectivity[1] number of a pixel x within its 8-neighborhood $V(x)$ is exactly equal to the number of N-connected components of $V(x) \backslash \{x\}$, except for N-interior points (points of the image for which all their N-neighbors are also in the image). The N-connectivity number provides an easy way to characterize the N-simplicity. *A point is N-simple if and only if its N-connectivity number equals one.*

[1] N = 4 or 8.

We can compute the 8-simplicity of pixels by using the procedure shown on Table 4, to be read column after column. It computes exactly the formula of Definition 6 and the related cost is 15 operations. The 8-simplicity can also be

Table 4. Hilditch procedure.

(1) $a = x \vee x_E$	(6) $a = x \vee x_N$	(10) $b = a_{SE} \setminus x_S$	
(2) $d = a_N \setminus x_E$	(7) $b = a_W \setminus x_N$	(11) $c = c \vee (d \wedge b)$	// $c = 1 \Longleftrightarrow \mathcal{N}c_8 > 1$
(3) $a = a_{SW} \setminus x_W$	(8) $c = c \vee (d \wedge b)$	(12) $d = d \vee b$	// $d = 1 \Longleftrightarrow \mathcal{N}c_8 \geq 1$
(4) $c = d \wedge a$	(9) $d = d \vee b$		
(5) $d = d \vee a$		(13) $a = d \setminus c$	// $a = 1 \Longleftrightarrow \mathcal{N}c_8 = 1$

expressed under the following pattern-matching form. A pixel x is neither an *interior* nor a *simple* point if its neighborhood matches one of the two patterns represented in Figure 4, along with their $\pi/2$ rotated configurations. The con-

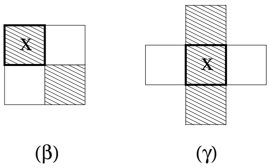

Fig. 4. Gap patterns.

verse is also true, except if $V(x)$ does not contain more than two points. By the way, this is the property that was used by Arcelli [1] for the computation of his skeleton. Here it reveals some redundancy between the computations of local maxima and simple points. Indeed, if a pixel x is considered as removable by the first part of the MB algorithm (because it matches pattern (α)), then its neighborhood may match pattern (β), but never pattern (γ). This fact makes it possible to save operations, by computing only the pattern matching of (β), as performed by the end of the MB procedure (lines 4-7).

So, knowing which pixels match pattern (α) (or a $\pi/2$ rotated configuration) reduces to 6 (instead of 15 !) the number of operations required for the computation of simplicity. This a posteriori justifies the enigmatic choice of (α) in the previous section. Anyway, we have now shown that the points removed by the MB algorithm are neither local maxima nor simple points. This is still not sufficient as a proof, since we must now consider the last but not least characteristic of the MB algorithm: *the parallel removal.*

4.3 Parallel computing of the skeleton

As a matter of fact, simplicity according to Definition 5 is an *individual* characteristic. In general, the parallel removal of simple points leads to *disconnection*, with the disappearance of the two-pixel-thick lines. To cope with this major difficulty, three different approaches have been devised:

• *Directional sub-iterations:* Rosenfeld [9] proved that simple points could be removed in parallel without changing topology if one allows the pixels to be removed in one direction only (e.g. east border simple points). In this kind of algorithms, that we now denote DS-algorithms, the thinning process peels the image removing border pixels in one direction after the other. The most efficient of these algorithms operate in two passes (e.g. N and W, followed by S and E) [19]. The characterization of the skeletal points is in general simplified [8], but the number of iterations multiplied by two. Note that, in such a case the skeleton is by nature non isotropic, since the result depends on the order in the processing directions.

• *Semi-parallel sub-iterations:* by degenerating the parallelism according, for example, to the respective parity of the lines and columns, one may *clusterize* the pixels, and compute a parallel iteration of the skeleton by processing as much passes as there are pixels in the cluster [12]. The result of these algorithms (SP-algorithms) is also sensitive to the choice of the order in the sub-iterations, and furthermore, it produces zigzagging artifacts on straight lines, as pointed out in [10].

• *Fully parallel:* removing non-skeletal pixels in all directions without changing the topology is possible provided that *larger* neighborhoods are examined [6]. As a consequence, if the *number of iterations* is reduced in comparison to multi-subiterations algorithms, the *computational cost of one iteration* is always larger, to such a point that their speed may be smaller in some cases. See for example, [18], [21], and [5]. A general scheme to prove the soundness of FP-algorithms is provided by Ronse [15], who extended the notion of simplicity to *sets of points*, and proved that the parallel notion of set-simplicity is equivalent to the existence of an order according to which all points in the set can be removed sequentially, every point being simple when it is removed [14].

Now the good news is that the MB algorithm proves to belong to the FP-class: *points that match pattern (α) but not (β) actually form a simple set!* In [15], Ronse gave minimal conditions for checking connectivity preservation of parallel thinning algorithms. Theorem 3.5 together with Proposition 3.6 of this paper leads to the following property: A parallel thinning algorithm preserves 8-connectivity if: (1) The pixels removed are simple, (2) A connected component contained in a 2x2 square cannot be completely removed, and (3) Two 4-connected pixels removed at the same time would have been removed as well if treated sequentially.

Condition (1) has been proven in Section 4.2. Condition (2) is obviously met, as a pixel is removed only if it is 4-connected to an interior pixel. To see that Condition (3) is also satisfied, let us notice that two 4-adjacent pixels are simultaneously removed only if they match the pattern showed on Figure 5, in which none of the two pixels belongs to the pattern (α) of the other. This proves

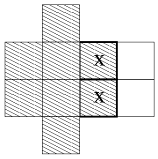

Fig. 5. Case of two pixels 4-connected.

that the MB algorithm is an FP one. As such, its number of iterations before convergence equals the maximum of the distance map Φ (cf. Definition 4), i.e. half the thickness of the biggest object in the image. Finally, removing simple sets, but as big as possible, while preserving the median axis, through such a compact isotropic procedure is the remarkable achievement of the MB algorithm. Its superiority can be appreciated on Table 5 which shows a comparison with several (fully or not) parallel algorithms: MB is nearly twice as fast as the fastest known algorithm !

Table 5. Comparing the cost of parallel thinning algorithms. d stands for the maximal object thickness (i.e. the diameter of the biggest ball contained in the image). The number of elementary operations given for the other algorithms is sometimes only an estimation, as the quoted papers did not always provide a logic minimization of their algorithm. Note that [5] was ranked second among ten algorithms in [11], and the best FP one.

Algorithm	Type	Size of neighborhood examined	Number of elementary operations required
Jang & Chin [8]	DS	7	$16 \times d$
Cardoner & Thomas [3]	DS	7	$20 \times d$
Stewart [18]	FP	19	$30 \times d$
Wu & Tsai [21]	FP	11	$30 \times d$
Guo & Hall [5]	FP	11	$40 \times d$
Manzanera & Bernard (MB)	FP	13	$9 \times d$

4.4 Isotropy and One-pixel Thickness

The MB algorithm computes the skeleton by logical operations that are based on the matching of patterns (α) and (β), with all their $\pi/2$ rotated configurations. Then these operators are completely symmetrical. As the algorithm is fully parallel (no spatial subiteration), this guarantees isotropy. Then the presence of two-pixel-thick lines in the skeleton is expected. If required, one-pixel thickness can be obtained by using two directional passes at a very low extra cost. For example, if we compute a North, then a West point deletion with the Jang & Chin algorithm, the post-processing has a cost of 16 operations (whereas an MB iteration costs 18).

5 Conclusion

We have presented a new parallel iterative thinning algorithm which computes the skeleton of an image in what we believe to be the purest form. This skeleton must (a) contain the median axis for the diamond distance, (b) preserve topology and (c) be obtained through a purely isotropic procedure.

To remove pixels from the original image, the MB algorithm fulfils the three former requirements by checking two pixel conditions that we may call "definitely not a local maximum" (i.e. *matching* α) and "semi-simple" (i.e. *not matching* β). While extremely simple to compute, the combination of these two conditions allows a most efficient fully parallel (or one-pass) operation of the MB algorithm. With the help of systematic logic minimization, this results into a dramatically low computational cost in terms of logical operations needed for the complete execution of the algorithm.

In spite of its (much) faster computation, the MB skeleton is comparable in quality to those produced by other algorithms, as shown by the experimental results. While its preferred implementation is on SIMD machines that compute binary logic, the speed advantage of the MB algorithm should remain on most computers, parallel or not, due to the fundamental character of the Shannon measure.

References

1. C. Arcelli. A condition for digital points removal. *Signal Processing*, 1-4:283–285, 1979.
2. H. Blum. A tranformation for extracting new descriptors of shape. In *Proc. Symposium Models for the perception of speech and visual form*, pages 362–380. W.Wathen-Dunn ed. M.I.T. Press Cambridge MA, 1967.
3. Rafael Cardoner and Federico Thomas. Residuals + Directional Gaps = Skeletons. *Pattern Recognition Letters*, 18:343–353, 1997.
4. Ching-Sung Chen and Wen-Hsiang Tsai. A new Fast One-Pass Thinning Algorithm and its Parallel Harware Implementation. *Pattern Recognition Letters*, 11:471–477, 1990.
5. Zicheng Guo and Richard W. Hall. Fast Fully Parallel Thinning Algorithms. *Computer Vision, Graphics and Image Processing*, 55-3:317–328, 1992.
6. Richard W. Hall. Optimal Small Operator Supports for Fully Parallel Thinning algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 15-8:828–833, 1993.
7. C.J. Hilditch. Linear skeletons from square cupboards. *Machine Intelligence*, 4:403–420, 1969.

8. Ben-Kwei Jang and Roland T. Chin. Analysis of thinning algorithms using mathematical morphology. *IEEE Transactions on pattern analysis and machine intelligence*, 12-6:514–551, 1990.
9. T.Y. Kong and A. Rosenfeld. Digital Topology : Introduction and Survey. *Computer Vision, Graphics and Image Processing*, 48:357–393, 1989.
10. Louisa Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies : A Comprehensive Survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14:869–885, 1992.
11. Louisa Lam and Chin Y. Suen. An Evaluation of Parallel Thinning Algorithms for Character Recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 17-9:914–919, 1995.
12. Jan Olszewski. A flexible thinning algorithm allowing parallel, sequential and distributed application. *ACM Transactions on Mathematical Software*, 18-1:35–45, 1992.
13. F. Paillet, D. Mercier, and T.M. Bernard. Making the most of 15k lambda2 silicon area for a digital retina PE. In T.M. Bernard, editor, *Proc. SPIE, Vol. 3410, Advanced Focal Plane Arrays and Electronic Cameras*, pages 158–167, Zürich, Switzerland, May 1998.
14. Christian Ronse. A topological characterization of thinning. *Theoretical Computer Science*, 43:31–41, 1986.
15. Christian Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Discrete Applied Mathematics*, 21:67–79, 1988.
16. D. Rutovitz. Pattern Recognition. *J.R. Statist. Soc.*, 129:504–530, 1966.
17. C.E. Shannon. *The synthesis of two-terminal switching circuits.* Bell Systems Tech. J., 1949.
18. Alan Stewart. A one-pass thinning algorithm with interference guards. *Pattern Recognition Letters*, 15:825–832, 1994.
19. S. Suzuki and K. Abe. Binary picture thinning by an iterative parallel two-subcycle operation. *Pattern Recognition*, 10-3:297–307, 1987.
20. Luc Vincent. *Algorithmes morphologiques à base de file d'attente et de lacets. Extension aux graphes.* PhD thesis, Ecole Nationale Supérieure des Mines de Paris, May 1990.
21. Rei-Yao Wu and Wen-Hsiang Tsai. A new One-Pass Parallel Thinning Algorithm for binary images. *Pattern Recognition Letters*, 13:715–723, 1992.
22. S. Yokoi, J.I. Toriwaki, and T. Fukumura. Topological properties in digitized binary pictures. *Sytems, Computers, Controls*, 4-6:32–39, 1973.