Parallel Thinning Algorithms Based on Ronse's Sufficient Conditions for Topology Preservation

Gábor Németh[a] and Kálmán Palágyi[b]

Department of Image Processing and Computer Graphics,
University of Szeged, Hungary
E-mail: [a]gnemeth@inf.u-szeged.hu; [b]palagyi@inf.u-szeged.hu

Thinning is a widely used pre–processing step in digital image processing and pattern recognition. It is an iterative layer by layer erosion until only the "skeletons" of the objects are left. This paper presents three thinning algorithms according to three kinds of endpoint criteria. The strategy which is used is called fully parallel, which means that the same parallel operator is applied at each iteration. The proposed algorithms are based on Ronse's sufficient conditions for topology preservation.

Keywords: shape representation, skeletonization, thinning, topology, topology preservation.

1. Introduction

Skeleton is a region-based shape descriptor which summarizes the general form of objects. An illustrative definition of the skeleton is given using the prairie-fire analogy: the object boundary is set on fire and the skeleton is formed by the loci where the fire fronts meet and extinguish each other [2].

Thinning is based on a digital simulation of the fire front propagation: the border points of a binary object that satisfy certain topological and geometric constraints are deleted in iteration steps [7,13].

A *2D binary picture* [6] is a mapping that assigns a value of 0 or 1 to each point with integer coordinates in the 2D digital space denoted by $\mathbb{Z}^2$. Points having the value of 1 are called *black* points, and those with a zero value are called *white* ones. Black points form the objects of a picture. White points form the background and the cavities of the picture.

A *reduction operator* transforms a binary picture only by changing some black points to white ones (which is referred to as the *deletion* of 1's). A *parallel reduction operator* deletes all points satisfying its deletion condition simultaneously. The *support* of a reduction operator applied to a black point p is the minimal set of points whose values determine whether p is deleted by the operator [4]. A reduction operator does *not* preserve topology [5] if any object in the input picture is split (into several objects) or is completely deleted, any cavity in the input picture

Progress in Combinatorial Image Analysis. Editors: Petra Wiederhold and Reneta Barneva

978-981-08-0000-0

is merged with the background or another cavity, or a cavity is created where there was none in the input picture.

Thinning algorithms use operators that delete some points from the actual object boundary in a topology preserving way. Parallel thinning algorithms delete a set of points simultaneously in each phase of the thinning process. A parallel thinning algorithm is *fully parallel* if it applies the same parallel operator at every iteration [4].

This paper presents a fully parallel thinning approach based on Ronse's sufficient conditions for parallel reduction operators. Three algorithms are proposed according to the three traditional definitions of an endpoint [4].

2. Basic Notions and Results

Some concepts of digital topology and their key results will be given below as they will be needed later on.

Let $p = (p_x, p_y)$ and $q = (q_x, q_y)$ be two points in $\mathbb{Z}^2$ and let us denote by $d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$ their Euclidean distance. These two points p and q are *4–adjacent* if $d(p,q) \leq 1$ and and they are *8–adjacent* if $d(p,q) \leq \sqrt{2}$. Let $N_j(p)$ (for $j = 4, 8$) denote the set of points *j–adjacent* to point p and $N_j^*(p) = N_j(p) \setminus \{p\}$ refers to the set consisting of the proper j–adjacent neighbors of p. The sequence of distinct points $\langle x_0, x_1, \ldots, x_n \rangle$ is called a *j–path* (for $j = 4, 8$) of length n from point x_0 to point x_n in a non–empty set of points X if each point of the sequence is in X and x_i is j–adjacent to x_{i-1} for each $1 \leq i \leq n$. Note that a single point is a j–path of length 0. Two points are said to be *j–connected* in the set X if there is a j–path in X between them. A set of points X is *j–connected* in the set of points $Y \supseteq X$ if any two points in X are j–connected in Y.

The *2D binary (8,4) digital picture* $\mathcal{P}$ is a quadruple $\mathcal{P} = (\mathbb{Z}^2, 8, 4, B)$ [6]. Each element of $\mathbb{Z}^2$ is called a *point* of $\mathcal{P}$. Each point in $B \subseteq \mathbb{Z}^2$ is called a *black point* and has a value of 1 assigned to it. Each point in $\mathbb{Z}^2 \setminus B$ is called a *white point* and has a value of 0 assigned to it. 8–adjacency and 4–adjacency are, respectively, used for the black points and the white ones. A *black component* is a maximal 8–connected set of points in B, while a *white component* is a maximal 4–connected set of points in $\mathbb{Z}^2 \setminus B$. It is assumed that any picture is finite (i.e., it contains finitely many black points).

A black point is called a *border point* in $(8,4)$ pictures if it is 4–adjacent to at least one white point. A black point p is called an *interior point* if it is not border point. A black point is called a *simple point* if its deletion preserves the topology of the picture [6].

There are numerous characterizations of simple points. One of them is stated as follows:

Theorem 1. *[6] Black point p is simple in picture $(\mathbb{Z}^2, 8, 4, B)$ if and only if all of the following conditions hold:*

(1) p is a border point.
(2) The set $N_8^*(p)$ contains exactly one black 8–component.

Note that the simplicity of point p in a $(8,4)$ picture is a local property; it can be decided in view of $N_8^*(p)$.

Thinning algorithms use operators that delete some simple points which are not end points, since preserving endpoints provides important geometrical information relative to the shape of the objects. The proposed algorithms consider the following three characterizations of an endpoint.

Definition 1. *Black point p in picture $(\mathbb{Z}^2, 8, 4, B)$ is an endpoint of "type i" if the i-th condition holds [4]:*

(1) $N_8^*(p) \cap B = \{q\}$ (i.e., exactly one black point is 8–adjacent to p).
(2) $N_8^*(p) \cap B = \{q\}$, or $N_8^*(p) \cap B = \{r, s\}$ and $r \in N_8^*(s)$.
(3) $N_8^*(p) \cap B = \{q\}$, or $N_8^*(p) \cap B = \{r, s\}$ and $r \in N_4^*(s)$.

Note that condition 2 of Definition 1 is less restrictive than condition 3 (and condition 1 is more restrictive than conditions 2 and 3), but Hall [4] used this order of the three traditional definitions of an endpoint. Figure 1 shows some examples of the considered types of endpoints.

Parallel reduction operators delete a set of black points and not just a single simple point. Ronse's sufficient conditions for parallel reduction operators of (8,4) pictures [10] can be described as follows.

Theorem 2. *A parallel reduction operator is topology preserving for (8,4) pictures if all of the following conditions hold:*

(1) Only simple points are deleted.
(2) For any two 4-adjacent points, p and q are deleted, p is simple after q is deleted, or q is simple after the deletion of p.
(3) No "small" black component contained in a 2×2 lattice square is deleted completely.

We propose a new thinning method that is based on Ronse's sufficient conditions for topology preservation (see Theorem 2). Three different algorithms are given according to the considered types of endpoints (see Definition 1).

1	0	0
0	p	0
0	0	0

endpoint of types 1, 2, and 3

0	1	0
1	p	0
0	0	0

endpoint of type 2

1	1	0
0	p	0
0	0	0

endpoint of types 2 and 3

Fig. 1. Examples of endpoints.

3. The Proposed Fully Parallel Thinning Algorithms

Each fully parallel thinning algorithm can be sketched as follows:

repeat
  simultaneous deletion of "deletable" points
until no points are deleted;

The kernel of the **repeat** cycle corresponds to one iteration step of the process. According to the fully parallel strategy, the same parallel operator is applied at each iteration. Existing fully parallel thinning algorithms differ from each other in just one regard: which types of points are considered as "deletable". The deletable points of the proposed algorithms are defined as follows:

Definition 2. *Black point p in an (8,4) picture $\mathcal{P}$ is i-self-deletable $(i = 1, 2, 3)$ if p is not an endpoint of type i (see Definition 1) and it is simple in $\mathcal{P}$ (see Theorem 1).*

Definition 3. *Black point p in $(\mathbb{Z}^2, 8, 4, B)$ is i-double-deletable $(i = 1, 2, 3)$ if p is i-self-deletable (see Definition 2), and for any i-self-deletable point $q \in N_4^*(p)$, point p is simple in picture $(\mathbb{Z}^2, 8, 4, B \backslash \{q\})$ or point q is simple in picture $(\mathbb{Z}^2, 8, 4, B \backslash \{p\})$.*

Definition 4. *Black point p in an (8,4) picture is i-square-deletable $(i = 1, 2, 3)$ if it does not match the configurations shown in Figure 2.*

Definition 5. *Black point p is i-deletable $(i = 1, 2, 3)$ in an (8,4) picture if p is i-self-deletable (see Definition 2), i-double-deletable (see Definition 3), and i-square-deletable (see Definition 4).*

Figure 3 shows some examples of 1-self-deletable, 1-double-deletable, 1-square-deletable, and 1-deletable points.

We are now ready to state the proposed fully parallel thinning algorithms:

Algorithms Ronse-*i* $(i = 1, 2, 3)$

Input: picture $(\mathbb{Z}^2, 8, 4, X)$
Output: picture $(\mathbb{Z}^2, 8, 4, Y)$

0	0	0	0
0	p	0	0
0	1	1	0
0	0	0	0

a

0	0	0	0
0	p	1	0
0	0	1	0
0	0	0	0

b

0	0	0	0
0	p	1	0
0	1	0	0
0	0	0	0

c

0	0	0	0
0	0	p	0
0	1	1	0
0	0	0	0

d

0	0	0	0
0	p	1	0
0	1	1	0
0	0	0	0

e

Fig. 2. Critical "small" black components. Black point p is *not* 1-square-deletable in all configurations (a)-(e) and it is neither 2- nor 3-square-deletable in configuration (e). It is easy to see that p is simple, not endpoint of type i $(i = 1, 2, 3$, see Definition 1), and it comes first in the lexicographic ordering of the "small" black component in question.

0	0	1	0
0	p	0	0
0	0	0	0

a

0	0	1	0
0	p	0	0
0	1	0	0

b

0	0	1	0
0	p	1	0
0	1	0	0

c

0	0	0	0
0	p	1	0
0	0	1	0

d

0	0	0	0
0	p	1	0
0	1	1	1

e

Fig. 3. Examples of 1-self-deletable, 1-double-deletable, 1-square-deletable, and 1-deletable points in (8,4) pictures. Black point p is *not* 1-self-deletable in configurations (a) and (b) (since p is endpoint of type 1 in (a) and it is not simple in (b)), but it is 1-self-deletable in the remaining three configurations (c)-(e). Black point p is *not* 1-double-deletable in (a), (b), and (c), but it is 1-double-deletable in (d) and (e). Black point p is *not* 1-square-deletable in configuration (d) (see Figure 2d), but it is 1-square-deletable in (e). Black point p 1-deletable in (e), but it is *not* 1-deletable in the remaining four configurations (a)-(d).

$Y = X$
repeat
  $D = \{ \ p \mid p$ is i-deletable in $Y \ \}$
  $Y = Y \setminus D$
until $D = \emptyset$

4. Discussion

It is easy to see that the deletability of a point p (according to Definition 5) can be decided by examining the 5×5 neighborhood of p, see Figure 4. Therefore, the support of the proposed algorithms is 5×5. It is not surprising since Rosenfeld proved in [11] that a fully parallel algorithm with 3×3 support is not suitable for shrinking (i.e., for transforming black components with no cavities to single-pixel residues).

	d	d	d	
d	s	s	s	d
d	s	p	s	d
d	s	s	s	d
q	d	d	d	q

Ronse-1

	d	d	d	
d	s	s	s	d
d	s	p	s	d
d	s	s	s	d
	d	d	d	q

Ronse-2 and Ronse-3

Fig. 4. Local supports of the proposed algorithms. Self-deletability (see Definition 2) of the black point "p" is based on points marked "s", double-deletability (see Definition 3) adds points marked "d", and square-deletability (see Definition 4) needs additional points marked "q".

A good thinning algorithm needs to take the following four aspects into account:

(1) force the "skeleton" to retain the topology of the original object;
(2) provide "shape preservation";
(3) force the "skeleton" to be in its geometrically correct position in the "center" of the black components;
(4) produce "thin" curves (i.e., the desired "width" of the "skeleton" is one point).

We can prove the topological correctness (the first requirement) of the proposed algorithms:

Theorem 3. *Algorithms Ronse-i* ($i = 1, 2, 3$) *are topology preserving.*

Proof. To prove it, we show that parallel reductions of the proposed algorithms (i.e., deletion of the sets of i-deletable points ($i = 1, 2, 3$)) satisfies Ronse's sufficient conditions for topology preserving parallel reduction operators of (8,4) pictures (see Theorem 2).

(1) i-self-deletable points (see Definition 2) are simple points, hence their deletion satisfies Condition 1 of Theorem 2.
(2) It can be readily seen that deletion of i-double-deletable points (see Definition 3), satisfies Condition 2 of Theorem 2.
(3) It is easy to see that an i-square-deletable point (see Definition 4 and Figure 2) is not the first element of a "small" black component according to the row-by-row traversal. Hence, deletion of i-square-deletable point does not delete a "small" component completely, thus Condition 3 of Theorem 2 holds.

Therefore, deletion of i-deletable points (that are self-deletable, double-deletable, and square-deletable) satisfy all conditions of Theorem 2. □

Shape preservation (the second requirement) is very important. For example, a black component like "b" cannot be thinned into a black component like "o". Thinning algorithms retain the shape of the original (elongated) black components by preserving endpoints. It is easy to see that algorithm Ronse-i does not delete an endpoint of type i since an i-deletable point (see Definitions 2 and 5) is not an endpoint of type i ($i = 1, 2, 3$).

Geometrical correctness (the third requirement) of the extracted skeleton is mostly achieved by the fully parallel thinning approach. Since the proposed algorithms use a symmetric support (see "s" and "d" points in Figure 4) for "large" black components (i.e., that are not "small" ones, see Condition 3 of Theorem 2), they are uniformly shrunk in the main directions.

Note that an algorithm that uses endpoint of type 2 is not able to create always 8-connected curve segments that are not 4-connected, see Figure 5. In addition, 2-point thick vertical and horizontal segments cannot be thinned due to the proposed thinning strategy (i.e., that is based on Ronse's conditions), see Figure 5.

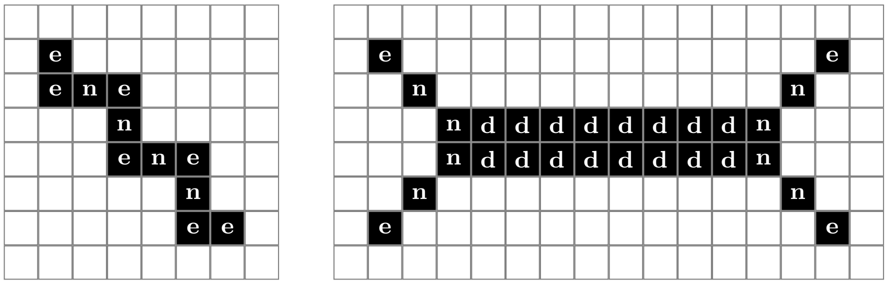

Fig. 5. Example of a "thick" black component that cannot be thinned by algorithm Ronse-2 (left). Simple points denoted by "**e**" are all endpoints of type 2 and all points denoted by "**n**" are not simple. Hence there is no 2-deletable point in this 2-point thick black component. Example of a black component that cannot be thinned by algorithms Ronse-i ($i = 1, 2, 3$) (right). All black points denoted by "**e**" are endpoints of types 1, 2, and 3, all points denoted by "**n**" are not simple, and all points denoted by "**d**" are not i-double-deletable. Hence there is no i-deletable ($i = 1, 2, 3$) point in this black component.

Hence, we can state that the fourth requirement about maximal thinning is not satisfied by the proposed algorithms. Fortunately it is easy to overcome this problem here. If unit thickness is desired, then the "skeletons" of our algorithms can be corrected by just one iteration step of any algorithm capable of producing maximally thinned results.

Note that thinning (as a skeletonization approach) is sensitive to coarse object boundaries. As a result, the "skeleton" produced generally includes some unwanted segments that must be removed by a pruning step [12].

5. Implementation

One may think that the proposed algorithms are time consuming and it is rather difficult to implement them on standard sequential computers. Thus we sketch here an efficient and fairly general implementation method. It can be used for the other parallel and some sequential thinning algorithms as well [8].

The proposed implementation uses three (algorithm–specific) pre–calculated look-up-tables to encode the deletion rules of the three thinning algorithms Ronse-i ($i = 1, 2, 3$). Since the support of the algorithm Ronse-1 contains 22 points with the exception of the point p in question (see Figure 4), its table has 2^{22} entries of 1 bit in size. It is not hard to see that this look-up-table requires just 0.5 MB of storage space in memory. The support of both algorithms Ronse-2 and Ronse-3 contains only 21 points, hence their look-up-tables need just 0.25 MB.

In addition, two lists are used to speed up the process: one for storing the border points in the current picture (since thinning can only delete border-points, thus the repeated scans/traverses of the entire array storing the picture are avoided); the second list is to store all deletable points in the current phase of the process. At each iteration, the deletable points are found and deleted, and the list of border points is updated accordingly.

A critic, however, might say that it does not make sense to suggest parallel algorithms to be implemented on sequential computers. In order to settle any doubt, consider the convolution and the Fourier transform. These frequently used operations are parallel by nature (since each component of their results can be calculated simultaneously), but their efficient implementations on standard sequential computers are extremely important.

6. Results

In experiments the proposed three fully parallel thinning algorithms were tested on numerous objects of different shapes, and their results were compared with "skeletons" produced by three existing fully parallel thinning algorithms: AK^2, EM93, and PAV81 [1,3,9]. Here we present some illustrative examples below (Figures 6–9). Numbers in parentheses mean the computation time (in sec.). Our algorithms were run under Linux on an Intel Pentium 4 CPU 2.80 GHz PC. (Note that

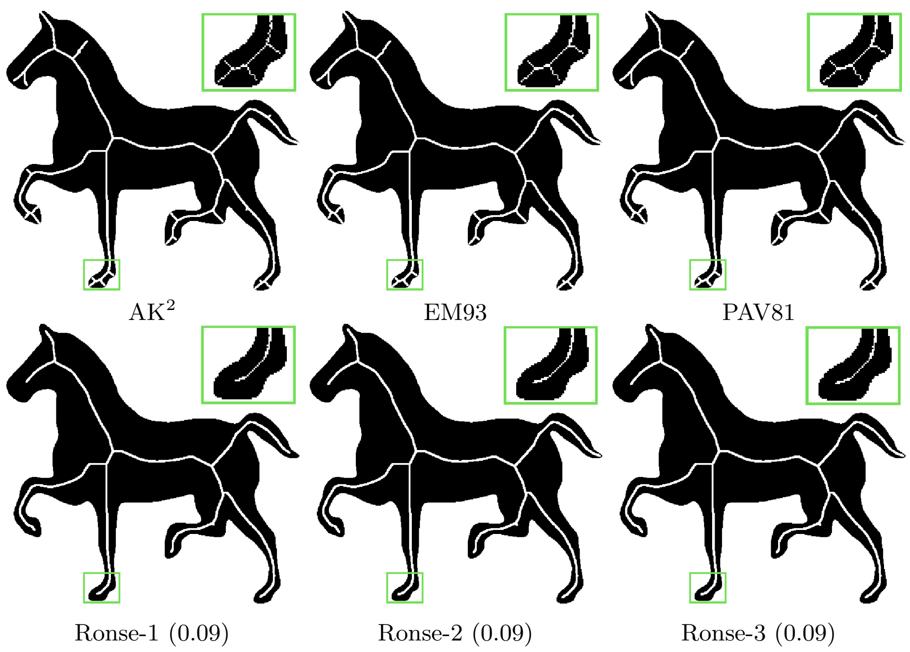

Fig. 6. "Skeletons" produced by the six algorithms under comparison superimposed on a 473 × 451 image of a horse.

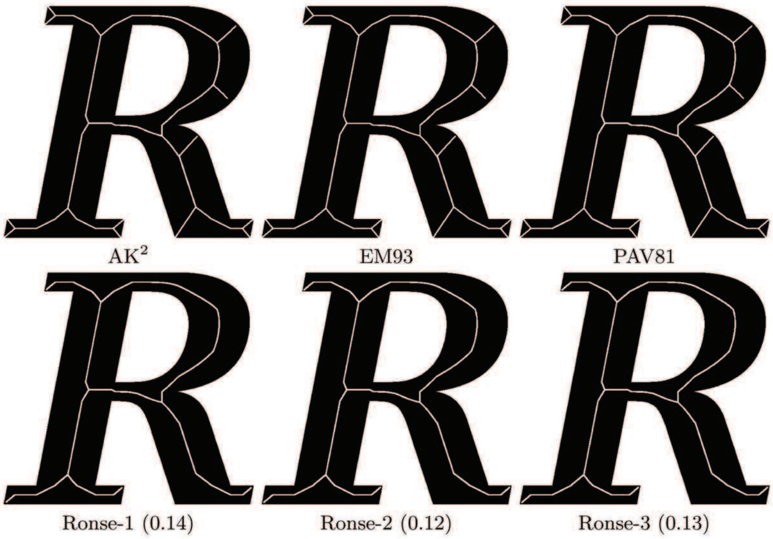

Fig. 7. "Skeletons" produced by the six algorithms under comparison superimposed on a 600×557 image of a letter "R".

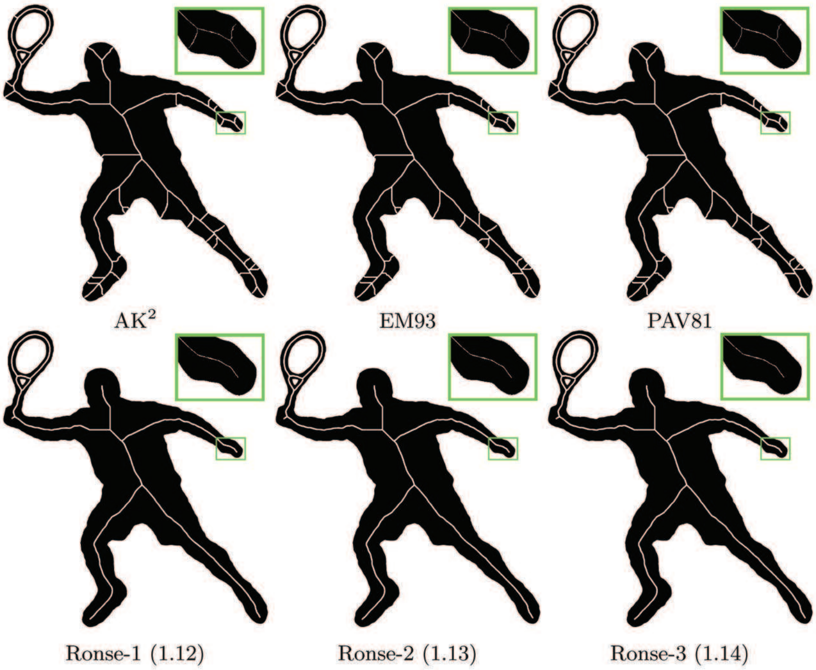

Fig. 8. "Skeletons" produced by the six algorithms under comparison superimposed on a 1832×2054 image of a tennis player.

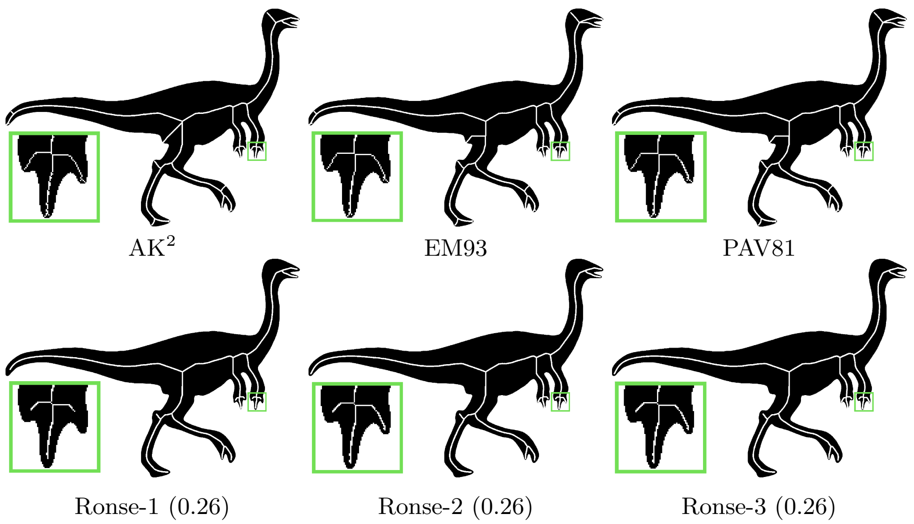

Fig. 9. "Skeletons" produced by the six algorithms under comparison superimposed on a 1188×881 image of a dinosaur.

just the thinning process itself was considered here; reading the input volume and writing the output image were not taken into account.)

One can state that the proposed algorithms Ronse-i ($i = 1, 2, 3$) produce less spurious branches than AK^2, EM93, and PAV81 do. It is why our algorithms do not preserve corners, see Figure 10.

Note that algorithm AK^2 leads to a skeleton which contains the medial axis of the object. This allows the perfect reconstruction of the object from its skeleton. This is not the case for the proposed algorithms.

0	0	0	0
0	p	1	x
0	1	1	x
0	x	x	x

0	0	0	0
0	p	1	.
0	1	1	.
.	1	.	.

0	0	0	.
0	p	1	1
0	1	1	.
.	1	.	.

Fig. 10. Corner configurations in which black point p is i-deletable ($i = 1, 2, 3$), therefore, all proposed algorithms delete p. Some existing parallel thinning algorithms preserve such corners, hence they get spurious branches. Notations: at least one point marked "x" is 1 and every "." may be either a "0" or a "1". Note that all reflections and rotations (where the rotation angles are 90°, 180°, and 270°) of these three base configurations give corners as well.

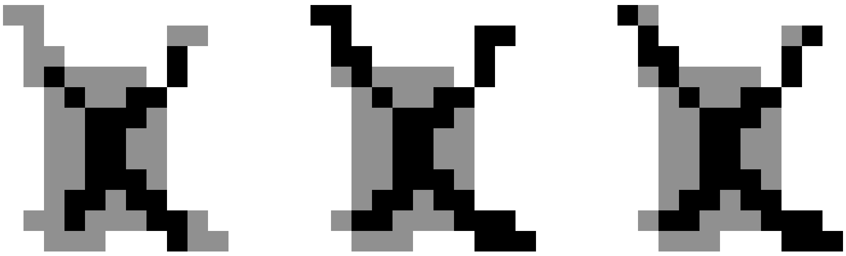

Fig. 11. "Skeletons" of a small object. Black squares (pixels) correspond to skeletal points and deleted object points are displayed in grey. Numbers in parentheses mean the count of skeletal points.

The proposed algorithms Ronse-i ($i = 1, 2, 3$) produce similar "skeletons" for large objects with smooth boundaries (see Figures 6–9). In order to show that our three algorithms are really different, we present an example (see Figure 11).

7. Conclusion

In this paper, we presented a new family of fully parallel thinning algorithms.

Thinning algorithms are generally constructed in the following way: first the thinning strategy and the deletion rules are figured out, then the topological correctness is proved. In the case of the proposed algorithms we used the converse way: first we considered Ronse's sufficient conditions for parallel reduction operators to preserve topology, then the deletion rules were accommodated to them. There are always round–trips against strategy and correctness, but the starting point is fairly important. In our algorithms, the correctness is predestinated, hence no complex proof–part is needed.

Our experiments show that our algorithms can produce "reasonable skeletons". Thanks to the efficient implementation method, the proposed algorithms are very fast.

Acknowledgements

The authors would like to thank Prof. Michel Couprie (Département Informatique, ESIEE Paris, et Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est ESIEE 2, France) for supplying the source code for algorithm AK^2.

This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

References

1. Bertrand, G., Couprie, M.: Two-dimensional parallel thinning algorithms based on critical kernels. Journal of Mathematical Imaging and Vision **31** (2008) 35–56
2. Blum, H.: A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form, MIT Press (1967) 362–380
3. Eckhardt, U., Maderlechner, G.: Invariant thinning. International Journal of Pattern Recognition and Artificial Intelligence **7** (1993) 1115–1144
4. Hall, R.W.: Parallel connectivity–preserving thinning algorithms. In: Topological algorithms for digital image processing, Kong, T.Y., Rosenfeld, A. (Eds.), Elsevier Science (1996) 145–179
5. Kong, T.Y.: On topology preservation in 2–d and 3–d thinning. International Journal of Pattern Recognition and Artificial Intelligence **9** (1995) 813–844
6. Kong, T.Y., Rosenfeld, A.: Digital topology: Introduction and survey. Computer Vision, Graphics, and Image Processing **48** (1989) 357–393
7. Lam, L., Lee, S.-W., Suen, C.Y.: Thinning methodologies — A comprehensive survey, IEEE Trans. Pattern Analysis and Machine Intelligence **14** (1992) 869–885
8. Palágyi, K.: A 3D fully parallel surface-thinning algorithm. Theoretical Computer Science **406** (2008) 119–135
9. Pavlidis, T.: A flexible parallel thinning algorithm. In: Proc. IEEE Comput. Soc. Conf. on Pattern Recognition and Image Processing (1981) 162–167
10. Ronse, C.: Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. Discrete Applied Mathematics **21** (1988) 67–79
11. Rosenfeld, A.: Connectivity in digital pictures. J. ACM **17** (1970) 146-160
12. Shaked, D., Bruckstein, A.: Pruning medial axes. Computer Vision and Image Understanding **69** (1998) 156–169
13. Suen, C.Y., Wang P.S.P. (eds.): Thinning methodologies for pattern recognition. Series in Machine Perception and Artificial Intelligence **8**, World Scientific (1994)