Topology Preserving Parallel Thinning Algorithms

Gábor Németh, Kálmán Palágyi

Department of Image Processing and Computer Graphics, University of Szeged, Szeged, Hungary

Received 3 February 2010; accepted 8 October 2010

ABSTRACT: Thinning is an iterative object reduction technique for extracting medial curves from binary objects. During a thinning process, some border points that satisfy certain topological and geometric constraints are deleted in iteration steps. Parallel thinning algorithms are composed of parallel reduction operators that delete a set of object points simultaneously. This article presents 21 parallel thinning algorithms for (8,4) binary pictures that are derived from the sufficient conditions for topology preservation accommodated to the three parallel thinning approaches. Int J Imaging Syst Technol, 21, 37–44, 2011; Published online in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/ima.20272

Key words: shape representation; skeletonization; thinning; parallel thinning algorithms; topology preservation

INTRODUCTION

Skeleton is a frequently applied region-based shape descriptor. 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 (Blum, 1967). It provides simpler objects that summarize the general forms of the original ones.

Thinning processes simulate the front propagation in digital spaces: some points in the outmost layer of a binary object that satisfy certain topological and geometric constraints are deleted in iteration steps, and the entire process is repeated until stability is achieved (Lam et al., 1992; Suen and Wang, 1994). Thinning in 2D digital spaces extracts skeleton-like shape features (that are called medial curves) in a topology preserving way (Kong and Rosenfeld, 1989; Kong 1995).

Parallel thinning algorithms are composed of parallel reduction operators that delete a set of object points simultaneously (Hall, 1996). Ronse gave sufficient (but not necessary) conditions for parallel reduction operators to preserve topology (Ronse, 1988).

There are three strategies for parallel thinning: directional, subfield-based, and fully parallel (Lam et al., 1992; Suen and Wang, 1994; Hall, 1996). This article presents 21 parallel thinning algorithms: six of them are directional, 12 use subfield-based approach (six of them follow the conventional scheme, and the remaining six ones use our novel iteration-level endpoint checking), and the last three algorithms belong to the fully parallel type. Our algorithms are based on some sufficient conditions for topology preservation.

BASIC NOTIONS AND RESULTS

In this article, we use the fundamental concepts of digital topology as reviewed by Kong and Rosenfeld (Kong and Rosenfeld, 1989).

Let $\boldsymbol{p}$ be a point in the digital space $\boldsymbol{Z}^2$. Let us denote $N_4(\boldsymbol{p}) = \{\boldsymbol{p}, \boldsymbol{p_N}, \boldsymbol{p_E}, \boldsymbol{p_S}, \boldsymbol{p_W}\}$ and $N_8(\boldsymbol{p}) = N_4(\boldsymbol{p}) \cup \{\boldsymbol{p_{NE}}, \boldsymbol{p_{SE}}, \boldsymbol{p_{SW}}, \boldsymbol{p_{NW}}\}$ the sets of points that are 4-adjacent and 8-adjacent to point $\boldsymbol{p}$, respectively (see Fig. 1), and $\boldsymbol{N_i^*}(\boldsymbol{p}) = \boldsymbol{N_i}(\boldsymbol{p})\setminus\{\boldsymbol{p}\}$ refers to the set containing the proper $\boldsymbol{i}$-adjacent neighbors of $\boldsymbol{p}$ (for $\boldsymbol{i}$ = 4,8).

A 2D binary (8,4) digital picture $\boldsymbol{P}$ is a quadruple $\boldsymbol{P} = (\boldsymbol{Z}^2, 8, 4, \boldsymbol{B})$ (Kong and Rosenfeld, 1989). The elements of $\boldsymbol{Z}^2$ are called points of $\boldsymbol{P}$. Each point in $\boldsymbol{B} \subseteq \boldsymbol{Z}^2$ is called a black point or an object point and value of ''1'' is assigned to it. Each point in $\boldsymbol{Z}^2\setminus\boldsymbol{B}$ is called a white point and has a value of ''0''. Eight-connectivity and four-connectivity are, respectively, used for the black components and the white ones.

A black point is a border point in (8,4) pictures if it is 4-adjacent to at least one white point. A border point is called an N-border point if the point marked $\boldsymbol{p}_N$ in Figure 1 is white. We can define E-, S-, and W-border points in the same way. A border point is called an NE-border point if at least one of the two points marked $\boldsymbol{p}_N$ and $\boldsymbol{p}_E$ in Figure 1 is white. Similarly, a border point is called an SW-border point if at least one of the two points marked $\boldsymbol{p}_S$ and $\boldsymbol{p}_W$ in Figure 1 is white. A black point is called an interior point if it is not a border point.

A black point is called a simple point if its deletion preserves the topology of the picture (Kong and Rosenfeld, 1989). The support (i.e., the minimal set of points whose values determine the property in question) of the operator which detects simple points is 3×3.

Thinning algorithms use operators that delete some simple points which are not endpoints, since preserving endpoints provides important geometrical information relative to the shape of the

Correspondence to: Kálmán Palágyi; e-mail: palagyi@inf.u-szeged.hu

The preliminary version of this article was presented in 2009 at the 13th International Workshop on Combinatorial Image Analysis.

Grant sponsors: This research was supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency and the NKTH-OTKA-CNK80370 Grant.

p_{NW}	p_N	p_{NE}
p_W	p	p_E
p_{SW}	p_S	p_{SE}

Figure 1. Notations for 4-adjacency and 8-adjacency.

objects. The algorithms presented in ''Special sufficient conditions for topology preservation'' section consider the following three characterizations of an endpoint.

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

1. $N_8^*(p) \cap B = \{q\}$.
2. $N_8^*(p) \cap B = \{q\}$, or $N_8^*(p) \cap B = \{q, r\}$ and $r \in N_4^*(q)$
3. $N_8^*(p) \cap B = \{q\}$, or $N_8^*(p) \cap B = \{q, r\}$ and $r \in N_8^*(q)$

Note that condition 1 is more restrictive than condition 2, and condition 2 is more restrictive than condition 3. That is why we interchanged conditions 2 and 3. Note that Hall (Hall, 1996) used different order in the three traditional definitions of an endpoint. Figure 2 shows some examples of the considered types of endpoints. It is not hard to see that any endpoint of type $\boldsymbol{i}$ is simple ($\boldsymbol{i}$ = 1,2,3).

Parallel reduction operators delete a set of black points and not just a single simple point. The following theorem about sufficient conditions for parallel reduction operators of (8,4) pictures is derived from Ronse's results (Ronse, 1988):

Theorem 1. (Kong, 1995) 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, $\boldsymbol{p}$ and $\boldsymbol{q}$ are deleted, $\boldsymbol{p}$ is simple after $\boldsymbol{q}$ is deleted, or $\boldsymbol{q}$ is simple after the deletion of $\boldsymbol{p}$.
3. No ''small'' black component contained in a 2×2 square is deleted completely.

It is easy to see that there are 10 possible ''small'' black components contained in a 2×2 square (see Fig. 3).

SPECIAL SUFFICIENT CONDITIONS FOR TOPOLOGY PRESERVATION

Theorem 1 provides a general method of verifying that a parallel thinning algorithm preserves topology (Kong, 1995). In this section, we present new sufficient conditions for topology preservation as a basis for designing new parallel thinning algorithms.

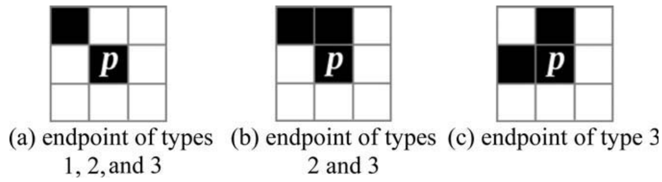


Figure 2. Examples of the considered three types of endpoints.

Let us consider the first three ''small'' black components (see Figs. 3a–3c).

Proposition 1. If a parallel reduction operator satisfies Condition 1 of Theorem 1, then Condition 3 of Theorem 1 is granted for the ''small'' black component depicted in Figure 3a.

It is obvious since this black component is formed by a single nonsimple point.

Proposition 2. If a parallel reduction operator $\boldsymbol{O}$ satisfies Condition 2 of Theorem 1, then none of the two ''small'' black components depicted in Figures 3b–3c can be deleted completely by $\boldsymbol{O}$.

Both considered ''small'' black components are formed by two 4-adjacent simple points. If one of them is deleted, then the remaining point is a single nonsimple point.

We are now ready to state some new sufficient conditions for topology preservation.

Theorem 2. Let $\boldsymbol{O}$ be a parallel reduction operator. Let $\boldsymbol{p}$ be any black point in any picture $\boldsymbol{P} = (\boldsymbol{Z}^2,8,4,\boldsymbol{B})$ such that $\boldsymbol{p}$ is deleted by $\boldsymbol{O}$. The operator $\boldsymbol{O}$ is topology preserving for (8,4) pictures if all of the following conditions hold:

1. Point $\boldsymbol{p}$ is simple in $\boldsymbol{P}$.
2. For any simple point $q \in N_4^*(p) \cap$B, $\boldsymbol{p}$ is simple in picture $(\boldsymbol{Z}^2,8,4,\boldsymbol{B}\backslash\{\boldsymbol{q}\})$, or $\boldsymbol{q}$ is simple in picture $(\boldsymbol{Z}^2,8,4,\boldsymbol{B}\backslash\{\boldsymbol{p}\})$.
3. Point $\boldsymbol{p}$ does not coincide with the points marked ''**x**'' in the seven black components depicted in Figures 3d–3j.

Proof. We need to show that all conditions of Theorem 1 are satisfied.

- Condition 1 of Theorem 2 corresponds to Condition 1 of Theorem 1.
- Condition 2 of Theorem 2 corresponds to Condition 2 of Theorem 1.
- The first three ''small'' black components (see Figs. 3a–3c) cannot be deleted completely by Propositions 1 and 2. In the

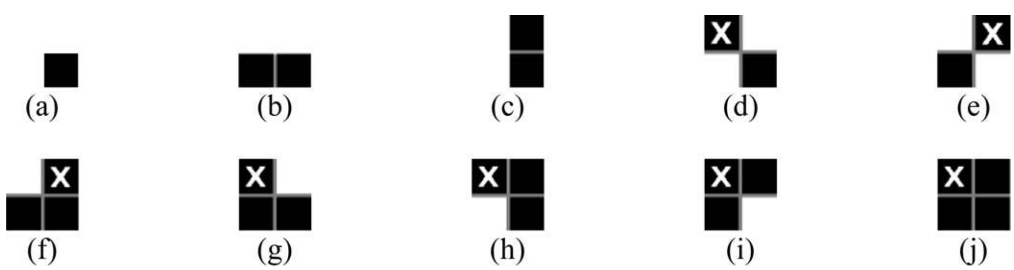


Figure 3. Possible black components contained in a 2×2 square. Points marked "**x**" are the "protected" elements of the last seven black components (see Condition 3 of Theorem 2).

remaining seven ''small'' black components (see Figs. 3d–3j) points marked ''**x**'' cannot be deleted by Condition 3 of Theorem 2. Hence Condition 3 of Theorem 1 hold. ■

The general Theorem 2 can be simplified by considering endpoint preserving parallel reductions. Let us state some properties of the seven ''small'' black components that are to be taken into account in Condition 3 of Theorem 2 (see Figs. 3d–3j).

Proposition 3. Let O be a parallel reduction operator that preserves endpoints of type 1. Condition 3 of Theorem 1 is granted for the two ''small'' black components depicted in Figures 3d–3e.

This holds since both black components are composed of two endpoints of type 1. Hence none of these black components is deleted completely.

Proposition 4. Let O be a parallel reduction operator that preserves endpoints of type i (i = 2,3). Condition 3 of Theorem 1 is granted for the six ''small'' black components depicted in Figures 3d–3i.

This holds since each of these black components contains at least one endpoint of type i (i = 2,3).

In consequences of Propositions 3 and 4, Theorem 2 can be modified for parallel reductions that preserve some type of endpoints.

Theorem 3. Let O be a parallel reduction operator that considers endpoints of type i (i= 1,2,3). Let p be any black point in any picture $P = (Z^2,8,4,B)$ such that p is deleted by O. The operator O is topology preserving for (8,4) pictures if all of the following conditions hold:

1. Point p is simple and not an endpoint of type i in P.
2. For any point $q \in N_4^*(p) \cap B$ being simple but not an endpoint of type i, p is simple in picture $(Z^2,8,4,B\backslash\{q\})$, or q is simple in picture $(Z^2,8,4,B\backslash\{p\})$.
3. Third condition is as follows:
   - If i = 1, then point p does not coincide with the points marked ''**x**'' in the five ''small'' black components depicted in Figures 3f–3j.
   - If $i \in \{2,3\}$, then point p does not coincide with the point marked ''**x**'' in the ''small'' black component depicted in Figure 3j.

Notice that the Condition 3 of Theorem 2 was remarkably changed by reducing the number of ''small'' black components to be considered.

21 VARIATIONS ON THINNING ALGORITHMS

In this section various parallel thinning algorithms are presented that are based on some sufficient conditions for topology preservation (see Theorem 3) and the three characterizations of an endpoint (see Definition 1).

A. Directional Thinning Algorithms. An iteration step of directional (or border sequential, or subiteration) algorithms is divided into a number of successive subiterations, where only border points of a certain kind can be deleted. Note that 2-subiteration (Zhang and Suen, 1984; Lü and Wang, 1986; Chen and Hsu, 1989; Gou and Hall, 1989; Zhang and Wang, 1996) and 4-subiteration (Stefanelli and Rosenfeld, 1971; Rosenfeld, 1975; Davies and Plummer, 1981; Arcelli et al., 1994) algorithms have been developed for this task. The support of their parallel reductions is usually 3 × 3.

In this subsection we present three 2-subiteration algorithms denoted by **SI-2-*i*** and three 4-subiteration ones denoted by **SI-4-*i***, where i (i = 1,2,3) is the type of endpoints (see Definition 1) to be preserved.

The two deletion directions of the 2-subiteration algorithms are $\mathbf{d_2}(1)$ = NE and $\mathbf{d_2}(2)$ = SW. At the j-th subiteration, some $\mathbf{d_2}(j)$-border points can be deleted (j = 1,2).

In the case of 4-subiteration directional algorithms, the four deletion directions are $\mathbf{d_4}(1)$ = N, $\mathbf{d_4}(2)$ = E, $\mathbf{d_4}(3)$ = S, and $\mathbf{d_4}(4)$ = W. Some $\mathbf{d_4}(j)$-border points can be deleted by the j-th subiteration (j = 1,2,3,4).

We are now ready to sketch the three 2-subiteration directional algorithms **SI-2-*i*** and the three 4-subiteration algorithms **SI-4-*i*** (i = 1,2,3):

Algorithm SI-*k*-*i*
Input: picture $(Z^2,8,4,X)$
Output: picture $(Z^2,8,4,Y)$
// endpoints of type *i* are preserved
$Y = X$
repeat
// one iteration step
for j=1 **to** k **do**
// j-th subiteration
$D(j)=\{p \mid p$ is $SI\text{-}k\text{-}i\text{-}d_k(j)$-deletable in $\mathbf{Y}\}$
$\mathbf{Y} = \mathbf{Y} \setminus D(j)$
until $D(1)\cup\ldots\cup D(k) = \emptyset$

Before we define the considered types of deletable points, some properties of the ''small'' black components to be taken into consideration are stated.

Proposition 5. Consider the NE-subiteration of algorithm **SI-2-*i*** (i = 1,2,3). There is at least one point in both ''small'' black components depicted in Figures 3g and 3j that is not an NE-border point. Consequently, these two ''small'' black components need not to be checked in the NE-subiteration.

Proposition 6. Consider the SW-subiteration of algorithm **SI-2-*i*** (i = 1,2,3). There is at least one point in both ''small'' black components depicted in Figures 3h and 3j that is not an SW-border point. Consequently, these two ''small'' black components need not to be checked in the SW-subiteration.

Proposition 7. *Consider* b-border points (for b = N,E,S,W). There is at least one point in each of the five ''small'' black components in Figures 3f–3j that is not a b-border point. Hence that point cannot be deleted by the reduction assigned to any subiteration of the 4-subiteration algorithm **SI-4-*i*** (i = 1,2,3). Consequently, the five ''small'' black components depicted in Figures 3f–3j need not to be checked in the 4-subiteration case.

In consequences of Propositions 5 and 6, ***SI*-2-*i*-*b***-deletable points can be defined as follows:

Definition 2. A black point p is ***SI*-2-*i*-*b***-deletable (i = 1,2,3; b = NE,SW) if all of the following conditions hold:

1. Point p is simple and b-border but not an endpoint of type i (see Condition 1 of Theorem 3).
2. For any point $q \in N_4^*(p)$ being simple and b-border but not an endpoint of type i, p is simple after q is deleted, or q is simple after the deletion of p (see Condition 2 of Theorem 3).
3. For the i = 1 case (see Condition 3 of Theorem 3):

- At the NE-subiteration, point p does not coincide with points marked ''**x**'' in the three ''small'' black components depicted in Figures 3f, 3h, and 3i.
- At the SW-subiteration, point p does not coincide with points marked ''**x**'' in the three ''small'' black components depicted in Figures 3f, 3g, and 3i.

In consequence of Proposition 7, we can define ***SI*-4-*i*-*j***-deletable points as follows:

Definition 3. A black point p is ***SI*-4-*i*-*b***-deletable (i = 1,2,3; b = N,E,S,W) if both of the following conditions hold:

1. Point p is simple and b-border but not an endpoint of type i (see Condition 1 of Theorem 3).
2. For any point $q \in N_4^*(p)$ being simple and b-border but not an endpoint of type i, p is simple after q is deleted, or q is simple after the deletion of p (see Condition 2 of Theorem 3).

It can readily be seen that deletable points of the proposed directional algorithms **SI-k-*i*** (k = 2,4; i = 1,2,3) are derived directly from the corresponding sufficient conditions for topology preservation. Hence all of the six algorithms are topology preserving.

Note that subiteration-based algorithms are sensitive to the orders of the deletion directions. Choosing different orders of directions yields various medial curves. To reduce their asymmetry, the order of the directions within each iteration step can be selected randomly.

B. Subfield-Based Thinning Algorithms. The second approach for parallel thinning is the subfield-based (or subfield-sequential) strategy. Subfield-based thinning algorithms with 3 × 3 support partition the digital space into some subsets which are alternatively activated, and only some points in the active subfield can be deleted (Preston and Duff, 1984; Guo and Hall, 1989; Gökmen and Hall, 1990; Neusius et al., 1992). In the case of $\boldsymbol{Z}^2$, two kinds of partitions into two and four subfields were introduced (Hall, 1996). Let us denote $\mathbf{SF}_k(j)$ the j-th subfield in the k-subfield partitions (k = 2,4; j = 0,1,. . .,k-1; see Fig. 4). Without loss of generality, we can assume that $(0,0) \in SF_k(0)$ (i.e., the origin as a point in $\boldsymbol{Z}^2$ is in the 0th subfield).

Now, let us state some properties of these two partitions.

Proposition 8. For the 2-subfield case (see Fig. 4a), two points p and $q \in N_8^*(p)$ are in the same subfield if $q \in N_8(p) \setminus N_4(p)$.

Proposition 9. For the 4-subfield case (see Fig. 4b), two points p and $q \in N_8^*(p)$ are not in the same subfield.

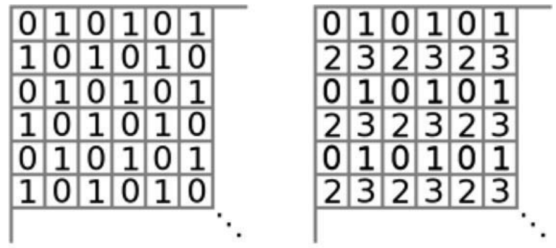
In consequences of Propositions 8 and 9, Theorem 3 can be dramatically simplified for subfield-based reductions.

Theorem 4. (Hall, 1996) A k-subfield (k = 2,4) parallel reduction operator that does not delete endpoints of type i (i = 1,2,3) is topology preserving for (8,4) pictures if only simple points are deleted.

Notice that the Condition 1 of Theorem 1 need to be checked since Conditions 2 and 3 automatically hold.

In this subsection we present 12 subfield-based parallel thinning algorithms that are derived from a single sufficient condition for topology preservation (see Theorem 4). These algorithms use the 2-subfield and 4-subfield partitions (see Fig. 4) and the three conventional characterizations of an endpoint (see Definition 1).

The first six k-subfield (k = 2,4) algorithms **SF-*k*-*i*** using endpoints of type i (i = 1,2,3) are described as follows:



(a) (b)

Figure 4. Partitions of Z^2 into 2 (a) and 4 (b) subfields. For the k-subfield case, all points marked j are in the subfield SFk(j) (k = 2,4; j = 0,1,. . .,k−1).

Algorithm SF-*k*-*i*
Input: picture ($\mathbf{Z}^2$,8,4,**X**)
Output: picture ($\mathbf{Z}^2$,8,4,**Y**)
// endpoints of type i are preserved
Y = **X**
repeat
// one iteration step
for j = 0 **to** k-1 **do**
// subfield $\mathbf{SF_k(j)}$ is activated
$D(j) = \{p \mid p$ is ***SF*-*i*-**deletable in $\mathbf{Y} \cap \mathbf{SF_k}(j)\}$
$\mathbf{Y} = \mathbf{Y} \setminus D(j)$
until $D(0) \cup \ldots \cup D(k-1) = \emptyset$

According to Theorem 4, we can define ***SF*-*i*-**deletable points as follows:

Definition 4. A black point is ***SF*-*i*-**deletable (i = 1,2,3) if it is simple but not an endpoint of type i.

To reduce the noise sensitivity and the number of skeletal points, we introduce a modified subfield-based thinning scheme. It takes the endpoints into consideration at the beginning of iteration steps, instead of preserving them in each parallel reduction as it is accustomed in existing subfield-based thinning algorithms.

The following six k-subfield (k = 2,4) algorithms **SF-IL-*k*-*i*** preserving endpoints of type i (i = 1,2,3) use the iteration-level endpoint checking scheme:

Algorithm SF-IL-*k*-*i*
Input: picture ($\boldsymbol{Z}^2$,8,4,$\boldsymbol{X}$)
Output: picture ($\boldsymbol{Z}^2$,8,4,$\boldsymbol{Y}$)
// endpoints of type i are preserved
$\boldsymbol{Y} = \boldsymbol{X}$
repeat
// one iteration step
$\boldsymbol{E} = \{p \mid p$ is a border point but not an endpoint of type i in $\boldsymbol{Y}\}$
for j = 0 **to** k-1 **do**
// subfield $\mathbf{SF_k}(j)$ is activated
$D(j) = \{p \mid p$ is ***SF*-*IL*-**deletable in $\boldsymbol{E} \cap \mathbf{SF_k}(j)\}$
$\boldsymbol{Y} = \boldsymbol{Y} \setminus D(j)$
until $D(0) \cup \ldots \cup D(k-1) = \emptyset$

Definition 5. A black point is ***SF*-*IL*-**deletable if it is simple.

It can readily be seen that all the presented 12 subfield-based thinning algorithms are topology preserving, since Theorem 4 holds for their deletable points.

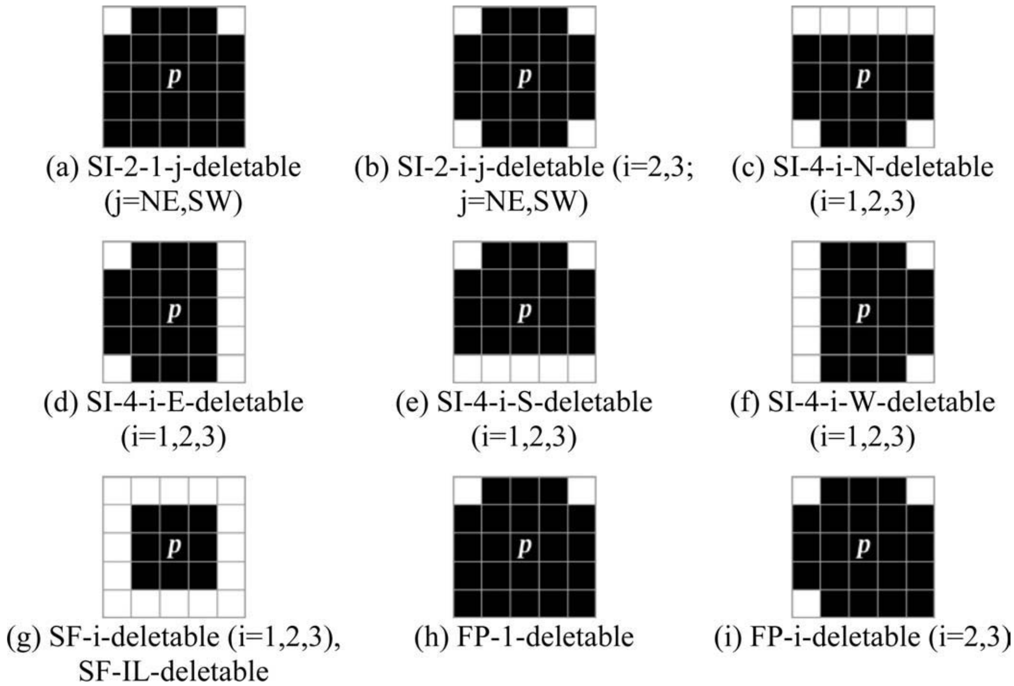


Figure 5. Supports of the parallel reduction operators assigned to the presented 21 parallel thinning algorithms.

Note that each subfield-based algorithm is sensitive to the order of the subfields. If we choose another order in subfield-activation, we may get another algorithm. We can use a randomly selected order of subfields in each iteration step.

C. Fully Parallel Thinning Algorithms. The third type of parallel thinning algorithms use the fully parallel approach (Rutovitz, 1966; Guo and Hall, 1992; Wu and Tsai, 1992; Manzanera et al, 2002). In this case, the same parallel reduction operator is applied at every phase of the thinning process. To preserve topology, the support of that operator is larger than 3×3; some additional points are needed that are in the 5×5 neighborhood. Note that some existing fully parallel thinning algorithms use asymmetric supports (Hall, 1996).

In this subsection we present three fully parallel thinning algorithms that use the three considered types of endpoints. Note that we have published these algorithms in (Németh and Palágyi, 2009).

Algorithm **FP-*i*** using endpoints of type $\boldsymbol{i}$ ($\boldsymbol{i} = 1{,}2{,}3$) is outlined as follows:

Algorithm FP-*i*
Input: picture $(\boldsymbol{Z}^2, 8, 4, \boldsymbol{X})$
Output: picture $(\boldsymbol{Z}^2, 8, 4, \boldsymbol{Y})$
  // endpoints of type $\boldsymbol{i}$ are preserved
  $\boldsymbol{Y} = \boldsymbol{X}$
  repeat
    // one iteration step
    $\boldsymbol{D} = \{\boldsymbol{p} \mid \boldsymbol{p}$ is ***FP-i***-deletable in $\boldsymbol{Y}\}$
    $\boldsymbol{Y} = \boldsymbol{Y} \setminus \boldsymbol{D}$
  until $\boldsymbol{D} = \emptyset$

According to Theorem 3, we can define ***SF-i***-deletable points as follows:

Definition 6. A black point $\boldsymbol{p}$ is ***FP*-1**-deletable if all of the following conditions hold:

1. Point $\boldsymbol{p}$ is simple but not an endpoint of type 1 (see Condition 1 of Theorem 3).
2. For any point $\boldsymbol{q} \in N_4^*(\boldsymbol{p})$ being simple but not an endpoint of type 1, $\boldsymbol{p}$ is simple after $\boldsymbol{q}$ is deleted, or $\boldsymbol{q}$ is simple after the deletion of $\boldsymbol{p}$ (see Condition 2 of Theorem 3).
3. Point $\boldsymbol{p}$ does not coincide with the points marked ''**x**'' in the five ''small'' black components depicted in Figures 3f–3j (see Condition 3 of Theorem 3).

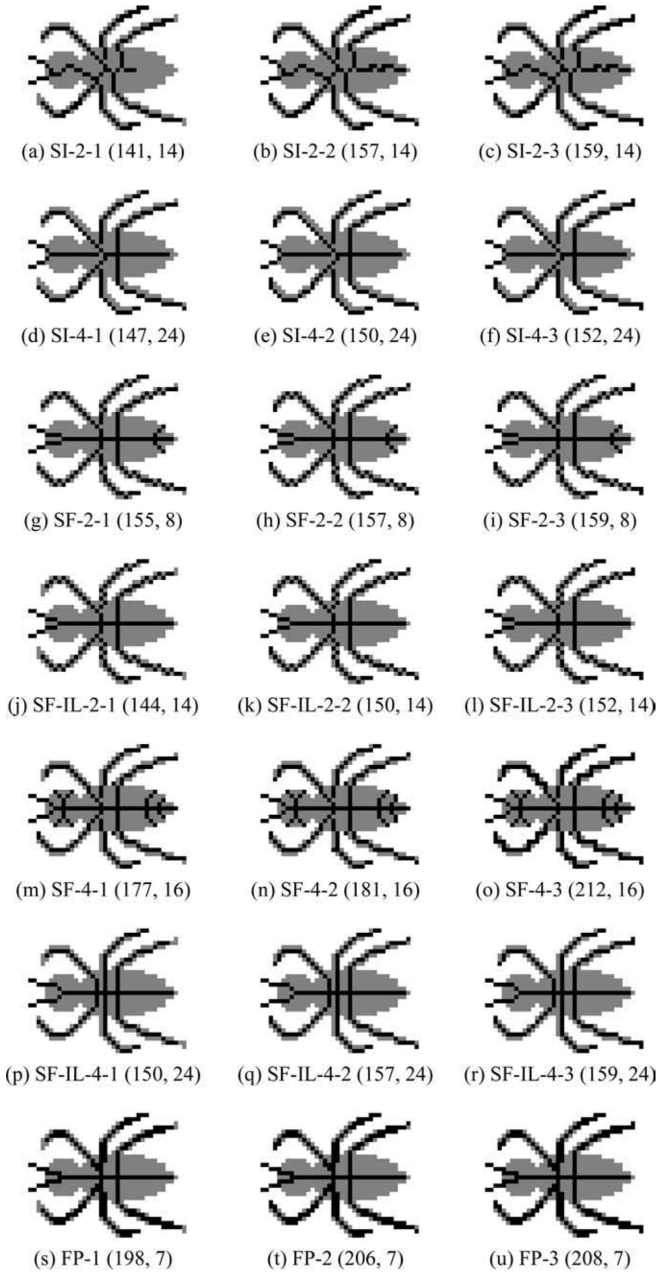
Definition 7. A black point $\boldsymbol{p}$ is ***FP-i***-deletable ($\boldsymbol{i} = 2{,}3$) if all of the following conditions hold:

1. Point $\boldsymbol{p}$ is simple but not an endpoint of type $\boldsymbol{i}$ ($\boldsymbol{i} = 2{,}3$) (see Condition 1 of Theorem 3).
2. For any point $\boldsymbol{q} \in N_4^*(\boldsymbol{p})$ being simple but not an endpoint of type $\boldsymbol{i}$ ($\boldsymbol{i} = 2{,}3$), $\boldsymbol{p}$ is simple after $\boldsymbol{q}$ is deleted, or $\boldsymbol{q}$ is simple after the deletion of $\boldsymbol{p}$ (see Condition 2 of Theorem 3).
3. Point $\boldsymbol{p}$ does not coincide with the point marked ''**x**'' in the ''small'' black component depicted in Figure 3j (see Condition 3 of Theorem 3).

It can readily be seen that all the three fully parallel thinning algorithms are topology preserving, since the corresponding sufficient conditions for topology preservation (see Theorem 3) hold for their deletable points.

IMPLEMENTATION

This section will present a method for implementing any parallel reduction operator on a conventional sequential computer. A fairly general framework is proposed, as similar schemes can be used for the other classes of parallel algorithms (Palágyi, 2008).



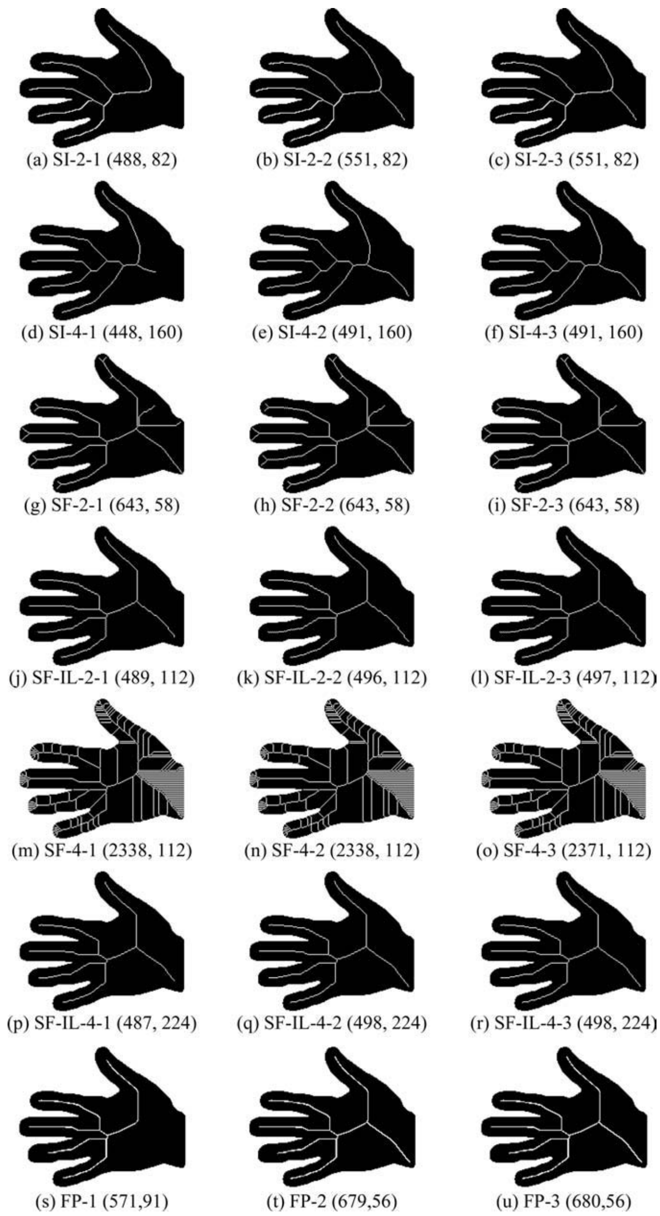
(a) SI-2-1 (141, 14) (b) SI-2-2 (157, 14) (c) SI-2-3 (159, 14)
(d) SI-4-1 (147, 24) (e) SI-4-2 (150, 24) (f) SI-4-3 (152, 24)
(g) SF-2-1 (155, 8) (h) SF-2-2 (157, 8) (i) SF-2-3 (159, 8)
(j) SF-IL-2-1 (144, 14) (k) SF-IL-2-2 (150, 14) (l) SF-IL-2-3 (152, 14)
(m) SF-4-1 (177, 16) (n) SF-4-2 (181, 16) (o) SF-4-3 (212, 16)
(p) SF-IL-4-1 (150, 24) (q) SF-IL-4-2 (157, 24) (r) SF-IL-4-3 (159, 24)
(s) FP-1 (198, 7) (t) FP-2 (206, 7) (u) FP-3 (208, 7)

Figure 6. Medial curves of a bug produced by the presented 21 parallel thinning algorithms. The original 40 $\times$ 33 image contains 474 object points.

The proposed method uses a tricolor array which stores the actual picture to be processed: a value of ''0'' corresponds to the white points, the value of ''1'' corresponds to (black) interior points, and a value of ''2'' is assigned to (black) border points in the actual picture, and a precalculated look-up-table to encode the deletion rule of the parallel reduction in question. In addition, two lists are used to speed up the process: one for storing the border-points in the current picture (since reduction operators of thinning algorithms can only delete border points, the other list is to store all deletable points of the parallel reduction. At the parallel reduction, the deletable points are found and deleted, and the list of border points is updated accordingly.

The sizes of the algorithm-specific look-up-tables depend on the supports of the parallel reduction operators assigned to the 21 thinning algorithms presented in ''21 variations on thinning algorithms'' section. For example, if a support contains 21 points, then the corresponding look-up-table requires just 0.25 MB of storage space in memory. Supports of the proposed algorithms are given in Figure 5.

According to this efficient implementation method, the time complexity of the presented algorithms depends just on the number of object points and the compactness of the objects (i.e., volume to area ratio); but it does not depend on the size of picture which contains the objects to be thinned. Medial curves of large objects containing 1.000.000 points can be extracted within 1 second on a usual PC.



(a) SI-2-1 (488, 82) (b) SI-2-2 (551, 82) (c) SI-2-3 (551, 82)
(d) SI-4-1 (448, 160) (e) SI-4-2 (491, 160) (f) SI-4-3 (491, 160)
(g) SF-2-1 (643, 58) (h) SF-2-2 (643, 58) (i) SF-2-3 (643, 58)
(j) SF-IL-2-1 (489, 112) (k) SF-IL-2-2 (496, 112) (l) SF-IL-2-3 (497, 112)
(m) SF-4-1 (2338, 112) (n) SF-4-2 (2338, 112) (o) SF-4-3 (2371, 112)
(p) SF-IL-4-1 (487, 224) (q) SF-IL-4-2 (498, 224) (r) SF-IL-4-3 (498, 224)
(s) FP-1 (571,91) (t) FP-2 (679,56) (u) FP-3 (680,56)

Figure 7. Medial curves of a hand produced by the presented 21 parallel thinning algorithms. The original 191 $\times$ 162 image contains 16,964 object points.

Figure 8. Medial curves of three characters produced by the presented 21 parallel thinning algorithms. The original 248×85 image contains 10,561 object points.

DISCUSSION AND RESULTS

In experiments the 21 parallel thinning algorithm presented in ''21 variations on thinning algorithms'' section were tested on objects of different shapes. Here we present some illustrative examples below (Figs. 6–8). The produced medial curves are superimposed on the original objects, and pair of numbers ''(***mc***,***ps***)'' mean the number of points in the medial curves (***mc***) and the parallel speed (***ps***). The parallel speed characterizes an algorithm by the number of parallel reduction operators required by the entire thinning process (Hall, 1996).

If we would like to summarize the properties of the presented algorithms, we can state the followings:

- All the 21 algorithms are different from each other (see Fig. 6).
- The 2-subiteration algorithms (SI-2-***i***; ***i*** = 1,2,3) are faster than the 4-subiteration ones (SI-4-***i***; ***i*** = 1,2,3), but they may produce asymmetric medial curves for symmetric objects (see Figs. 6a–6f).
- The 2-subfield algorithms (SF-2-***i*** and SF-IL-2-***i***; ***i*** = 1,2,3) are faster than the 4-subfield ones (SF-4-***i*** and SF-IL-4-***i***; ***i*** = 1,2,3).
- Subfield-based algorithms SF-IL-***k***-***i*** (***k*** = 2,4; ***i*** = 1,2,3) with iteration-level endpoint checking produce much less unwanted side branches than algorithms SF-***k***-***i*** that use the conventional thinning scheme (see Figs. 6–8g–r). Note that unwanted side branches in medial curves can be removed by a pruning process (Shaked and Bruckstein, 1998).
- The 4-subfield algorithms with iteration-level endpoint checking (SF-IL-4-***i***; ***i*** = 1,2,3) are the slowest ones.
- The fully parallel algorithms FP-***i*** (***i*** = 1,2,3) are the fastest ones, but they may produce two-point thick vertical and horizontal segments due to the sufficient conditions for topology preservation accommodated to them (see Figs. 6s–6u). That is why their medial curves contain relatively large numbers of object points. It is easy to overcome this problem by postprocessing. It is not hard to see that just one iteration step of any of the remaining 18 directional and subfield-based algorithms can produce one-point thick segments from those two-point thick ones.

CONCLUSION

We presented 21 parallel thinning algorithms that are working on (8,4) binary pictures. Their deletion rules were derived from sufficient conditions for topology preservation accommodated to them, hence their topological correctness is guaranteed. The proposed algorithms use all the three parallel thinning approaches (i.e., directional, subfield-based, and fully parallel) and the three conventional characterizations of an endpoint. In addition, we proposed a novel iteration-level endpoint checking for subfield-based thinning algorithms.

REFERENCES

C. Arcelli, G. Sanniti Di Baja, and G. Kwok, ''Parallel pattern compression by octagonal propagation,'' In Thinning methodologies for pattern recognition, C.Y. Suen and P.S.P. Wang (Editors), series in Machine Perception and Artificial Intelligence 8, World Scientific, Singapore; New Jersey; London; Hong Kong, 1994, pp. 113–137.

H. Blum, A transformation for extracting new descriptors of shape, models for the perception of speech and visual form, MIT Press, Boston, 1967, 362–380.

Y.S. Chen and W.H. Hsu, A systematic approach for designing 2-subcycle and pseudo 1-subcycle parallel thinning algorithm, Pattern Recognition 22 (1989), 267–282.

E.R. Davies and A.P.N. Plummer, Thinning algorithms: A critique and a new methodology, Pattern Recognit 14 (1981), 53–63.

M. Gökmen and R.W. Hall, Parallel shrinking algorithms using 2-subfields approaches, Comput Vis Graph Image Process 52 (1990), 191–209.

Z. Guo and R.W. Hall, Parallel thinning with two subiteration algorithms, Commun ACM 32 (1989), 359–373.

Z. Guo and R.W. Hall, Fast fully parallel thinning algorithms, CVGIP: Image Understanding 55 (1992), 317–328.

R.W. Hall, ''Parallel connectivity-preserving thinning algorithms,'' In Topological algorithms for digital image processing, T.Y. Kong and A. Rosenfeld, (Editors), Elsevier Science, Amsterdam, 1996, pp. 145–179.

T.Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, Comput Vis Graph Image Process 48 (1989), 357–393.

T.Y. Kong, On topology preservation in 2-d and 3-d thinning, Intern J Pattern Recognit Artif Intell 9 (1995), 813–844.

L. Lam, S.-W. Lee, and C.Y. Suen, Thinning methodologies—A comprehensive survey, IEEE Trans Pattern Analysis Machine Intell 14 (1992), 869–885.

H.E. Lü and P.S.P. Wang, A comment on ''a fast parallel algorithm for thinning digital patterns,'' Commun ACM 29 (1986), 239–242.

A. Manzanera, T.M. Bertrand, F. Prêteux, and B. Longuet, nD skeletonization: A unified mathematical framework, J Electron Imaging 11 (2002), 25–37.

G. Németh and K. Palágyi, ''Parallel thinning algorithms based on Ronse's sufficient conditions for topology preservation,'' In Progress in Combinatorial Image Analysis, P. Wiederhold and R. Barneva (Editors), Research Publishing Services, Singapore, 2009, pp. 183–194.

C. Neusius, J. Olszewski, and D. Scheerer, An efficient distributed thinning algorithm, Parallel Comput 18 (1992), 47–55.

K. Palágyi, A 3D fully parallel surface-thinning algorithm, Theor Comput Sci 406 (2008), 119–135.

K. Preston and M.J.B. Duff, Modern cellular automata—Theory and applications, Plenum Press, New York, London, 1984.

C. Ronse, Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images, Discrete Appl Math 21 (1988), 67–79.

A. Rosenfeld, A characterization of parallel thinning algorithms, Inform Control 29 (1975), 286–291.

D. Rutovitz, Pattern recognition, J R Stat Soc 129 (1966), 504–530.

D. Shaked, A. Bruckstein, Pruning medial axes, Comput Vis Image Underst 69 (1998), 156–169.

R. Stefanelli and R. Rosenfeld, Some parallel thinning algorithm for digital pictures, J ACM 18 (1971), 255–264.

C.Y. Suen and P.S.P. Wang (Editors),Thinning methodologies for pattern recognition, Series in Machine Perception and Artificial Intelligence 8, World Scientific, Singapore; New Jersey; London; Hong Kong, 1994.

R.Y. Wu and W.H. Tsai, A new one-pass parallel thinning algorithm for binary images, Pattern Recognit Lett 13 (1992), 715–723.

T.Y. Zhang and C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun ACM 27 (1984), 236–239.

Y.Y. Zhang and P.S.P. Wang, A parallel thinning with two-subiterations that generates one-pixel-wide skeletons, Proc Int Conf Pattern Recognit (ICPR), 1996, pp. 457–461.