

A Parallel 3D 12-Subiteration Thinning Algorithm

Kálmán Palágyi* and Attila Kuba

Department of Applied Informatics, József Attila University, Szeged H-6701, Hungary

E-mail: palagyi@inf.u-szeged.hu

Received July 11, 1997; revised April 19, 1999; accepted June 22, 1999

Thinning on binary images is an iterative layer by layer erosion until only the “skeletons” of the objects are left. This paper presents an efficient parallel thinning algorithm which produces either curve skeletons or surface skeletons from 3D binary objects. It is important that a curve skeleton is extracted directly (i.e., without creating a surface skeleton). The strategy which is used is called directional: each iteration step is composed of a number of subiterations each of which can be executed in parallel. One iteration step of the proposed algorithm contains 12 subiterations instead of the usual six. The algorithm makes easy implementation possible, since deletable points are given by $3 \times 3 \times 3$ matching templates. The topological correctness for (26, 6) binary pictures is proved. © 1999 Academic Press

1. INTRODUCTION

Skeletonization provides shape features that are extracted from binary image data. It is a common preprocessing operation in raster-to-vector conversion or in pattern recognition. In the 3D Euclidean space, the skeleton consists of the locus of the centers of all the maximal inscribed spheres [5]. A very 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 quench each other [3]. In discrete spaces, the thinning process is a frequently used method for producing an approximation to the *Euclidean skeleton* (that corresponds to the definition) in a topology-preserving way [9]. It based on digital simulation of the fire front propagation: border points of a binary object that satisfy certain topological and geometric constraints are deleted in iteration steps. The entire process is repeated until only the “skeleton” is left.

A *3D binary picture* [9, 8] is a mapping that assigns the value of 0 or 1 to each point with integer coordinates in the 3D digital space denoted by Z^3 . Points having the value of 1 are called black points, while 0's are called white ones. Black points form objects of the picture.

* Corresponding author.

White points form the background and the cavities of the picture. Both the input and the output of a picture operation are pictures. An operation is a reduction if it can delete some black points (i.e., changes them to white) but white points remain the same. A thinning algorithm (as a reduction operation) does *not* preserve topology [8] if

- any object in the input picture is split (into two or more ones) or completely deleted,
- any cavity in the input picture is merged with the background or another cavity, or
- a cavity is created where there was none in the input picture.

There is an additional concept called a hole in 3D pictures. A hole (that doughnuts have) is formed by 0's, but it is not a cavity [9]. Topology preservation implies that eliminating or creating any hole is not allowed.

A *simple* point is an object point whose deletion does not alter the topology of the picture [15]. Thinning algorithms delete simple points [15] which are not end-points, since preserving end-points provides important information relative to the shape of the objects. *Curve thinning* preserves line end-points while *surface thinning* does not delete surface end-points.

Note that the Euclidean skeleton represents some kinds of *local object symmetries* [25]. The skeleton of a 3D object can contain some surface patches (representing mirror symmetry and/or rotational symmetry) and some line segments (representing axial symmetry). The results of the surface thinning algorithms are closer to the 3D Euclidean skeleton than the "skeletons" produced by curve thinning algorithms. Axial symmetry is emphasized by curve thinning and other kinds of symmetries are suppressed.

Most of the existing thinning algorithms are parallel, since the fire front propagation is by nature parallel. Those algorithms delete a set of simple points simultaneously that can alter the topology. A possible approach to overcome this problem is to use *directional strategy*; each iteration step is composed of a number of parallel subiterations where only border points of a certain kind can be deleted in each subiteration. There are six kinds of *major directions* in 3D images on a cubic grid (see Fig. 1a); therefore, 6-subiteration parallel thinning algorithms were generally proposed [2, 7, 10, 16, 17, 26].

This paper presents a new 3D directional approach that uses 12 subiterations, according to the selected 12 *deletion directions* (see Fig. 1b). It demonstrates a possible way for constructing nonconventional directional thinning algorithms. Some experiments are made on synthetic and natural objects and the topology preservation for (26, 6) binary pictures [8, 9] is proved.

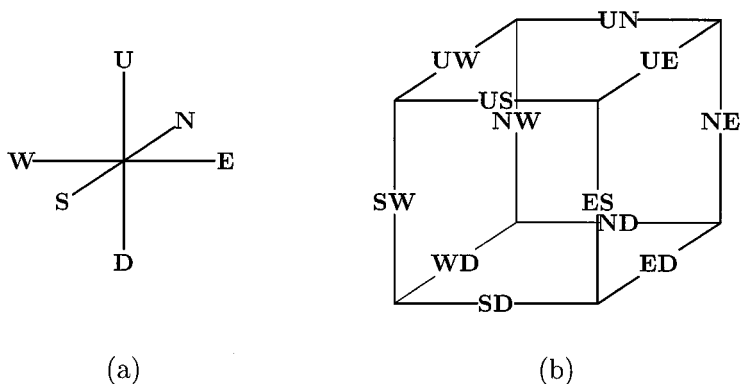


FIG. 1. The six major directions (a) and the proposed 12 deletion directions (b). The 12 directions correspond to the 12-neighbors of a point on a face-centered cubic grid [9].

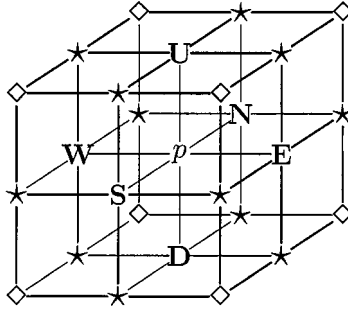


FIG. 2. Frequently used adjacencies in Z^3 . The set $N_6(p)$ contains the central point p and points marked **U**, **N**, **E**, **S**, **W**, and **D**. The set of points $N_{18}(p)$ contains the set $N_6(p)$ and points marked “*.” The set of points $N_{26}(p)$ contains the set $N_{18}(p)$ and points marked “◇.”

2. THINNING METHODOLOGIES

At first some concepts are to be presented. Let p be a point in the 3D digital space Z^3 . Let us denote $N_j(p)$ (for $j = 6, 18, 26$) the set of points j -adjacent to a point p (see Fig. 2). A black point p is said to be a *border point* if the set $N_6(p)$ contains at least one white point. A border point p is called a *U-border point* if the point marked by **U** in Fig. 2 is white. We can define **N**-, **E**-, **S**-, **W**-, and **D**-border points in the same way.

The six points in $N_6(p) \setminus \{p\}$ determine the six *major directions* (see Fig. 1a). There are three kinds of *opposite* (unordered) pair of points in $N_6(p) \setminus \{p\}$ denoted by **UD**, **NS**, and **EW**. The 12 possible *nonopposite* (unordered) pair of points in $N_6(p) \setminus \{p\}$ are denoted by **UN**, **UE**, **US**, **UW**, **NE**, **NW**, **ND**, **ES**, **ED**, **SW**, **SD**, and **WD**. The 12 kinds of nonopposite pairs of points determine the 12 *additional directions*. These can be associated with the 12 edges of a cube; see Fig. 1a. Note that the six major directions correspond to the six faces of a cube.

Each parallel thinning algorithm can be sketched by the program:

```

repeat
  changing “deletable” border points to white
until no points are deleted

```

Existing 3D thinning algorithms differ from each other in two regards:

- how to organize an iteration step (i.e., the kernel of the **repeat** cycle)?,
- which types of border points are regarded as “deletable”?

To answer the first question, three kinds of parallel thinning methodologies have been proposed:

- Algorithms belonging to the first type do not divide an iteration step into subiterations [12, 13]. In order to preserve topology, the existing two *fully parallel* algorithms investigate some points that are in the $5 \times 5 \times 5$ neighborhood but not in the $3 \times 3 \times 3$ neighborhood.

These algorithms can be sketched by the program:

```

repeat
  simultaneous deletion of the border points that satisfy the global condition
until no points are deleted

```

- The second type of algorithms examines the $3 \times 3 \times 3$ neighborhood of each border point. Iteration steps are divided into a number of successive subiterations, where only border points of a certain kind can be deleted in each subiteration. Consequently, each subiteration uses a different deletion rule. These algorithms are called *directional* or *border sequential* ones. Each subiteration is executed in parallel (i.e., all border points satisfying the deletion condition of the actual subiteration are simultaneously deleted). Most of existing parallel thinning algorithms are border sequential. Since there are six kinds of major directions in 3D images, six-subiteration directional thinning algorithms were generally proposed [2, 7, 10, 16, 17, 26]. Srihari *et al.* [24] have reported an early sequential 3D thinning algorithm that uses the 12 additional directions. The algorithm proposed in this paper uses those 12 directions, too. The major six and the additional 12 directions are illustrated in Fig. 1.

A directional algorithm consisting of k subiterations can be sketched by the program:

```

repeat
  for  $i = 1$  to  $k$  do
    simultaneous deletion of the border points that satisfy
    the condition assigned to the  $i$ -th direction
  until no points are deleted

```

- The third approach is the *subfield sequential* method. The set of points Z^3 is divided into more disjoint subsets which are alternatively activated. At a given iteration step, only border points of the active subfield are designated to be deleted.

A subfield based algorithm consisting of k subfields can be sketched by the program:

```

repeat
  for  $i = 1$  to  $k$  do
    simultaneous deletion of the border points in the  $i$ -th subfield
    that satisfy the global condition (assigned to each subfield)
  until no points are deleted

```

Two subfield sequential 3D thinning algorithms working in cubic grid have been proposed so far [1, 22]. Both algorithms use eight subfields. It is not by accident, since using those eight subfields ensures the topology preservation. Note that Palágyi and Kuba proposed a hybrid thinning algorithm [18]. It uses both subfield sequential and directional approaches.

The second important question is: Which types of black points are designated to be deleted? Some algorithms (in each phase) delete all simple points of a given type which are not end points [1, 2, 10], others give the prescribed neighborhood of deletable points [7, 12, 13, 16, 17, 18, 22, 26]. Curve thinning (or medial line thinning) preserves line end-points while surface thinning (or medial surface thinning) do not delete surface end-points [10, 26]. Note that different surface end-point characterizations have been proposed by various authors [1, 2, 7, 10, 12, 19, 26].

3. THE NEW THINNING ALGORITHM

In this section, a new algorithm is presented for thinning 3D binary images. We use directional strategy in which each iteration step contains 12 successive parallel reduction operations according to the 12 directions illustrated in Fig. 1b.

The ordered list of the deletion directions is proposed:

(US, NE, WD, ES, UW, ND, SW, UN, ED, NW, UE, SD).

Note that choosing another order of the deletion directions yields another algorithm. It does not alter the topological correctness, since a correct directional thinning algorithm is composed of topology-preserving reductions.

The proposed order shows a kind of symmetry: the ordered list can be subdivided into four groups of three items, each group contains all six major directions. Therefore, the thinned objects are in their geometrically correct positions (i.e., in the “middle” of the original objects).

Now, the successive parallel reduction operations are to be given. We propose a curve-thinning algorithm and a surface-thinning algorithm. The deletable points are border points of certain types and not end points (i.e. which are not extremities of curves/surfaces).

The proposed algorithm uses the following characterizations of the curve end-points and the surface end-points.

DEFINITION 4.1. A black point p is a *curve end-point* in a picture if the set $N_{26}(p) \setminus \{p\}$ contains exactly one black point.

DEFINITION 4.2. A black point p is a *surface end-point* in a picture if the set $N_6(p)$ contains at least one opposite pair of white points. (Note that each curve end-point is surface end-point.)

Deletable points in a subiteration are given by a set of $3 \times 3 \times 3$ matching templates. A black point is deletable if at least one template in the set of templates matches it. Templates are usually described by three kinds of elements, “•” (black), “○” (white), and “·” (“don’t care”), where “don’t care” matches either black or white point in a given picture. In order to reduce the number of masks we use additional notations.

The first subiteration assigned to the deletion direction **US** can delete certain **U**- or **S**-border points; the second subiteration associated with the deletion direction **NE** attempt to delete **N**- or **E**-border points, and so on.

The set of templates \mathcal{T}_{US} (containing 14 templates) is presented in Fig. 3. It belongs to the first subiteration of the curve-thinning algorithm. The deletable points of the other subiterations can be obtained by proper rotations and/or reflections of the templates in Fig. 3. Note that template positions marked “x” are used for preserving curve end-points.

The set of templates \mathcal{T}'_{US} (containing only six templates) assigned to the first subiteration of our surface-thinning algorithm is given by Fig. 4. Note that template positions marked “x” and “y” are used for preserving surface end-points. It is easy to see that \mathcal{T}'_{US} is derived from \mathcal{T}_{US} . Templates that delete surface end-points are omitted or modified. Note that the redundant ones are also left out. For example, $\mathbf{T3} \notin \mathcal{T}'_{US}$, since if $\mathbf{T3}$ matches a surface end-point then that point can be deleted by $\mathbf{T7} \in \mathcal{T}'_{US}$ or $\mathbf{T8} \in \mathcal{T}'_{US}$, too. Therefore, template $\mathbf{T3}$ is regarded as redundant in surface thinning. The deletable points of the other 11 subiterations of the surface-thinning algorithm can be obtained by proper rotations and/or reflections of the templates in Fig. 4.

4. DISCUSSION

4.1. Template Building

There are some thinning algorithms that give their deletable points by sets of templates (or formulae) [7, 12, 13, 16–18, 22, 26]. A number of configurations around a point are classified as deletable, but they form only a proper subset of simple points of certain kinds.

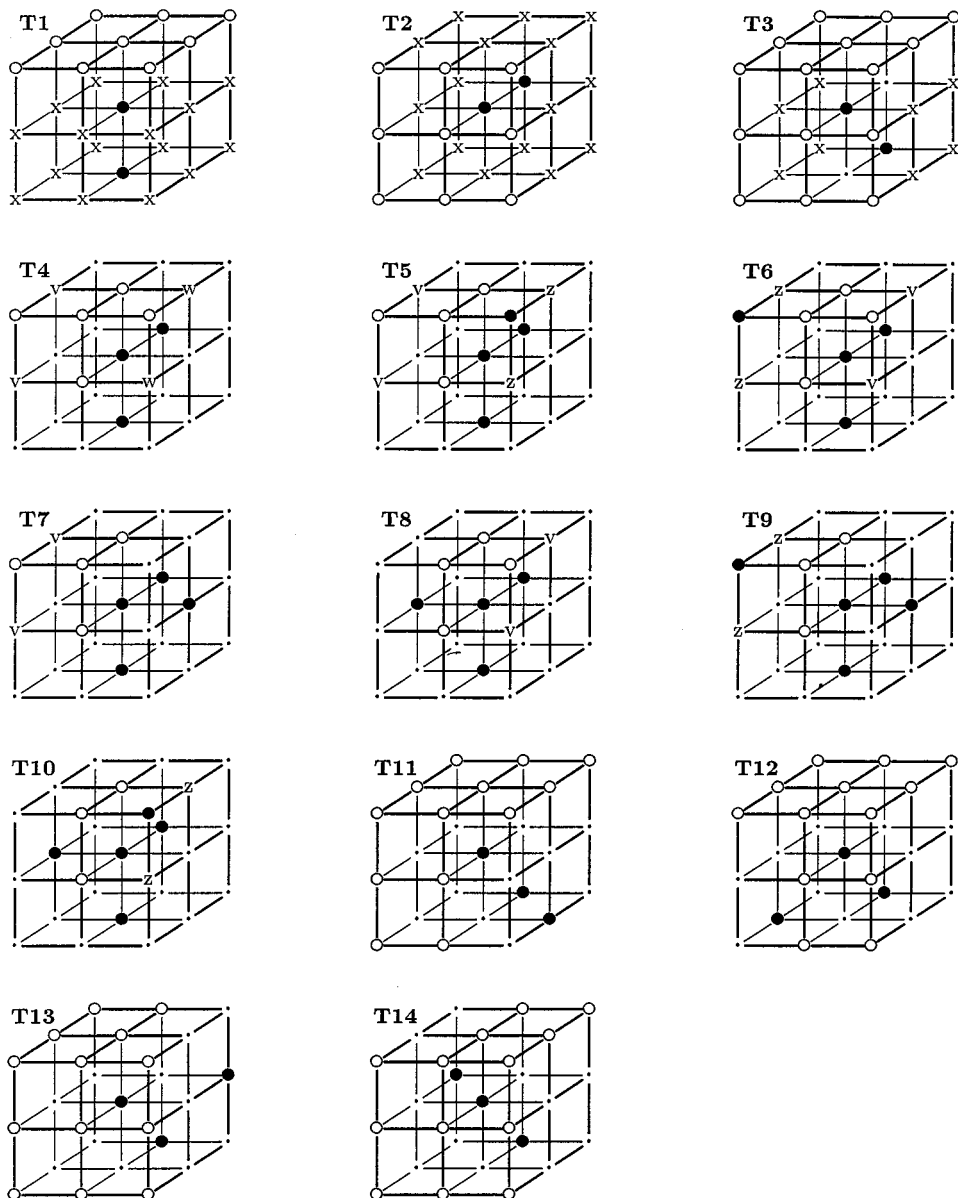


FIG. 3. The set of templates \mathcal{T}_{US} assigned to the first subiteration of the proposed curve thinning algorithm. Notations: every position marked “•” matches a black point; every position marked “o” matches a white point; every “.” (“don’t care”) matches either a black or a white point; at least one position marked “x” matches a black point; at least one position marked “v” matches a white point; at least one position marked “w” matches a white point; two positions marked “z” match different points (one of them matches a black point and the other one matches a white one).

Generally, it is not explained how the templates are designed. One may think that they are pulled out of a hat. Therefore, we try to show our motivation.

Thinning algorithms have to take care of the following four aspects:

1. forcing the “skeleton” to retain the topology of the original object (i.e., topology is to be preserved);

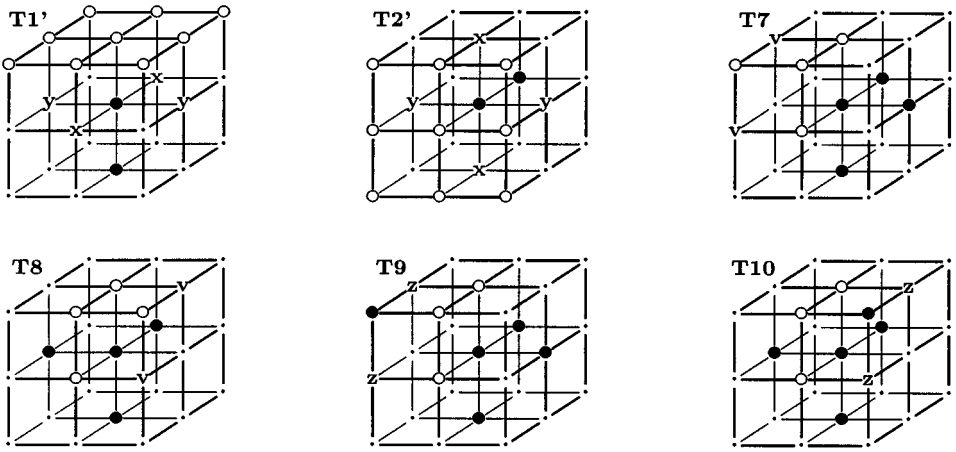


FIG. 4. The set of templates \mathcal{T}'_{US} derived from \mathcal{T}_{US} . It is assigned to the first subiteration of the proposed surface thinning algorithm. Notations: every position marked “•” matches a black point; every position marked “o” matches a white point; every “.” (“don’t care”) matches either a black or a white point; at least one position marked “x” matches a black point; at least one position marked “y” matches a black point; at least one position marked “v” matches a white point; two positions marked “z” match different points (one of them matches a black point and the other one matches a white one). Note that the missing templates delete surface end-points or they are covered by other templates.

2. providing “shape preservation” (i.e., significant features of the original object are to be produced);
3. forcing the “skeleton” to be in its geometrically correct position (i.e., in the “middle” of the object);
4. producing “maximal” thinning (i.e., the desired “width” of the “skeleton” is one point).

The first requirement (about the topological correctness) is proved in Section 5. It is based on the properties of the templates in \mathcal{T}_{US} (see Observations 5.4–5.7).

Shape preservation is a fairly important requirement, too. For example, an object like “b” cannot be thinned into an object like “o.” The aim of the thinning is not to produce the topological kernel [1] of an object; thinning differs from shrinking. That is the reason why end-point criteria are used in thinning. We propose the end-point characterizations given by Definitions 4.1–4.2. According to this requirement, the end points are not removed.

Geometrical correctness of the extracted skeleton is mostly achieved by the multidirectional thinning approach. An object is to be shrunk uniformly in each direction. Mention was made that the chosen order of deletion directions is not by accident in the proposed algorithm. In addition, suppose that a black point p is deletable by the set of templates \mathcal{T}_{US} . Let q be a black point so that the configuration around q corresponds to the reflected version of the neighborhood of p . Planar reflections are to be considered, where the two symmetry planes are illustrated in Fig. 5. Then point q is also deletable. Therefore, templates in \mathcal{T}_{US} show a kind of symmetry for providing medialness.

Maximal thinning and skeletonization without creating spurious branches are contradictory goals. It is rather difficult to prove that the requirement about maximal thinning is satisfied. It is worth testing an algorithm for lots of (natural) objects containing various segments. For example, Gerig *et al.* [6] demonstrated that the curve-thinning algorithm

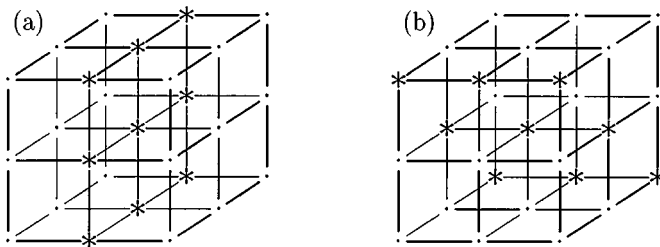


FIG. 5. The two symmetry planes for the set of templates \mathcal{T}_{US} . Points belonging to the vertical (a) and the slant (b) planes are marked “*.”

proposed by Tsao and Fu [26] does not provide a maximally thinned “skeleton.” However, templates **T11–T14** in \mathcal{T}_{US} can match relatively few configurations; they are fairly important in thinning some kinds of segments.

Thinning-based skeletonization concentrates on the topological correctness. Unfortunately, the distance induced by the investigated local neighborhood cannot be regarded as a good approximation to the Euclidean distance. Therefore, thinning algorithms are not invariant under rotation. The proposed algorithm is sensitive to object orientation, too. In addition, thinning algorithms which use subfield approach are not invariant under translation; they are sensitive to object position. It can be stated that there is no thinning algorithm capable of preserving the sharpness of any corners (if a local neighborhood of border points is investigated). Skeletonization methodology based on Euclidean distance mapping [4] can provide a geometrically correct result, but the topological correctness is not guaranteed. The two approaches (i.e., thinning and skeleton extraction from distance maps) can be combined to ensure both topologically and geometrically correct “skeleton.” Saito and Toriwaki [23] and Pudney [19] proposed 3D algorithms using both approaches.

Generally, there are two ways to define deletable points. One is to delete all simple points of a given type which are not end points. The second way is to select a subset of simple points, for example, by a set of templates. Deleting all simple points may create spurious skeleton branches. On the other hand, template matching may not provide maximal thinning. Template matching can be regarded as a trade-off between general and controlled deletion. Lee *et al.* [10] discussed different methods of characterization of 3D simple points in (26, 6) pictures; point p is simple in 25,985,144 cases of the $2^{26} = 67,108,864$ possible $3 \times 3 \times 3$ configurations. One might think that the algorithm deleting all the simple points of the considered type would be the best. However, the number of simple points recognized as deletable points does not mean the goodness of a border sequential thinning algorithm. For example, the surface thinning algorithm proposed by Gong and Bertrand [7] classify 2^{20} kinds of simple points as deletable points (see [10]). Despite of this fact, that algorithm can produce fairly nice “skeletons.”

It is easy to see that \mathcal{T}_{US} can delete approximately 2^{20} kinds of simple points. It can be seen that adding the template illustrated in Fig. 6 to the set of templates \mathcal{T}_{US} does not alter the topological correctness of the proposed algorithm, but it yields some unwanted curve/surface segments. Note that the same template is included in more sets of templates; for instance, template **T1** in \mathcal{T}_{US} is in the sets of templates assigned to deletion directions **UN**, **UE**, and **UW**, too. On the other hand, the same configuration can be matched by more than one template in \mathcal{T}_{US} . For instance, the configuration illustrated in Fig. 7 is matched by 8 of the 14 templates.

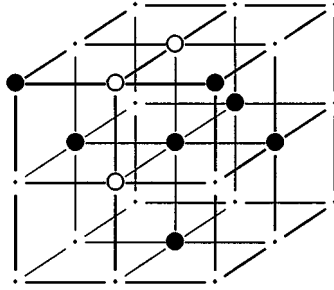


FIG. 6. Template that can delete some extremities of curves/surfaces. If the set of templates \mathcal{T}_{US} contains it, then unwanted curve/surface segments may be created.

4.2. Complexity and Boolean Implementation

The proposed algorithm goes around the object to be thinned according to the 12 deletion directions. One can say that the new 12-subiteration algorithm seems to be slower than a 6-subiteration solution. It is not necessarily true. Suppose that our sample object is a cube of size $16 \times 16 \times 16$. An iteration step of any 6-subiteration algorithm can generate a $14 \times 14 \times 14$ cube from the original one. A cube of size $8 \times 8 \times 8$ is created by the 12-subiteration algorithms. According to our experiments, less iteration steps are required by the proposed algorithm. Note that it is rather difficult to give any kind of estimation of the complexity, since the number of the required iteration steps is “data-dependent.”

The proposed curve-thinning algorithm directly extracts curve “skeleton.” Some authors (for example, Tsao and Fu [26]) proposed a 2-phase curve-thinning process. After the first phase, the surface “skeleton” is extracted from the original object. After the second phase that intermediate result is converted to a curve “skeleton.” In our opinion, the direct extraction of the curve “skeleton” is better than the 2-phase method. This is because there are more kinds of surface end-points than curve end-points, according to any end-point characterization. If the “noisy” boundary of an object contains some surface end-points that are not curve end-points, the surface “skeleton” extraction (the first phase of the 2-phase method) results in some parasitic surface segments that cannot be completely eliminated by the second phase. Note that the curve-thinning algorithm proposed by Saha *et al.* [22] does not use this 2-phase process. After a “primary thinning” step some two-point thick slanted surface segments may occur. A “primary skeleton” is not a “surface skeleton”; it can be regarded as an “almost curved skeleton.” A single iteration process

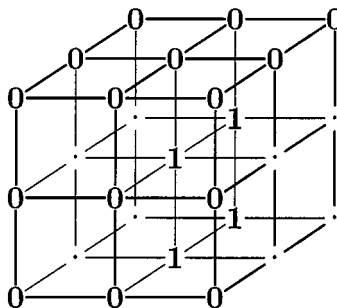


FIG. 7. Sample configuration that is matched by the following eight templates in \mathcal{T}_{US} : **T1–T4, T11–T14.**

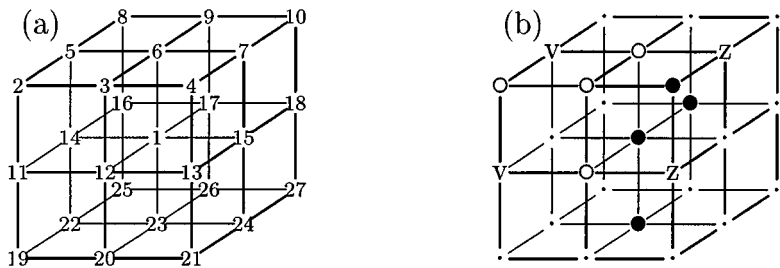


FIG. 8. The set of indices $\{1, 2, \dots, 27\}$ is assigned to the $3 \times 3 \times 3$ neighborhood of a point (a) and the template **T5** contained in \mathcal{T}_{US} (b). (Note that at least one position marked “v” matches a white point and the two positions marked “z” match different points.) The sample template can be described by $g_5(x_2, \dots, x_{27}) = \overline{x_2} \wedge \overline{x_3} \wedge \overline{x_6} \wedge \overline{x_{12}} \wedge x_4 \wedge x_{17} \wedge x_{23} \wedge (\overline{x_5} \wedge x_{11}) \wedge [(x_7 \wedge \overline{x_{13}}) \vee (\overline{x_7} \wedge x_{13})]$.

called the “final thinning” is applied that produces the “proper skeleton” from the “primary skeleton.”

The proposed algorithm makes easy implementation possible, since deletable points are given by $3 \times 3 \times 3$ templates. This dependence for the first subiteration can be given by a Boolean function of 27-variables $F : \{0, 1\}^{27} \rightarrow \{0, 1\}$. We refer to the individual variables by assigning indices to them as illustrated in Fig. 8a. Boolean function F can be expressed in the form

$$x'_1 = F(x_1, x_2, \dots, x_{27}) = x_1 \wedge \bigvee_{i=1}^{14} g_i(x_2, x_3, \dots, x_{27}),$$

where Boolean functions of 26-variables g_i ($i = 1, \dots, 14$) correspond to the templates in \mathcal{T}_{US} . For example, the conjunctive formula of g_5 assigned to the template **T5** is presented in Fig. 8. Since the templates belonging to other subiterations can be derived from rotations and/or reflections of templates in \mathcal{T}_{US} , Boolean functions assigned the derived templates can be given by the function $g_i(\Pi(x_2, \dots, x_{27}))$, where Π is the proper permutation of 26-variables.

It is easy to see that evaluation of the new value of a point takes approximately 300 elementary Boolean operations including logical or (“ \vee ”), logical and (“ \wedge ”), and logical not (“ $\overline{}$ ”). Optimization of the Boolean function evaluation could be achieved, for instance, with binary decision diagrams [20].

4.3. Results

Different shapes of objects have been tested by the proposed curve-thinning algorithm and the surface thinning algorithm. We have implemented some existing 3D thinning algorithms, too. Figures 9 and 10 are to compare the proposed method with others. The test objects are a “noise-free” solid torus of size $40 \times 40 \times 20$ and its noisy version, where noise is added to the boundary. It means that some border points are deleted from the original noise-free object and some white points 26-adjacent to a border point are changed to black. Note that we consider a model of noise which consists of deleting and adding simple points.

The following algorithms have been investigated:

1. **D12-PK-C**—the proposed directional 12-subiteration curve thinning algorithm.
2. **D12-PK-S**—the proposed directional 12-subiteration surface thinning algorithm.

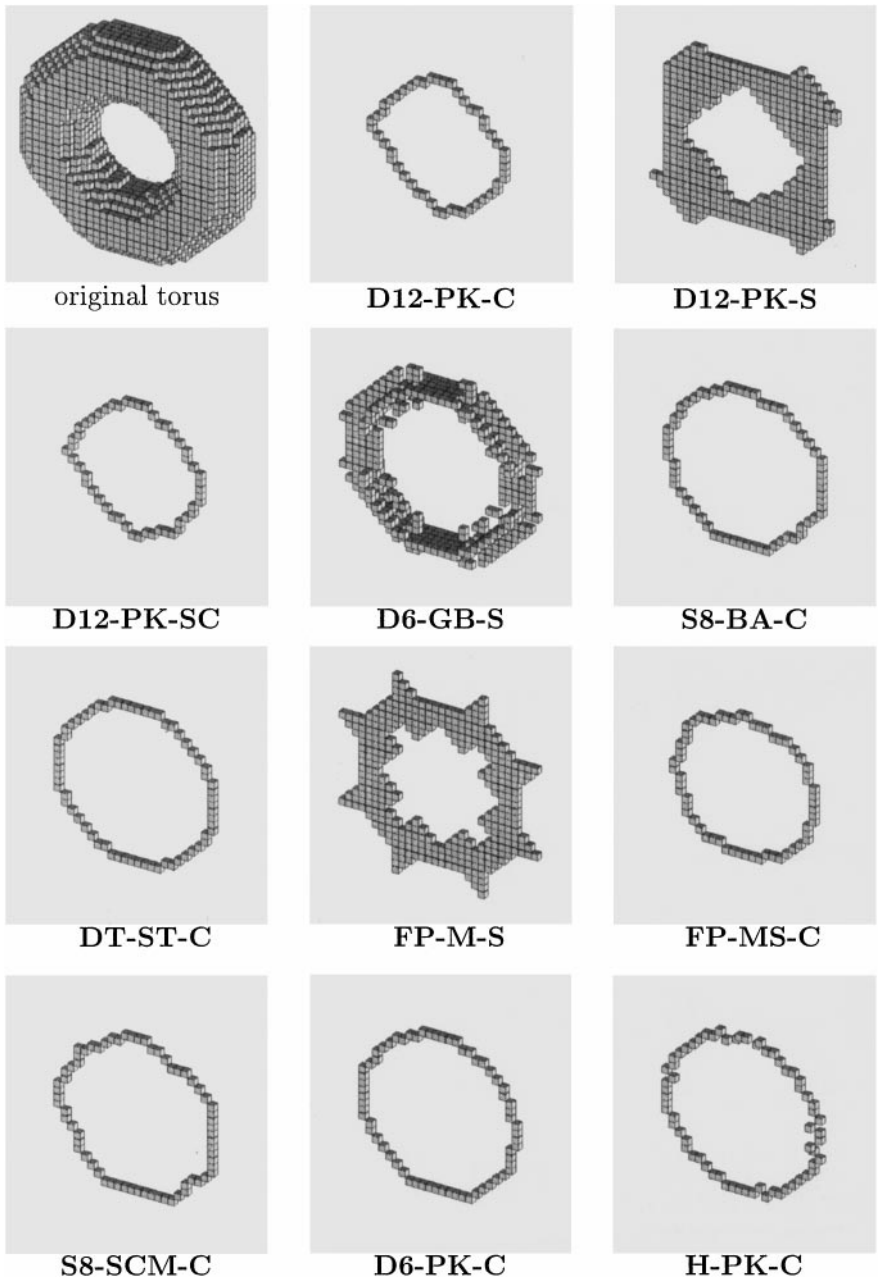


FIG. 9. Thinning of a solid torus: the original object and the “skeletons” produced by the 11 considered algorithms. (Each small cube represents a black point.)

3. **D12-PK-SC**—the 2-phase directional 12-subiteration curve thinning algorithm, where after the first phase, the surface “skeleton” is extracted by the algorithm **D12-PK-S** and after the second phase that intermediate result is converted to a curve “skeleton” by algorithm **D12-PK-C**.

4. **D6-GB-S**—directional six-subiteration surface thinning algorithm that was proposed by Gong and Bertrand [7].

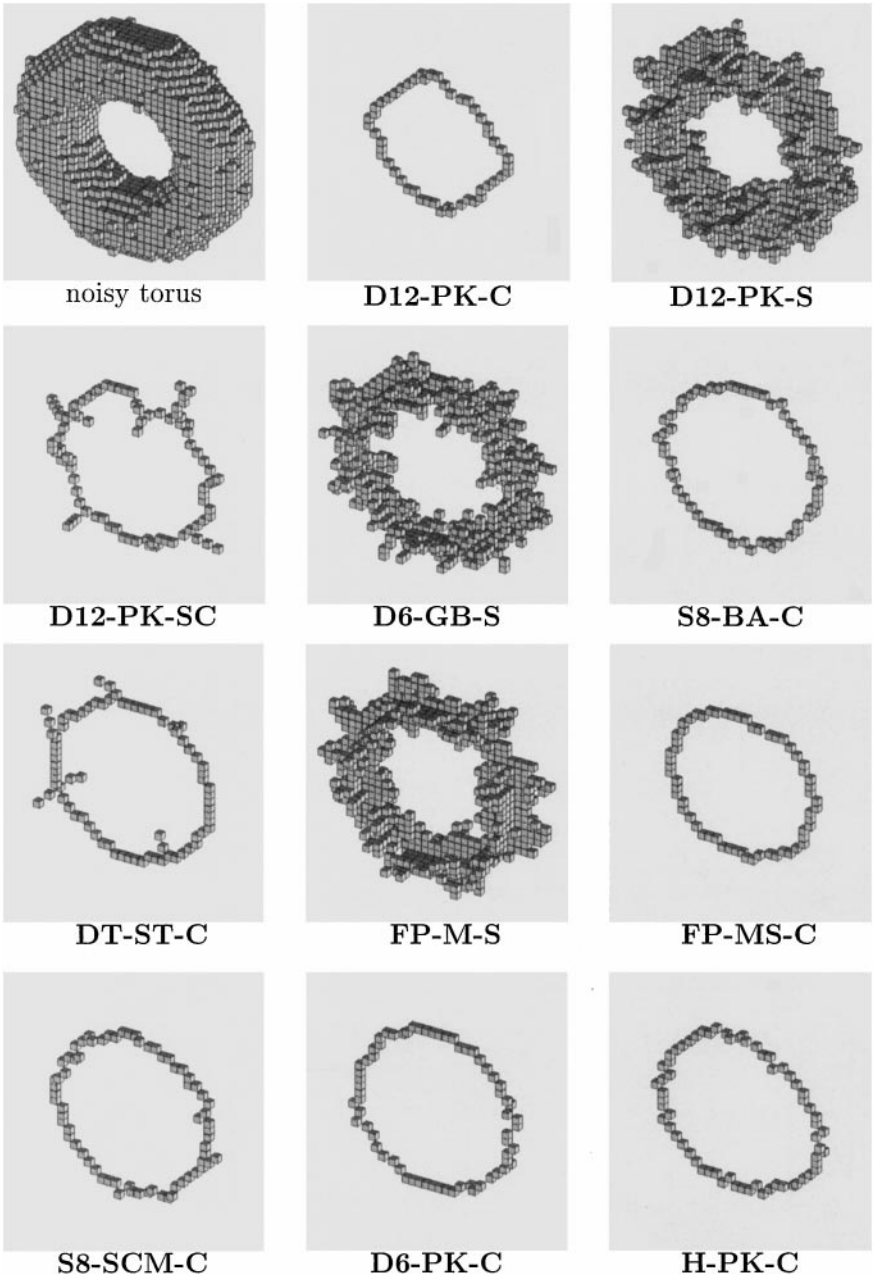


FIG. 10. Thinning of a noisy torus: the original object and the “skeletons” produced by the 11 considered algorithms.

5. **S8-BA-C**—subfield-sequential curve-thinning algorithm proposed by Bertrand and Aktouf [1].

6. **DT-ST-C**—sequential curve-thinning algorithm using a Euclidean distance transform proposed by Saito and Toriwaki [23]. (Note that it is the only sequential algorithm investigated here.)

7. **FP-M-S**—fully parallel surface-thinning algorithm proposed by Ma [12].
8. **FP-MS-C**—fully parallel curve-thinning algorithm proposed by Ma and Sonka [13].
9. **S8-SCM-C**—the parallel version (using eight subfields) of the curve-thinning algorithm proposed by Saha, Chaudhuri, and Dutta Majumder [22]. (Note that it is probable that the sequential version of this method can produce nicer “skeletons”.)
10. **D6-PK-C**—directional six-subiteration curve-thinning algorithm proposed by Palágyi and Kuba [17].
11. **H-PK-C**—hybrid curve-thinning algorithm proposed by Palágyi and Kuba [18].

We can state that surface-thinning algorithms (**D12-PK-S**, **D6-GB-S**, and **FP-M-S**) are more sensitive to boundary noise than the curve-thinning ones. It can be seen that algorithm **D12-PK-C** (direct extraction of the curve “skeleton”) produces nicer results than algorithm **D12-PK-SC** (the 2-phase method). The proposed curve-thinning algorithm does not create parasitic line segments.

Algorithms **D12-PK-C** and **D12-PK-S** were also tested for natural objects, too. An example for thinning a ventricle is shown in Fig. 11.

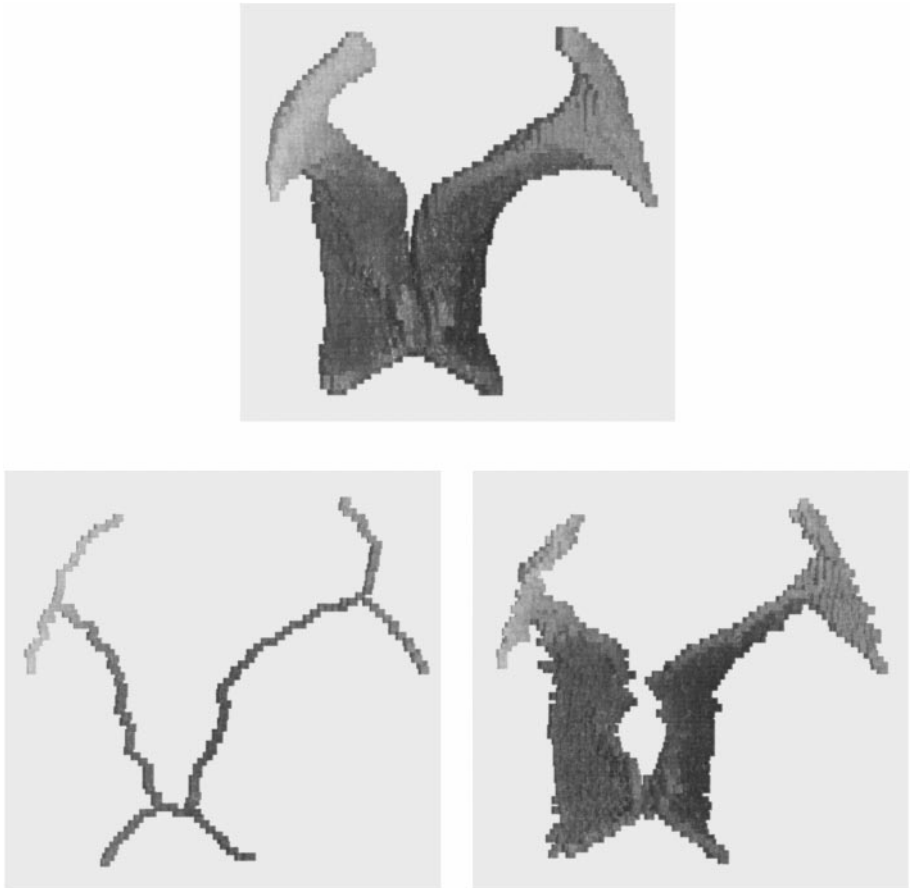


FIG. 11. Thinning of a ventricle extracted from a (greyscale) 3D MR brain study (top); the result of the proposed curve thinning algorithm **D12-PK-C** (bottom left); the result of the proposed surface thinning algorithm **D12-PK-S** (bottom right). These pictures are displayed by using the 3DVIEWNIX software system [27].

5. VERIFICATION

In this section, we show that both the curve-thinning algorithm and the surface-thinning algorithm are topology preserving. At first, some concepts of the digital topology and the applied results are presented.

The sequence of distinct points $\langle x_0, x_1, \dots, x_n \rangle$ is a j -path (for $j = 6, 18, 26$) of length n from point x_0 to point x_n in a nonempty 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 $1 \leq i \leq n$ (see Fig. 2). Note that a single point is a j -path of length 0. Two points are 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 3D binary (m, n) digital picture \mathcal{P} is a quadruple $\mathcal{P} = (\mathbb{Z}^3, m, n, B)$ [9]. Each element of \mathbb{Z}^3 is called a *point* of \mathcal{P} . Each point in $B \subseteq \mathbb{Z}^3$ is called a *black point* and value 1 is assigned to it. Each point in $\mathbb{Z}^3 \setminus B$ is called a *white point* and value 0 is assigned to it. Adjacency m belongs to the black points and adjacency n belongs to the white points. A *black component* is a maximal m -connected set of points in B . A *white component* is a maximal n -connected set of points in $\mathbb{Z}^3 \setminus B$.

We are dealing with $(26, 6)$ pictures. It is assumed that any picture contains finitely many black points.

A black point is called a *simple point* if its deletion does not alter the topology of the picture. We make use the following result for $(26, 6)$ pictures.

CRITERION 5.1 [2, 14, 21]. Black point p is simple in picture $(\mathbb{Z}^3, 26, 6, B)$ if and only if all three conditions hold:

1. The set $(B \setminus \{p\}) \cap N_{26}(p)$ contains exactly one 26-component.
2. The set $(\mathbb{Z}^3 \setminus B) \cap N_6(p)$ is not empty, and
3. it is six-connected in the set $(\mathbb{Z}^3 \setminus B) \cap N_{18}(p)$.

Parallel reduction operations delete a set of black points and not only a single simple point. We need to consider what is meant by topology preservation when a number of black points are deleted simultaneously. Ma [11] gave sufficient conditions for parallel reduction operations of 3D $(26, 6)$ pictures. Those require some additional concepts to be defined. Let \mathcal{P} be a picture. The set $D = \{d_1, \dots, d_k\}$ of black points is called a *simple set* of \mathcal{P} if D can be arranged in a sequence $\langle d_{i_1}, \dots, d_{i_k} \rangle$ in which each d_{i_j} is simple after $\{d_{i_1}, \dots, d_{i_{j-1}}\}$ is deleted from \mathcal{P} for $j = 1, \dots, k$. (By definition, let the empty set be simple.) A *unit lattice square* is a set of four corners of a unit square embedded in \mathbb{Z}^3 ; a *unit lattice cube* is a set of eight corners of a unit cube embedded in \mathbb{Z}^3 .

THEOREM 5.2 [11]. A 3D parallel reduction operation is topology preserving for $(26, 6)$ pictures if all of the following conditions hold:

1. Only simple points are deleted.
2. If two black corners, p and q , of a unit lattice square are deleted, then the set $\{p, q\}$ is simple.
3. If three black corners, p, q , and r , of a unit lattice square are deleted, then the set $\{p, q, r\}$ is simple.
4. If four black corners, p, q, r , and s , of a unit lattice square are deleted, then the set $\{p, q, r, s\}$ is simple.
5. No black component contained in a unit lattice cube can be deleted completely.

Instead of proving the conditions of Theorem 5.2, we use the more general theorem that provides *new sufficient conditions* for 3D parallel reduction operations to preserve topology.

THEOREM 5.3. *Let \mathcal{T} be a parallel reduction operation. Let p be any black point in any picture $\mathcal{P} = (\mathbb{Z}^3, 26, 6, B)$ so that p is deleted by \mathcal{T} . Let $Q \subseteq (N_{18}(p) \setminus \{p\}) \cap B$ be any set of black points in picture \mathcal{P} so that each point in Q is deleted by \mathcal{T} and $q_1 \in N_{18}(q_2)$ for any $q_1 \in Q$ and $q_2 \in Q$. Operation \mathcal{T} is topology preserving for $(26, 6)$ pictures if all of the conditions hold:*

1. p is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus Q)$.
2. No black component contained in a unit lattice cube can be deleted completely by operation \mathcal{T} .

Proof. Condition 2 of Theorem 5.3 corresponds to condition 5 of Theorem 5.2. We show that condition 1 of Theorem 5.3 implies conditions 1–4 of Theorem 5.2.

It is obvious that the set $N_{18}(p)$ contains any unit lattice squares in which p is a corner. Suppose that the deletable point p is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus Q)$ for any $Q \subseteq (N_{18}(p) \setminus \{p\}) \cap B$ containing deletable points:

(a) Suppose that $Q = \emptyset$. Since condition 1 of Theorem 5.3 holds, p is simple in the picture $(\mathbb{Z}^3, 26, 6, B)$. Therefore, condition 1 of Theorem 5.2 is satisfied.

(b) Let a and b be two distinct deletable points of a unit lattice square. If $p = b$ and $Q = \emptyset$ then b is simple by case (a). Suppose that $p = a$ and $Q = \{b\}$. Since condition 1 of Theorem 5.3 holds, a is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus \{b\})$. Consequently, the set $\{a, b\}$ is simple. Therefore, condition 2 of Theorem 5.2 is satisfied.

(c) Let a, b , and c be three distinct deletable points of a unit lattice square. In this case, b and c are two corners of a unit lattice square and set $\{b, c\}$ is simple by case (b). Suppose that $p = a$ and $Q = \{b, c\}$. Since condition 1 of Theorem 5.3 holds, a is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus \{b, c\})$. Consequently, the set $\{a, b, c\}$ is simple. Therefore, condition 3 of Theorem 5.2 is satisfied.

(d) Let a, b, c , and d be four distinct deletable points of a unit lattice square. In this case, b, c , and d are three corners of a unit lattice square and set $\{b, c, d\}$ is simple by case (c). Suppose that $p = a$ and $Q = \{b, c, d\}$. Since condition 1 of Theorem 5.3 holds, a is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus \{b, c, d\})$. Consequently, the set $\{a, b, c, d\}$ is simple. Therefore, condition 4 of Theorem 5.2 is satisfied. ■

In order to prove both conditions of Theorem 5.3, we classify the elements of templates and state some properties of the set of templates \mathcal{T}_{US} (see Fig. 3). A noncentral template element is called *black* if it is always black. A noncentral template element is called *white* if it is always white. Any other noncentral template element which is not white and not black, is called *potentially black*. A black or a potentially black noncentral template element is called *nonwhite*. A black point p is *deletable* if it can be deleted by at least one template in \mathcal{T}_{US} ; p is *nondeletable* otherwise.

Observation 5.4. Let us examine the configuration of $3 \times 3 \times 3$ points illustrated in Fig. 12. If black point p is deletable, then all the conditions hold:

1. Point q is white.
2. At least one point in $\{r, s\}$ is white.
3. At least one point in $\{t, u, v\}$ is black.

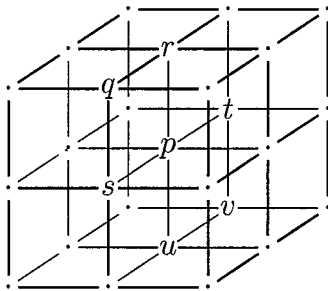


FIG. 12. Configuration assigned to Observations 5.4 and 5.6.

Observation 5.5. Let us examine both configurations illustrated in Fig. 13. Let points p and q be black. If point q is deletable, then all the conditions hold:

1. Point r is black.
2. All points marked s are white.

Observation 5.6. Let us examine the configuration of $3 \times 3 \times 3$ points illustrated in Fig. 12. If point q is black, then point p is nondeletable.

Observation 5.7. Let us examine both configurations illustrated in Fig. 14. If p is deletable and q or r is black, then point s is white.

The following four lemmas are useful in proving that the parallel reduction operation given by the set of templates \mathcal{T}_{US} is topology preserving.

LEMMA 5.8. *Each template in \mathcal{T}_{US} deletes only simple points of (26, 6) pictures.*

Proof. The first point is to verify that there exists a 26-path between any two nonwhite positions (condition 1 of Criterion 5.1). It is sufficient to show that any potentially black position is 26-adjacent to a black position and any black position is 26-adjacent to another black position (if the template contains at least two black positions). It is obvious by careful examination of the fourteen templates in \mathcal{T}_{US} .

Let us examine conditions 2 and 3 of Criterion 5.1. To prove it, it is sufficient to show for each template in \mathcal{T}_{US} :

1. that there exists a white position six-adjacent to the central position,
2. that for any two white positions six-adjacent to the central position p are 6-connected in the set of white positions 18-adjacent to p ,
3. and that for any potentially black position six-adjacent to the central position p , there exists a six-adjacent white 18-neighbor which is six-adjacent to a white position six-adjacent to p .

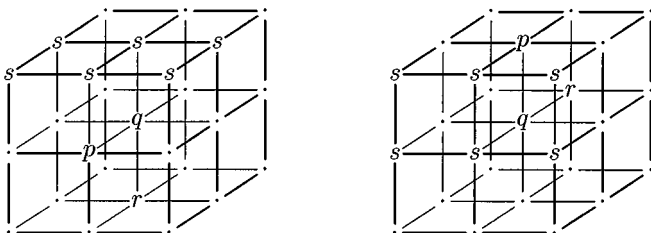


FIG. 13. Configurations assigned to Observation 5.5.

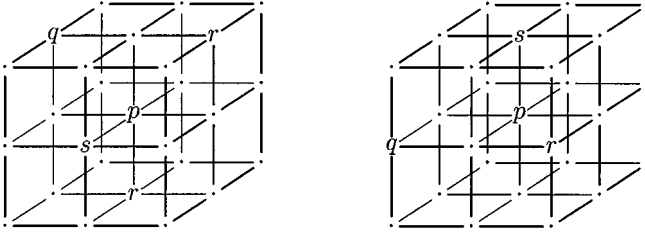


FIG. 14. Configurations assigned to Observation 5.7.

The three points are obvious by a careful examination of the templates in \mathcal{T}_{US} . ■

LEMMA 5.9. *Let us examine the $3 \times 3 \times 3$ configurations illustrated in Fig. 15. Point p is simple in each of them.*

Proof. It is to be shown that Criterion 5.1 is satisfied. It can be seen as we did it in the case of Lemma 5.8. ■

LEMMA 5.10. *Let p be any black point in any picture $\mathcal{P} = (\mathbb{Z}^3, 26, 6, B)$ so that p is deletable. Let $Q \subseteq (N_{18}(p) \setminus \{p\}) \cap B$ be any set of black points in picture \mathcal{P} so that each point in Q is deletable and $q_1 \in N_{18}(q_2)$, for any $q_1 \in Q$ and $q_2 \in Q$. Then p is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus Q)$.*

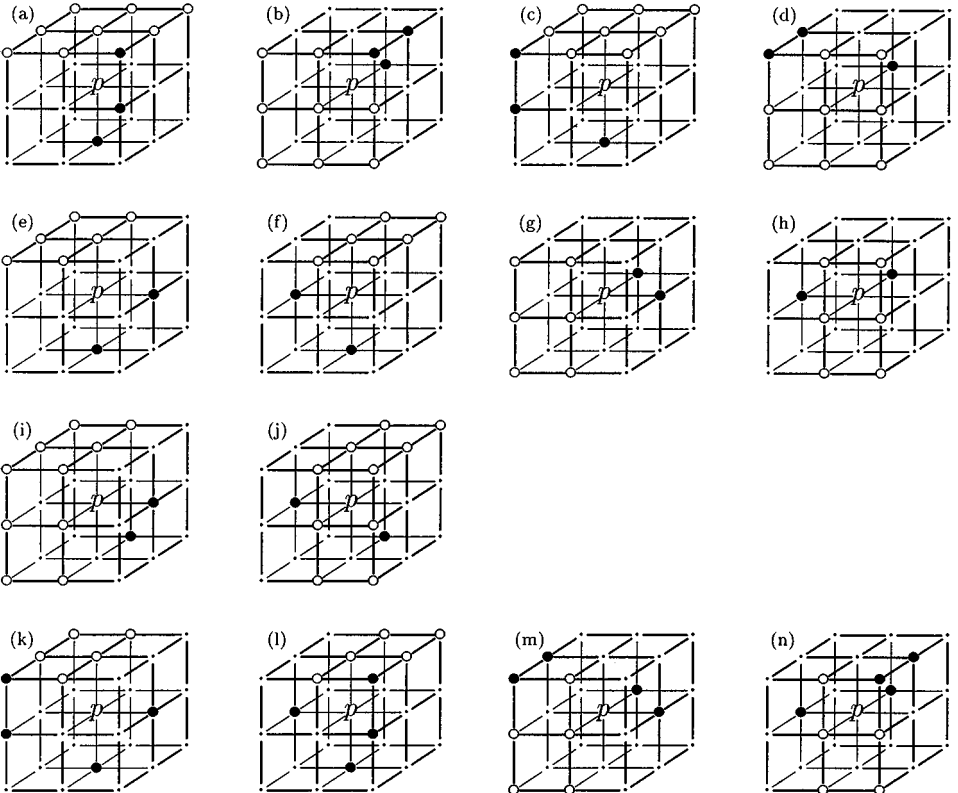


FIG. 15. Configurations of Lemma 5.9. Point p is simple in each of them. Note that each point marked “•” is black, each point marked “○” is white, and each point marked “◐” may be either black or white.

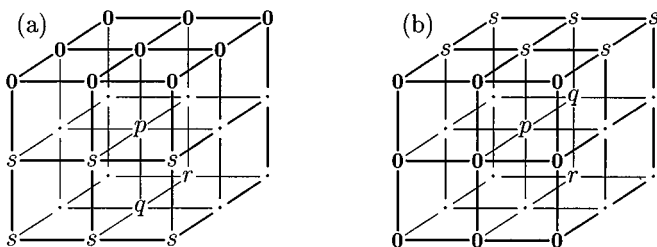


FIG. 16. Configurations if point p can be deleted by template **T1** (a) or template **T2** (b).

Proof. We know that each deletable point p is simple by Lemma 5.8. It can be stated that the value of any point coinciding with a potentially black template position does not alter the simplicity of p . Therefore, it is sufficient to deal with deletable points that coincide with black template positions. For brevity, each template position marked “ \mathbf{x} ” (see Fig. 3) is regarded as “ \cdot ” (don’t care). It is easy to see that the modified templates (**T1**, **T2**, and **T3**) delete only simple points, too.

The following points are to be investigated:

1. Point p can be deleted by template **T1** or **T2**. Let us examine both configurations of $3 \times 3 \times 3$ points illustrated in Fig. 16. Point q coincides with the only black position of the considered templates. If $q \in Q$ then each point marked s is white and point r is black by Observation 5.5. Then point p can be deleted by template **T3**. Therefore, p is simple after the deletion of Q by Lemma 5.8. (Since Q contains deletable points, $r \notin Q$ by Observation 5.6.)

2. Point p can be deleted by template **T3**. Let q be the black point so that coincides with the only black position of **T3**. Since Q contains deletable points, $q \notin Q$ by Observation 5.6.

3. Point p can be deleted by template **T4**. Let us examine the configuration of $3 \times 3 \times 3$ points illustrated in Fig. 17. Since the template contains two black positions (coinciding with q_1 and q_2), the following three cases are to be distinguished:

(a) $q_1 \in Q$, $q_2 \notin Q$. Each point marked s is white by Observation 5.5. Then point p can be deleted by template **T1**. Therefore, p is simple after the deletion of Q by Lemma 5.8.

(b) $q_1 \notin Q$, $q_2 \in Q$. Each point marked t is white by Observation 5.5. Then point p can be deleted by template **T2**. Therefore, p is simple after the deletion of Q by Lemma 5.8.

(c) $q_1 \in Q$, $q_2 \in Q$. All points marked s are white, all points marked t are white, and point r is black by Observation 5.5. Then point p can be deleted by template **T3**. Therefore, p is simple after the deletion of Q by Lemma 5.8. (Since Q contains deletable points, $r \notin Q$ by Observation 5.6.)

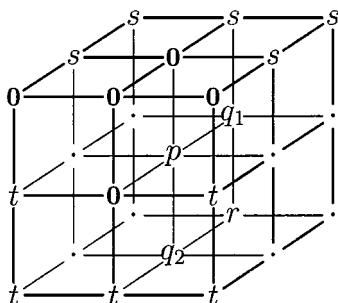


FIG. 17. Configuration if point p can be deleted by template **T4**.

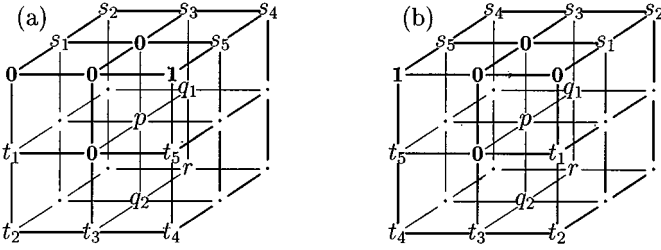


FIG. 18. Configurations if point p can be deleted by template **T5** (a) or template **T6** (b).

4. Point p can be deleted by template **T5** or **T6**. Let us examine the configurations of $3 \times 3 \times 3$ points illustrated in Fig. 18. These templates contain two black positions (coinciding with q_1 and q_2) and a third one that is not 18-adjacent to p . The following three cases are to be distinguished:

- (a) $q_1 \in Q, q_2 \notin Q$. Points $\{s_1, \dots, s_5\}$ are white by Observation 5.5. Point t_5 is black, since points marked “z” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (a) and (c) in Fig. 15).
- (b) $q_1 \notin Q, q_2 \in Q$. Points $\{t_1, \dots, t_5\}$ are white by Observation 5.5. Point s_5 is black, since points marked “z” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (b) and (d) in Fig. 15).
- (c) $q_1 \in Q, q_2 \in Q$. Both points s_5 and t_5 are white by Observation 5.5. It is a contradiction, since points marked “z” are different in these templates.

5. Point p can be deleted by template **T7** or **T8**. Let us examine the configurations of $3 \times 3 \times 3$ points illustrated in Fig. 19. Since these templates contain three black positions (coinciding with q_1, q_2 , and q_3), the following six cases are to be distinguished:

- (a) $q_1 \in Q, q_2 \notin Q, q_3 \notin Q$. Points $\{s_1, \dots, s_5\}$ are white by Observation 5.5. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (e) and (f) in Fig. 15).
- (b) $q_1 \notin Q, q_2 \in Q, q_3 \notin Q$. Points $\{t_1, \dots, t_5\}$ are white by Observation 5.5. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (g) and (h) in Fig. 15).
- (c) $q_1 \notin Q, q_2 \notin Q, q_3 \in Q$. Point u is white and at least one point in $\{s_5, t_5\}$ is white by Observation 5.4. Then point p can be deleted by template **T4**. Therefore, p is simple after the deletion of Q by Lemma 5.8.
- (d) $q_1 \in Q, q_2 \in Q, q_3 \notin Q$. Points $\{s_1, \dots, s_5, t_1, \dots, t_5\}$ are white and point r is black by Observation 5.5. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (i) and (j) in Fig. 15).

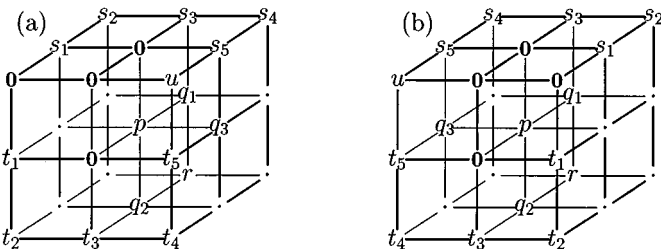


FIG. 19. Configurations if point p can be deleted by template **T7** (a) or template **T8** (b).

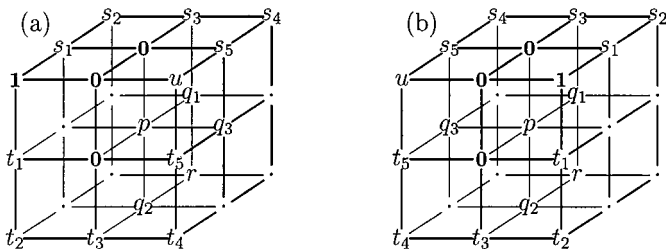


FIG. 20. Configurations if point p can be deleted by template **T9** (a) or template **T10** (b).

(e) $q_1 \in Q, q_2 \notin Q, q_3 \in Q$. Points $\{s_1, \dots, s_5\}$ are white by Observation 5.5. Point u is white is white by Observation 5.4. Then point p can be deleted by template **T1**. Therefore, p is simple after the deletion of Q by Lemma 5.8.

(f) $q_1 \notin Q, q_2 \in Q, q_3 \in Q$. Points $\{t_1, \dots, t_5\}$ are white by Observation 5.5. Point u is white is white by Observation 5.4. Then point p can be deleted by template **T2**. Therefore, p is simple after the deletion of Q by Lemma 5.8.

6. Point p can be deleted by template **T9** or **T10**. Let us examine the configurations of $3 \times 3 \times 3$ points illustrated in Fig. 20. These templates contain three black positions (coinciding with q_1, q_2 , and q_3) and a fourth one that is not 18-adjacent to p . The following six cases are to be distinguished:

(a) $q_1 \in Q, q_2 \notin Q, q_3 \notin Q$. Points $\{s_1, \dots, s_5\}$ are white by Observation 5.5. Point t_1 is black, since points marked “ z ” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (k) and (l) in Fig. 15).

(b) $q_1 \notin Q, q_2 \in Q, q_3 \notin Q$. Points $\{t_1, \dots, t_5\}$ are white by Observation 5.5. Point s_1 is black, since points marked “ z ” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (m) and (n) in Fig. 15).

(c) $q_1 \notin Q, q_2 \notin Q, q_3 \in Q$. Point u is white and at least one point in $\{s_5, t_5\}$ is white by Observation 5.4. Then point p can be deleted by template **T5** or **T6**. Therefore, p is simple after the deletion of Q by Lemma 5.8.

(d) $q_1 \in Q, q_2 \in Q, q_3 \notin Q$. Both points s_1 and t_1 are white by Observation 5.5. It is a contradiction, since points marked “ z ” are different in these templates.

(e) $q_1 \in Q, q_2 \notin Q, q_3 \in Q$. Points $\{s_1, \dots, s_5\}$ are white by Observation 5.5. Point u is white is white by Observation 5.4. Point t_1 is black, since points marked “ z ” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (a) and (c) in Fig. 15).

(f) $q_1 \notin Q, q_2 \in Q, q_3 \in Q$. Points $\{t_1, \dots, t_5\}$ are white by Observation 5.5. Point u is white is white by Observation 5.4. Point s_1 is black, since points marked “ z ” are different in these templates. Then point p is simple after the deletion of Q by Lemma 5.9 (see configurations (b) and (d) in Fig. 15).

7. Point p can be deleted by template **T11**, **T12**, **T13**, or **T14**. Let us examine the configurations of $3 \times 3 \times 3$ points illustrated in Fig. 21. There are two points (q_1 and q_2) coinciding with black positions. We can state that $q_1 \notin Q$ by Observation 5.6 (since each point Q is deletable). If $q_2 \in Q$, then point s is white by condition 1 of Observation 5.4 and point t is white by Observation 5.7. Then point p can be deleted by template **T3**, therefore, p is simple after the deletion of Q by Lemma 5.8. ■

LEMMA 5.11. *Let C be an object in picture $\mathcal{P} = (\mathbb{Z}^3, 26, 6, B)$. If object C is contained in a unit lattice cube, then C cannot be deleted completely by the set of templates \mathcal{T}_{U_S} .*

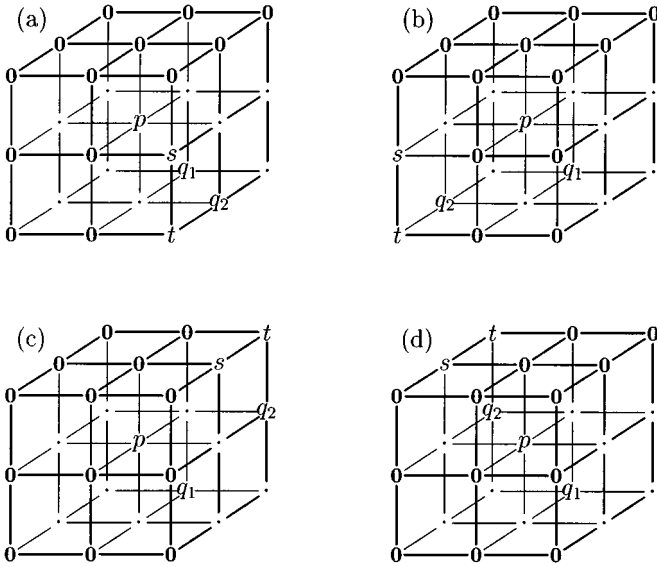


FIG. 21. Configurations if point p can be deleted by T11 (a), T12 (b), T13 (c), or T14 (d).

Proof. Let us examine the unit lattice cube as it is illustrated in Fig. 22. The eight corners are subdivided into two disjoint sets of points $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$. Two cases are to be distinguished:

- (a) $X \cap C \neq \emptyset$;
- (b) $X \cap C = \emptyset$.

Let us examine case (a). If point $x_4 \in C$ (i.e., x_4 is black), then it is nondeletable since condition 2 of Observation 5.4 is not satisfied. If point $x_4 \notin C$ (i.e., x_4 is white), then $\{x_1, x_2, x_3\} \cap C \neq \emptyset$. If point $x_3 \in C$ (and x_4 is white), then it is nondeletable, since condition 2 of Observation 5.4 is not satisfied. If point $x_2 \in C$ (and x_4 is white), then it is nondeletable, since condition 2 of Observation 5.4 is not satisfied, too. Finally, if $\{x_2, x_3, x_4\} \cap C = \emptyset$ (i.e., they are white), then point $x_1 \in C$ (since $X \cap C \neq \emptyset$). In this case, x_1 is nondeletable since condition 2 of Observation 5.4 is not satisfied. It has been shown that C contains at least one nondeletable point. Therefore, C cannot be deleted completely in case (a).

As far as (b) is concerned, $Y \cap C \neq \emptyset$, since object C contains at least one black point. It can be carried out by analogy to the case (a). ■

We are ready to prove the main theorem.

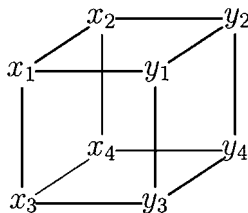


FIG. 22. Unit lattice cube containing the sets of points $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$.

THEOREM 5.12. *Both the curve thinning algorithm and the surface thinning algorithm are topology preserving.*

Proof. The parallel reduction operation given by the set of templates \mathcal{T}_{US} fulfills both conditions of Theorem 5.3 by Lemma 5.10 and Lemma 5.11. Consequently, the first subiteration of the curve thinning algorithm is topology preserving. It can be proved for the other 11 subiterations in a similar way. The entire curve thinning algorithm is topology preserving, since it is composed of topology preserving operations.

It is easy to see that the parallel reduction operation given by the set of templates \mathcal{T}'_{US} deletes less kinds of simple points than the operation given by the set of templates \mathcal{T}_{US} . Consequently, the first subiteration of the surface-thinning algorithm is topology preserving. It can be seen for the other 11 subiterations of the surface-thinning algorithm; therefore, it is topology preserving. ■

ACKNOWLEDGMENTS

This work was supported by OTKA T023804 and FKFP 0908/1997 Grants. The authors thank G. T. Herman, Y. K. Udupa, and D. Odhner (Medical Image Processing Group, University of Pennsylvania, Philadelphia) for their valuable suggestions about this work. We are grateful to G. Székely (Image Science Group, Communication Technology Laboratory, ETH, Zürich) for supplying the image data for Fig. 11.

REFERENCES

1. G. Bertrand and Z. Aktouf, A 3D thinning algorithms using subfields, in *Proceedings, SPIE Conference on Vision Geometry III, Vol. 2356, 1994*, pp. 113–124.
2. G. Bertrand, A parallel thinning algorithm for medial surfaces, *Pattern Recognit. Lett.* **16**, 1995, 979–986.
3. H. Blum, A transformation for extracting new descriptors of shape, *Models for the Perception of Speech and Visual Form*, pp. 362–380, MIT Press. Cambridge, MA, 1967.
4. G. Borgefors, Distance transformations in arbitrary dimensions, *Comput. Vision Graphics Image Process.* **27**, 1984, 321–345.
5. L. Calabi, *A Study of the Skeleton of Plane Figures*, Technical report 60429, Parke Mathematical Laboratories, 1965.
6. G. Gerig, Th. Koller, G. Székely, Ch. Brechbühler, and O. Kübler, Symbolic description of 3-D structures applied to cerebral vessel tree obtained from MR angiography volume data, in *Proceedings of 13th International Conference Information Processing in Medical Imaging, IPMI'93, Lecture Notes in Computer Science, Vol. 687*, pp. 94–111, Springer-Verlag, New York/Berlin, 1993.
7. W. X. Gong and G. Bertrand, A simple parallel 3D thinning algorithm, in *Proceedings 10th IEEE International Conference on Pattern Recognition, 1990*, pp. 188–190.
8. T. Y. Kong, On topology preservation in 2-d and 3-d thinning, *Internat. J. Pattern Recognit. Artif. Intell.* **9**, 1995, 813–844.
9. T. Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, *Comput. Vision Graphics Image Process.* **48**, 1989, 357–393.
10. T. Lee, R. L. Kashyap, and C. Chu, Building skeleton models via 3-D medial surface/axis thinning algorithms, *CVGIP: Graph. Models Image Process.* **56**, 1994, 462–478.
11. C. M. Ma, On topology preservation in 3D thinning, *CVGIP: Image Understand.* **59**, 1994, 328–339.
12. C. M. Ma, A 3D fully parallel thinning algorithm for generating medial faces, *Pattern Recognit. Lett.* **16**, 1995, 83–87.
13. C. M. Ma and M. Sonka, A fully parallel 3D thinning algorithm and its applications, *Comput. Vision Image Understand.* **64**, 1996, 420–433.

14. G. Malandain, G. Bertrand, Fast characterization of 3D simple points, in *Proceedings 11th IEEE International Conference on Pattern Recognition, 1992*, pp. 232–235.
15. D. G. Morgenthaler, *Three-Dimensional Simple Points: Serial Erosion, Parallel Thinning and Skeletonization*, TR-1005, Computer Vision Laboratory, Computer Science Center, Univ. of Maryland, College Park, MD, 1981.
16. J. Mukherjee, P. P. Das, and B. N. Chatterjee, On connectivity issues of ESPTA, *Pattern Recognit. Lett.* **11**, 1990, 643–648.
17. K. Palágyi and A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, *Pattern Recognit. Lett.* **19**, 1998, 613–627.
18. K. Palágyi and A. Kuba, A hybrid thinning algorithm for 3D medical images, *J. Comput. Inform. Technol. CIT* **6**, 1998, 149–164.
19. C. Pudney, Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images, *Comput. Vision Image Understand.* **72**, 1998, 404–413.
20. L. Robert and G. Malandain, Fast binary image processing using binary decision diagrams, *Comput. Vision Image Understand.* **72**, 1998, 1–9.
21. P. K. Saha and B. B. Chaudhuri, Detection of 3-D simple points for topology preserving transformations with application to thinning, *IEEE Trans. Pattern Anal. Mach. Intell.* **16**, 1994, 1028–1032.
22. P. K. Saha, B. B. Chaudhuri, and D. D. Majumder, A new shape-preserving parallel thinning algorithm for 3D digital images, *Pattern Recognit.* **30**, 1997, 1939–1955.
23. T. Saito and J. Toriwaki, A sequential thinning algorithm for three-dimensional digital pictures using the Euclidean distance transformation, in *Proceedings of the 9th Scandinavian Conference on Image Analysis (SCIA'95), 1995*, pp. 507–516.
24. S. N. Srihari, J. K. Udupa, M. M. Yau, Understanding the bin of parts, in *Proceedings IEEE Conference on Decision Control, 1979*, 44–49.
25. G. Székely, *Shape characterization by Local Symmetries*, Habilitationsschrift, ETH Zürich, 1996.
26. Y. F. Tsao and K. S. Fu, A parallel thinning algorithm for 3-D pictures, *Comput. Graphics Image Process.* **17**, 1981, 315–331.
27. J. K. Udupa, D. Odhner, S. Samarasekera, R. Goncalves, K. Iyer, K. Venugopal, and S. Furuie, 3DVIEWNIX: An open, transportable, multidimensional, multimodality, multiparametric imaging software system, in *SPIE Proceedings, Vol. 2164*, pp. 58–73. Int. Soc. Opt. Eng., Bellingham, WA, 1994.