Chapter 6
Topology Preserving Parallel 3D Thinning Algorithms

Kálmán Palágyi, Gábor Németh, and Péter Kardos

Abstract A widely used technique to obtain skeletons of binary objects is thinning, which is an iterative layer-by-layer erosion in a topology preserving way. Thinning in 3D is capable of extracting various skeleton-like shape descriptors (i.e., centerlines, medial surfaces, and topological kernels). This chapter describes a family of new parallel 3D thinning algorithms for (26, 6) binary pictures. The reported algorithms are derived from some sufficient conditions for topology preserving parallel reduction operations, hence their topological correctness is guaranteed.

6.1 Introduction

Skeleton is a region-based shape descriptor which represents the general shape of objects. 3D skeleton-like shape features (i.e., centerlines, medial surfaces, and topological kernels) play important role in various applications in image processing, pattern recognition, and visualization [6, 10, 38, 41, 42, 44].

An illustrative definition of the skeleton uses 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 [5]. Thinning is a digital simulation of the fire front propagation: the border points that satisfy certain topological and geometric constraints are deleted in iteration steps [12].

A *3D binary picture* [11, 12] is a mapping that assigns a value of 0 or 1 to each point with integer coordinates in the 3D digital space $\mathbb{Z}^3$. Points having the value of 1 are called *black* points, and those with a zero value are called *white* ones. Black points form the components of a picture, while white points form the background and the cavities. We consider (26, 6)-pictures, where 26-adjacency and 6-adjacency are, respectively, used for the components and their complement [12].

K. Palágyi (✉) · G. Németh · P. Kardos
Institute of Informatics, University of Szeged, Szeged, Hungary
e-mail: palagyi@inf.u-szeged.hu

G. Németh
e-mail: gnemeth@inf.u-szeged.hu

P. Kardos
e-mail: pkardos@inf.u-szeged.hu

V.E. Brimkov, R.P. Barneva (eds.), *Digital Geometry Algorithms*,
Lecture Notes in Computational Vision and Biomechanics 2,
DOI 10.1007/978-94-007-4174-4_6,

A *reduction operation* transforms a binary picture only by changing some black points to white ones (which is referred to as the *deletion* of 1's). A *parallel reduction operation* deletes all points satisfying its condition simultaneously. A reduction operation does *not* preserve topology [11] if

- any component in the input picture is split (into several components) or is 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 *hole (or tunnel)* in 3D pictures. A hole (which doughnuts have) is formed of 0's, but it is not a cavity [12]. Topology preservation implies that eliminating or creating any hole is not allowed.

There are three types of 3D thinning algorithms for producing the three types of skeleton-like shape features: *curve-thinning* algorithms are used to extract medial lines or centerlines, *surface-thinning* algorithms produce medial surfaces, while *kernel-thinning* algorithms are capable of extracting topological kernels. A topological kernel is a minimal set of points that is topologically equivalent [12] to the original object (i.e., if we remove any further point from it, then the topology is not preserved). Note that kernel-thinning algorithms are often referred to as reductive shrinking algorithms [9]. 3D curve-thinning and surface-thinning algorithms use operations that delete some points which are not *endpoints*, since preserving endpoints provides important geometrical information relative to the shape of the objects. Kernel-thinning algorithms for extracting topological kernels do not take any endpoint characterization into consideration. Medial surfaces are usually extracted from general shapes, tubular structures can be represented by their centerlines, and extracting topological kernels is useful in topological description.

Most of the existing thinning algorithms are parallel as the fire front propagation is by nature parallel. These algorithms are composed of parallel reduction operations. Parallel reduction operations delete a set of points simultaneously which may lead to altering the topology. Note that deletion rules of parallel thinning algorithms are generally given by matching templates. In order to verify that a given parallel 3D thinning algorithm preserves the topology for all possible $(26, 6)$ pictures, some sufficient conditions for topology preservation have been proposed [11, 18, 36]. However, verifying these conditions usually means checking several configurations of points, hence papers presenting thinning algorithms contain long proof parts. Despite of complex proofs, it was claimed in [14, 45] that two parallel 3D thinning algorithms [18, 19] are not topology preserving. That is why we propose a safe technique for designing topologically correct parallel 3D thinning algorithms. Our approach is based on some new sufficient conditions for topology preservation. These conditions consider individual points (instead of point configurations) and can be combined with various thinning strategies.

In this chapter we present 15 algorithms that are derived from the new sufficient conditions combined with the three major strategies for parallel thinning (i.e., fully parallel, subiteration-based, and subfield-based [8]) and three types of endpoints.

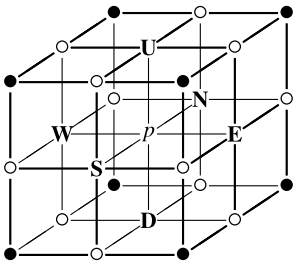


Fig. 6.1 Frequently used adjacencies in $\mathbb{Z}^3$. The set $N_6(p)$ contains point p and the six points marked **U**, **D**, **N**, **E**, **S**, and **W**. The set $N_{18}(p)$ contains $N_6(p)$ and the twelve points marked by "○". The set $N_{26}(p)$ contains $N_{18}(p)$ and the eight points marked by "●"

The rest of this chapter is organized as follows. Section 6.2 reviews the basic notions and results of 3D digital topology, and we present our sufficient conditions for topology preservation. Then, in Sect. 6.3 we propose 15 parallel 3D thinning algorithms and their topological correctness is proved. Since fast extraction of skeleton-like shape features is extremely important in numerous applications for large 3D shapes, Sect. 6.4 is devoted to the efficient implementation of the proposed algorithms, and Sect. 6.5 presents some illustrative results. In Sect. 6.6 some possible future works and open problems are outlined. Finally, we round off the chapter with some concluding remarks.

6.2 Topology Preserving Parallel Reduction Operations

In this section, we present new sufficient conditions for topology preservation. First we outline some concepts of digital topology and related key results that will be used in the sequel.

Let p be a point in the 3D digital space $\mathbb{Z}^3$. Let us denote $N_j(p)$ (for $j = 6, 18, 26$) the set of points that are *j-adjacent* to point p (see Fig. 6.1).

The sequence of distinct points $\langle x_0, x_1, \ldots, x_n \rangle$ is called a *j-path* (for $j = 6, 26$) of length n from point x_0 to point x_n in a non-empty set of points X if each point of the sequence is in X and x_i is j-adjacent to x_{i-1} for each $1 \leq i \leq n$ (see Fig. 6.1). Note that a single point is a j-path of length 0. Two points are said to be *j-connected* in the set X if there is a j-path in X between them ($j = 6, 26$). 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 ($j = 6, 26$).

A *3D binary* $(26, 6)$ *digital picture* $\mathscr{P}$ is a quadruple $\mathscr{P} = (\mathbb{Z}^3, 26, 6, B)$ [12]. Each element of $\mathbb{Z}^3$ is called a *point* of $\mathscr{P}$. Each point in $B \subseteq \mathbb{Z}^3$ is called a *black point* and has a value 1. Each point in $\mathbb{Z}^3 \setminus B$ is called a *white point* and has a value 0. An *object* is a maximal 26-connected set of black points, while a *white component* is a maximal 6-connected set of white points. Here it is assumed that a picture contains finitely many black points.

The *lexicographical order* relation "$\prec$" between two distinct points $p = (p_x, p_y, p_z)$ and $q = (q_x, q_y, q_z)$ in $\mathbb{Z}^3$ is defined as follows:

$$p \prec q \quad \Leftrightarrow \quad (p_z < q_z) \vee (p_z = q_z \wedge p_y < q_y) \vee (p_z = q_z \wedge p_y = q_y \wedge p_x < q_x).$$

Let $C \subseteq \mathbb{Z}^3$ be a set of points. Point $p \in C$ is the *smallest element* of C if for any $q \in C \backslash \{p\}$, $p \prec q$.

A *unit lattice square* is a set of four mutually 18-adjacent points in $\mathbb{Z}^3$, while a *unit lattice cube* is a set of eight mutually 26-adjacent points in $\mathbb{Z}^3$.

A black point is called a *border point* in $(26, 6)$ pictures if it is 6-adjacent to at least one white point. A border point p is called a **U**-*border point* if the point marked $\mathbf{U} = u(p)$ in Fig. 6.1 is a white point. We can define **D**-, **N**-, **E**-, **S**-, and **W**-border points in the same way. A black point is called an *interior point* if it is not a border point. A *simple* point in a $(26, 6)$ picture is a black point whose deletion is a topology preserving reduction operation [12]. Note that simplicity of point p in $(26, 6)$ pictures is a local property that can be decided by investigating the set $N_{26}(p)$ [12].

Parallel reduction operations delete a set of black points and not just a single simple point. Hence we need to consider what is meant by topology preservation when a number of black points are deleted simultaneously.

Ma [17] gave some *sufficient conditions* for 3D parallel reduction operations to preserve topology. Later, Palágyi and Kuba proposed the following simplified conditions [36]:

Theorem 1 ([36]) *The parallel reduction operation $\mathcal{O}$ is topology preserving for* $(26, 6)$ *pictures if all the following conditions hold.*

1. *Only simple points are deleted by $\mathcal{O}$.*
2. *Let p be any black point in a picture $(\mathbb{Z}^3, 26, 6, B)$ such that p is deleted by $\mathcal{O}$. Let $Q \subseteq B$ be any set of simple points in $(\mathbb{Z}^3, 26, 6, B)$ such that $p \in Q$, and Q is contained in a unit lattice square. Then point p is simple in picture $(\mathbb{Z}^3, 26, 6, B \backslash (Q \backslash \{p\}))$.*
3. *No object contained in a unit lattice cube is deleted completely by $\mathcal{O}$.*

Theorem 1 provides a general method of verifying that a parallel thinning algorithm preserves topology. In this section, we present some new sufficient conditions for topology preservation as a basis for designing 3D parallel thinning algorithms.

Theorem 2 *The parallel reduction operation $\mathcal{O}$ is topology preserving for* $(26, 6)$ *pictures if all the following conditions hold for any black point p in any picture $(\mathbb{Z}^3, 26, 6, B)$ such that p is deleted by $\mathcal{O}$.*

1. *Point p is simple in $(\mathbb{Z}^3, 26, 6, B)$.*
2. *Let $Q \subseteq B$ be any set of simple points in $(\mathbb{Z}^3, 26, 6, B)$ such that $p \in Q$, and Q is contained in a unit lattice square. Then point p is simple in picture $(\mathbb{Z}^3, 26, 6, B \backslash (Q \backslash \{p\}))$, or p is not the smallest element of Q.*
3. *Point p is not the smallest element of any object $C \subseteq B$ in $(\mathbb{Z}^3, 26, 6, B)$ such that C is contained in a unit lattice cube.*

Proof It can be readily seen that if the parallel reduction operation $\mathcal{O}$ satisfies Condition i of Theorem 2, then Condition i of Theorem 1 is also satisfied ($i =$

1, 2, 3). Hence, parallel reduction operation $\mathscr{O}$ is topology preserving for (26, 6) pictures. □

6.3 Variations on Parallel 3D Thinning Algorithms

In this section, 15 parallel 3D thinning algorithms are presented. These algorithms are composed of parallel reduction operations derived from our sufficient conditions for topology preservation (see Theorem 2).

Thinning algorithms preserve endpoints and some border points that provide relevant geometrical information with respect to the shape of the object. Here, we consider three types of endpoints.

Definition 1 There is no endpoint of type **TK**.

To standardize the notations, shrinking algorithms capable of producing topological kernels are considered as kernel-thinning algorithms, where no endpoint is preserved, hence we use endpoints of type **TK** (i.e., the empty set of the endpoints).

Definition 2 A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is a curve-endpoint of type **CE** if $(N_{26}(p)\backslash\{p\}) \cap B$ contains exactly one point (i.e., p is 26-adjacent to exactly one further black point).

Endpoints of type **CE** have been considered by numerous existing 3D curve-thinning algorithms [26–28, 34–36, 38].

Definition 3 A black point p in picture $(\mathbb{Z}^3, 26, 6, B)$ is a surface-endpoint of type **SE** if there is no interior point in $N_{26}(p) \cap B$.

Note that the characterization of endpoints **SE** is applied in some existing surface-thinning algorithms [24, 26–28, 31, 33, 37].

In the rest of this section we present parallel 3D thinning algorithms composed of parallel reduction operations that satisfy Theorem 2.

6.3.1 Fully Parallel Algorithms

In fully parallel algorithms, the same parallel reduction operation is applied in each iteration step [1, 15, 16, 18, 19, 33, 45].

The scheme of the proposed fully parallel thinning algorithm 3D-FP-ε using endpoint of type ε is sketched in Algorithm 1 ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$). Note that Palágyi and Németh reported three fully parallel 3D surface-thinning algorithms in [37], but they are based on sufficient conditions that differ from the conditions of Theorem 2.

Algorithm 1 Algorithm 3D-FP-ε

```
1: Input: picture (ℤ^3, 26, 6, X)
2: Output: picture (ℤ^3, 26, 6, Y)
3: Y = X
4: repeat
5:    // one iteration step
6:    D = {p | p is 3D-FP-ε-deletable in Y}
7:    Y = Y \ D
8: until D = ∅
```

3D-FP-ε-deletable points are defined as follows:

Definition 4 A black point is *3D-FP-ε-deletable* if it is not an endpoint of type ε, and all conditions of Theorem 2 hold ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$).

We have the following theorem.

Theorem 3 *Algorithm* 3D-FP-ε ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$) *is topology preserving for* (26, 6) *pictures*.

Proof Deletable points of the proposed fully parallel algorithms (see Definition 4) are derived directly from conditions of Theorem 2. Hence, all of the three algorithms are topology preserving. □

Note that all objects contained in a unit lattice cube are formed of endpoints of type **SE**. Hence, Condition 3 of Theorem 2 can be ignored in algorithm 3D-FP-**SE**.

6.3.2 Subiteration-Based Algorithms

In subiteration-based (or frequently referred to as directional) thinning algorithms, an iteration step is decomposed into k successive parallel reduction operations according to k deletion directions [8]. If the current deletion direction is d, then a set of d-border points can be deleted by the parallel reduction operation assigned to it. Since there are six kinds of major directions in 3D cases, 6-subiteration algorithms were generally proposed [2, 7, 13, 20, 25, 34, 43, 46]. Moreover, 3-subiteration [30–32], 8-subiteration [35], and 12-subiteration [36] algorithms have also been developed for this task.

In what follows, we present three examples of parallel 3D 6-subiteration thinning algorithms. Algorithm 2 sketches the scheme of 3D 6-subiteration parallel thinning algorithm 3D-6-SI-ε that uses the endpoint of type ε ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$).

Algorithm 2 Algorithm 3D-6-SI-ε

Input: picture $(\mathbb{Z}^3, 26, 6, X)$
Output: picture $(\mathbb{Z}^3, 26, 6, Y)$
$Y = X$
repeat
  // one iteration step
  for each $i \in \{\mathbf{U}, \mathbf{D}, \mathbf{N}, \mathbf{E}, \mathbf{S}, \mathbf{W}\}$ **do**
    // subiteration for deleting some i-border points
    $D(i) = \{p \mid p$ is a 3D-6-SI-i-ε-deletable point in $Y\}$
    $Y = Y \setminus D(i)$
  end for
until $D(\mathbf{U}) \cup D(\mathbf{D}) \cup D(\mathbf{N}) \cup D(\mathbf{E}) \cup D(\mathbf{S}) \cup D(\mathbf{W}) = \emptyset$

The ordered list of deletion directions $\langle \mathbf{U}, \mathbf{D}, \mathbf{N}, \mathbf{E}, \mathbf{S}, \mathbf{W} \rangle$ [7, 34] is considered in the proposed algorithm 3D-6-SI-ε ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$). Note that subiteration-based thinning algorithms are not invariant under the order of deletion directions (i.e., choosing different orders may yield various results).

In the first subiteration of our 6-subiteration thinning algorithms, the set of *3D-6-SI-***U**-ε*-deletable* points are deleted simultaneously, and the set of *3D-6-SI-***W**-ε*-deletable* points are deleted in the last (i.e., the 6th) subiteration. Now we lay down *3D-6-SI-***U**-ε*-deletable* points.

Definition 5 A black point p in picture $(\mathbb{Z}^3, 26, 6, X)$ is 3D-6-SI-**U**-ε-deletable if all of the following conditions hold:

1. Point p is a simple and **U**-border point, but it is not an endpoint of type ε in picture $(\mathbb{Z}^3, 26, 6, X)$.
2. Let $\mathscr{A}(p)$ be the family of the following 13 sets (see Fig. 6.2b):

$$\{e\}, \{s\}, \{se\}, \{sw\}, \{dn\}, \{de\}, \{ds\}, \{dw\},$$
$$\{e, s\}, \{e, se\}, \{s, se\}, \{s, sw\},$$
$$\{e, s, se\}.$$

   For any set A in the family $\mathscr{A}(p)$ composed of simple and **U**-border points, but not endpoints of type ε in picture $(\mathbb{Z}^3, 26, 6, X)$, point p remains simple in picture $(\mathbb{Z}^3, 26, 6, X \setminus A)$.
3. Let $\mathscr{B}(p)$ be the family of the following 42 objects in picture $(\mathbb{Z}^3, 26, 6, X)$ (see Fig. 6.2c):

$$\{a, h\}, \{b, g\}, \{c, f\}, \{d, e\},$$
$$\{a, h, b\}, \{a, h, c\}, \{a, h, f\}, \{a, h, g\}, \{b, g, a\}, \{b, g, d\}, \{b, g, e\}, \{b, g, h\},$$
$$\{c, f, a\}, \{c, f, d\}, \{c, f, e\}, \{c, f, h\}, \{d, e, b\}, \{d, e, c\}, \{d, e, f\}, \{d, e, g\},$$
$$\{b, c, h\}, \{d, g, f\}, \{a, d, f\}, \{b, e, h\}, \{b, c, e\}, \{a, f, g\}, \{a, d, g\}, \{c, e, h\},$$
$$\{a, h, b, c\}, \{a, h, b, g\}, \{a, h, c, f\}, \{a, h, f, g\}, \{b, g, a, d\}, \{b, g, d, e\},$$

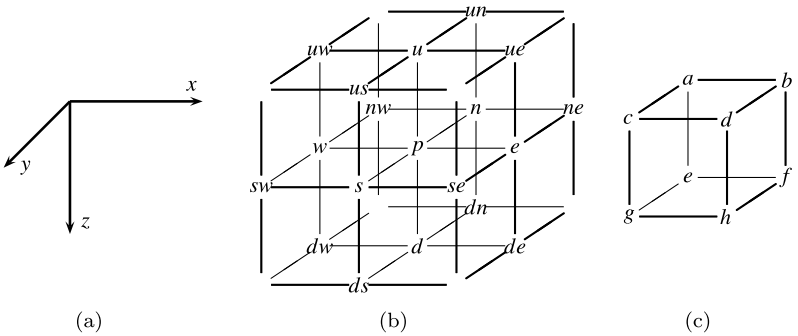


Fig. 6.2 The considered right-handed 3D coordinate system (**a**). Notation for the points in $N_{18}(p)$ (**b**). Notation for the points in a unit lattice cube (**c**)

$$\{b, c, e, h\}, \{b, g, e, h\}, \{c, f, a, d\}, \{c, f, d, e\}, \{c, f, e, h\}, \{d, e, b, c\},$$
$$\{d, e, f, g\}, \{a, d, f, g\}.$$

Point p is not the smallest element of any object in $\mathscr{B}(p)$.

Note that the deletable points at the remaining five subiterations can be derived from 3D-6-SI-**U**-ε-deletable points (assigned to the deletion direction **U**, see Definition 5) by reflexions and rotations.

Theorem 4 *Algorithm* 3D-6-SI-ε ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$) *is topology preserving for* (26, 6) *pictures*.

Proof Without loss of generality, it is sufficient to prove that the first subiteration of algorithm 3D-6-SI-ε is topology preserving. To this end, we show that the parallel reduction operation $\mathscr{T}$ that deletes 3D-6-SI-**U**-ε-deletable points ($\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$) satisfies all conditions of Theorem 2.

1. Operation $\mathscr{T}$ may delete simple points by Condition 1 of Definition 5. Hence Condition 1 of Theorem 2 is satisfied.
2. It is easy to see that the family $\mathscr{A}(p)$ (see Condition 2 of Definition 5 and Fig. 6.2a–b) contains all possible sets of simple **U**-border points that are considered by Condition 2 of Theorem 2. Therefore, this latter condition is satisfied.
3. It can be readily seen that the family of objects $\mathscr{B}(p)$ (see Condition 3 of Definition 5 and Fig. 6.2c) contains all possible objects of **U**-border points that are considered in Condition 3 of Theorem 2. Hence, this last condition is satisfied.

   Since objects contained in a unit lattice cube are composed of endpoints of type **SE**, Condition 3 of Definition 5 can be ignored in algorithm 3D-6-SI-**SE**. □

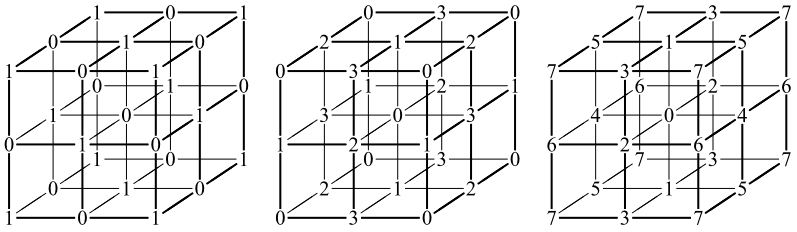


Fig. 6.3 The divisions of $\mathbb{Z}^3$ into 2 (**a**), 4 (**b**), and 8 (**c**) subfields. If partitioning into k subfields is considered, then points marked "i" are in the subfield $SF_k(i)$ ($k = 2, 4, 8$; $i = 0, 1, \ldots, k-1$)

6.3.3 Subfield-Based Algorithms

The third type of parallel thinning algorithms applies subfield-based technique [8]. In existing subfield-based parallel 3D thinning algorithms, the digital space $\mathbb{Z}^3$ is partitioned into two [21, 22, 26], four [23, 27], and eight [3, 27] subfields which are alternatively activated. At a given iteration step of a k-subfield algorithm, k successive parallel reduction operations associated to the k subfields are performed. In each of them, some border points in the active subfield can be designated for deletion.

Let us denote $SF_k(i)$ the i-th subfield if $\mathbb{Z}^3$ is partitioned into k subfields ($k = 2, 4, 8$; $i = 0, \ldots, k-1$). $SF_k(i)$ is defined formally as follows:

$$SF_2(i) = \big\{(p_x, p_y, p_z) \mid (p_x + p_y + p_z \bmod 2) = i\big\},$$

$$SF_4(i) = \big\{(p_x, p_y, p_z) \mid (p_x + 1 \bmod 2) \cdot \big[2 \cdot (p_y \bmod 2) + (p_z \bmod 2)\big] + (p_x \bmod 2) \cdot \big[2 \cdot (p_y + 1 \bmod 2) + (p_z + 1 \bmod 2)\big] = i\big\},$$

$$SF_8(i) = \big\{(p_x, p_y, p_z) \mid 4 \cdot (p_x \bmod 2) + 2 \cdot (p_y \bmod 2) + (p_z \bmod 2) = i\big\}$$

The considered divisions are illustrated in Fig. 6.3.

Proposition 1 *For the* 2*-subfield case* (*see Fig.* 6.3a), *two points* p *and* $q \in N_{26}(p)$ *are in the same subfield, if* $q \in N_{18}(p) \backslash N_6(p)$.

Proposition 2 *For the* 4*-subfield case* (*see Fig.* 6.3b), *two points* p *and* $q \in N_{26}(p)$ *are in the same subfield, if* $q \in N_{26}(p) \backslash N_{18}(p)$.

Proposition 3 *For the* 8*-subfield case* (*see Fig.* 6.3c), *two points* p *and* $q \in N_{26}(p)$ *are* not *in the same subfield.*

In order to reduce the noise sensitivity and the number of skeletal points (without overshrinking the objects), Németh, Kardos, and Palágyi introduced a new subfield-based thinning scheme [26]. It takes the endpoints into consideration at the beginning of iteration steps, instead of preserving them in each parallel reduction operation as it is accustomed in the conventional subfield-based thinning scheme.

Next, we present nine parallel 3D subfield-based thinning algorithms. The scheme of the subfield-based parallel thinning algorithm 3D-k-SF-ε with iteration-level endpoint checking using endpoint of type ε is sketched in Algorithm 3 (with $k = 2, 4, 8$; $\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$).

Algorithm 3 Algorithm 3D-k-SF-ε

Input: picture $(\mathbb{Z}^3, 26, 6, X)$
Output: picture $(\mathbb{Z}^3, 26, 6, Y)$
$Y = X$
repeat
  // one iteration step
  $E = \{p \mid p$ is a border point, but not an endpoint of type ε in $Y\}$
  for $i = 0$ **to** $k - 1$ **do**
    // subfield $SF_k(i)$ is activated
    $D(i) = \{q \mid q$ is a 3D-SF-k-deletable point in $E \cap SF_k(i)\}$
    $Y = Y \setminus D(i)$
  end for
until $D(0) \cup D(1) \cup \ldots \cup D(k-1) = \emptyset$

The 3D-SF-k-deletable points are defined as follows ($k = 2, 4, 8$):

Definition 6 A black point p is 3D-SF-k-deletable ($k = 2, 4, 8$) in picture $(\mathbb{Z}^3, 26, 6, X)$ if all of the following conditions hold:

1. Point p is simple in $(\mathbb{Z}^3, 26, 6, X)$.
2. If $k = 2$, then point p is simple in picture $(\mathbb{Z}^3, 26, 6, X \setminus \{q\})$ for any simple point q such that $q \in N_{18}(p) \setminus N_6(p)$ and $p \prec q$.
3. - If $k = 2$, then point p is not the smallest element of the ten objects depicted in Fig. 6.4.
   - If $k = 4$, then point p is not the smallest element of the four objects depicted in Fig. 6.5.

Theorem 5 *Algorithm* 3D-k-SF-ε ($k = 2, 4, 8$; $\varepsilon \in \{\mathbf{TK}, \mathbf{CE}, \mathbf{SE}\}$) *is topology preserving for* $(26, 6)$ *pictures*.

Proof To prove it, we show that the parallel reduction operation $\mathscr{T}$ that deletes 3D-SF-k-deletable points satisfies all conditions of Theorem 2.

1. Operation $\mathscr{T}$ may delete simple points by Condition 1 of Definition 6. Hence Condition 1 of Theorem 2 is satisfied.
2. - Let $k = 2$ and let $p \in SF_2(i)$ be any black point in picture $(\mathbb{Z}^3, 26, 6, X)$ that is deleted by $\mathscr{T}$ ($i = 0, 1$).

     Let $Q \subseteq X \cap SF_2(i)$ be any set of black points in $(\mathbb{Z}^3, 26, 6, X)$ such that $p \in Q$, Q is contained in a unit lattice square, and each point in $Q \setminus \{p\}$ is simple in picture $(\mathbb{Z}^3, 26, 6, X)$.

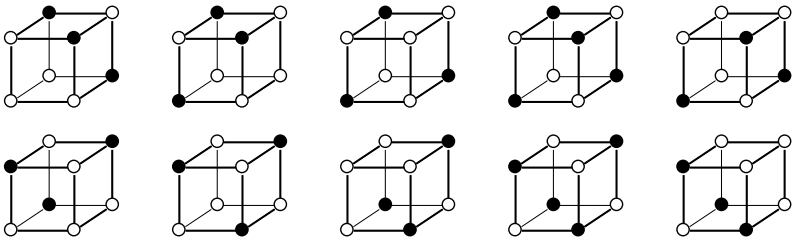


Fig. 6.4 The ten objects that are taken into consideration by 2-subfield algorithms. Notations: each point marked by "●" is a black point; each point marked by "○" is a white point. (Note that each of these objects is contained in a unit *lattice cube*)

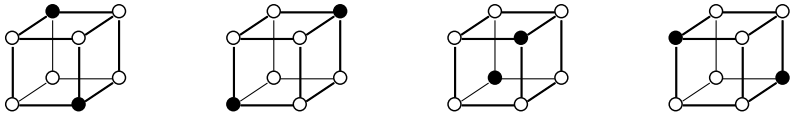


Fig. 6.5 The four objects considered by 4-subfield algorithms. Notations: each point marked "●" is a black point; each point marked "○" is a white point. (Note that each of these objects is contained in a unit *lattice cube*)

Then $Q = \emptyset$ or $Q = \{q\}$ by Proposition 1, and such kind of sets are considered by Condition 2 of Definition 6. Hence Condition 2 of Theorem 2 is satisfied.

- Let $k = 4$ and let $p \in SF_4(i)$ be any black point in picture $(\mathbb{Z}^3, 26, 6, X)$ that is deleted by $\mathscr{T}$ $(i = 0, 1, 2, 3)$.

  Let $Q \subseteq X \cap SF_4(i)$ be any set of black points in $(\mathbb{Z}^3, 26, 6, X)$ such that $p \in Q$, Q is contained in a unit lattice square, and each point in $Q \backslash \{p\}$ is simple in picture $(\mathbb{Z}^3, 26, 6, X)$.

  Then $Q = \emptyset$ by Proposition 2. Hence Condition 2 of Theorem 2 is satisfied.
- Let $k = 8$ and let $p \in SF_8(i)$ be any black point in picture $(\mathbb{Z}^3, 26, 6, X)$ that is deleted by $\mathscr{T}$ $(i = 0, 1, \ldots, 7)$.

  Let $Q \subseteq X \cap SF_8(i)$ be any set of black points in $(\mathbb{Z}^3, 26, 6, X)$ such that $p \in Q$, Q is contained in a unit lattice square, and each point in $Q \backslash \{p\}$ is simple in picture $(\mathbb{Z}^3, 26, 6, X)$.

  Then $Q = \emptyset$ by Proposition 3. Hence Condition 2 of Theorem 2 is satisfied.

3. - Let $k = 2$ and let $C \subseteq X \cap SF_2(i)$ be any object in picture $(\mathbb{Z}^3, 26, 6, X)$ that is contained in a unit lattice cube $(i = 0, 1)$.

     It can be readily seen by Proposition 1 that all the ten possible cases for such objects are depicted in Fig. 6.4, and these objects cannot be deleted completely by Condition 3 of Definition 6.

     Hence Condition 3 of Theorem 2 is satisfied.
   - Let $k = 4$ and let $C \subseteq X \cap SF_4(i)$ be any object in picture $(\mathbb{Z}^3, 26, 6, X)$ that is contained in a unit lattice cube $(i = 0, 1, 2, 3)$.

It can be readily seen by Proposition 2 that all the four possible cases for such objects are depicted in Fig. 6.5, and these objects cannot be deleted completely by Condition 3 of Definition 6.

Hence Condition 3 of Theorem 2 is satisfied.

- Let $k = 8$ and let $C \subseteq X \cap SF_8(i)$ be any object in picture $(\mathbb{Z}^3, 26, 6, X)$ that is contained in a unit lattice cube ($i = 0, 1, \ldots, 7$).

  It is easy to see that there is no such an object by Proposition 3.

  Hence Condition 3 of Theorem 2 is satisfied. □

Since objects contained in a unit lattice cube are composed of endpoints of type **SE**, Condition 3 of Definition 6 can be ignored in algorithm 3D-k-SF-**SE** ($k = 2, 4, 8$).

6.4 Implementation

One may think that the proposed algorithms are time consuming and it is rather difficult to implement them. That is why we outline a method for implementing any 3D fully parallel thinning algorithm on a conventional sequential computer. This framework is fairly general, as similar schemes can be used for the other classes of parallel algorithms and some sequential 3D thinning algorithms [33, 37, 38].

The proposed method uses a pre-calculated look-up-table to encode simple points. In addition, two lists are used to speed up the process: one for storing the border points in the current picture (since thinning can only delete border points, thus the repeated scans/traverses of the entire array storing the picture are avoided); the other list is to collect all deletable points in the current phase of the process. At each iteration, the deletable points are found and deleted, and the list of border points is updated accordingly. The algorithm terminates when no further update is required.

For simplicity, the pseudocode of the proposed 3D fully parallel thinning algorithms is given (see Algorithm 4). The subiteration-based and the subfield-based variants can be implemented in similar ways.

The two input parameters of the procedure are array A which stores the input picture to be thinned and the type of the considered endpoints ε. In input array A, the value "1" corresponds to black points and the value "0" denotes white ones. According to the proposed scheme, the input and the output pictures can be stored in the same array, hence array A will contain the resultant structure.

First, the input picture is scanned and all the border points are inserted into the list *border_list*. We should mention here that it is the only time consuming scanning. Since only a small part of points in a usual picture belong to the objects, the thinning procedure is much faster if we just deal with the set of border points in the actual picture. This subset of object points is stored in *border_list* (i.e., a dynamic data structure). The *border_list* is then updated: if a border point is deleted, then all interior points that are 6-adjacent to it become border points. These brand new

Algorithm 4 Fully parallel thinning algorithm

```
1: Input: array A and endpoint characterization ε
2: Output: array A
3: // collect border points
4: border_list = ⟨empty list⟩
5: for each p = (x, y, z) in A do
6:    if p is border point then
7:       border_list = border_list + ⟨p⟩
8:       A[x, y, z] = 2
9:    end if
10: end for
11: // thinning
12: repeat
13:    deleted = 0
14:    deletable_list = ⟨empty list⟩
15:    // checking Condition 1 of Theorem 2
16:    for each point p = (x, y, z) in border_list do
17:       if p is a simple point and not an endpoint of type ε then
18:          deletable_list = deletable_list + ⟨p⟩
19:          A[x, y, z] = 3
20:       else
21:          A[x, y, z] = 2
22:       end if
23:    end for
24:    // checking Condition 2 of Theorem 2
25:    for each point p in deletable_list do
26:       if deletion p does not satisfy Condition 2 of Theorem 2 then
27:          deletable_list = deletable_list − ⟨p⟩
28:       end if
29:    end for
30:    // checking Condition 3 of Theorem 2
31:    for each point p in deletable_list do
32:       if deletion p does not satisfy Condition 3 of Theorem 2 then
33:          deletable_list = deletable_list − ⟨p⟩
34:       end if
35:    end for
36:    // deletion
37:    for each point p = (x, y, z) in deletable_list do
38:       A[x, y, z] = 0
39:       border_list = border_list − ⟨p⟩
40:       deleted = deleted+1
41:       // update border_list
42:       for each point q = (x′, y′, z′) that is 6-adjacent to p do
43:          if A[x′, y′, z′] = 1 then
44:             A[x′, y′, z′] = 2
45:             border_list = border_list + ⟨q⟩
46:          end if
47:       end for
48:    end for
49: until deleted = 0
```

border points of the actual picture are added to the *border_list*. In order to avoid storing more than one copy of a border point in *border_list*, array A represents a four-color picture during the thinning process: the value "0" corresponds to the white points, the value "1" corresponds to (black) interior points, the value "2" is assigned to all (black) border points in the actual picture (added to *border_list*), and the value "3" corresponds to points that are added to the *deletable_list* (i.e., a sublist of *border_list*).

The kernel of the *repeat* cycle corresponds to one iteration step of the thinning process. The number of deleted points is stored in the variable called *deleted*. The thinning process terminates when no more points can be deleted (i.e., no further changes occur). After thinning, all points having a nonzero value belong to the produced skeleton-like shape feature.

Simple points in $(26, 6)$ pictures can be locally characterized; the simplicity of a point p can be decided by examining the set $N_{26}(p)$ [12]. There are 2^{26} possible configurations in the $3 \times 3 \times 3$ neighborhood if the central point is not considered. Hence we can assign an index (i.e., a non-negative integer code) for each possible configuration and address a pre-calculated (unit time access) look-up-table having 2^{26} entries of 1 bit in size, therefore, it requires only 8 megabytes storage space in memory.

By adapting this efficient implementation method, our algorithms can be well applied in practice: they are capable of extracting skeleton-like shape features from large 3D pictures containing 1 000 000 object points within one second on a standard PC.

6.5 Results

The proposed 15 algorithms were tested on objects of different shapes. Here we present some of them (see Figs. 6.6–6.12). The pairs of numbers in parentheses are the counts of object points in the produced skeleton-like structure and the parallel speed (i.e., the number of the performed parallel reduction operations [8]).

6.6 Possible Future Works and Open Problems

In this section, we will outline some possible future works and open problems concerning parallel 3D thinning.

- Conventional thinning algorithms preserve endpoints to provide important geometric information relative to the object to be represented. Bertrand and Couprie proposed an alternative strategy [4]. They developed a sequential thinning scheme based on a generalization of curve/surface interior points that are called *isthmus*es. Isthmuses are dynamically detected and accumulated in a constraint set of non-simple points.

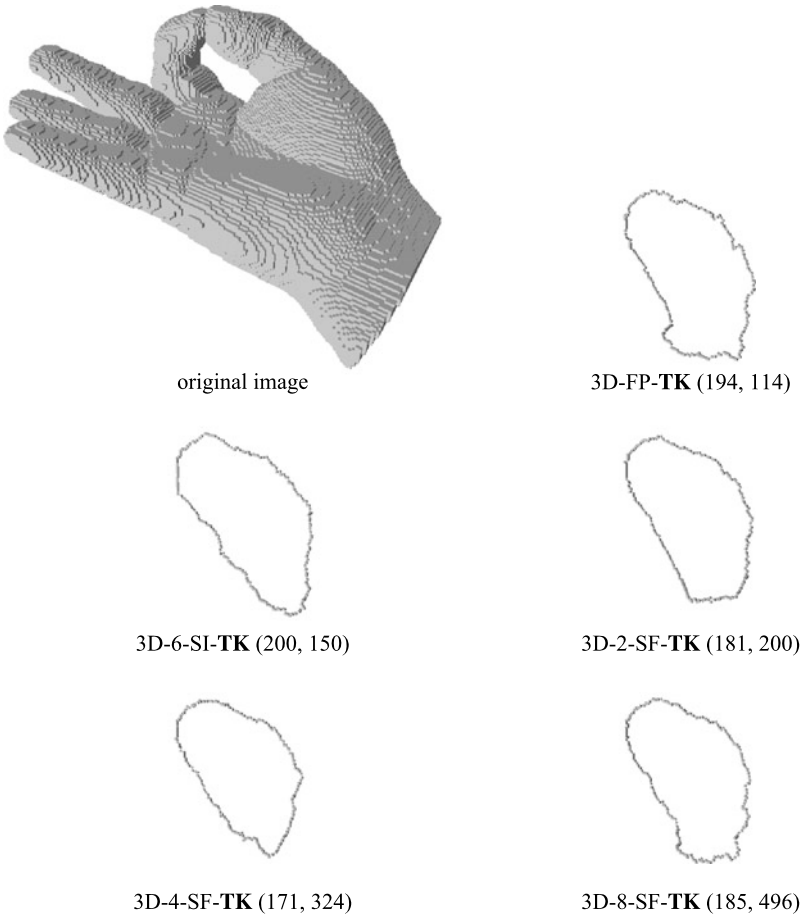


Fig. 6.6 A 191 × 96 × 114 image of a *hand* and its topological kernels produced by the five proposed parallel 3D kernel-thinning algorithms. The original image contains 455 295 black points. Since the original object contains a hole, there are holes in its topological kernels, too

The very first parallel 3D isthmus-based curve-thinning algorithm was designed by Raynal and Couprie [39]. Each iteration step of their 6-subiteration algorithm consists of two phases:

1. Updating the constraint set, by adding points detected as isthmuses;
2. Removing "deletable" points which are not in the constraint set.

Raynal and Couprie gave these "deletable" points by $3 \times 3 \times 3$ matching templates, and proved that simultaneous deletion of "deletable" points is a topology preserving reduction operation. Hence their algorithm is topology preserving.

In a forthcoming work, we are going to combine our sufficient conditions for topology preservation (see Theorem 2) with various parallel thinning strategies (i.e., fully parallel, subiteration-based, and subfield-based) and some character-

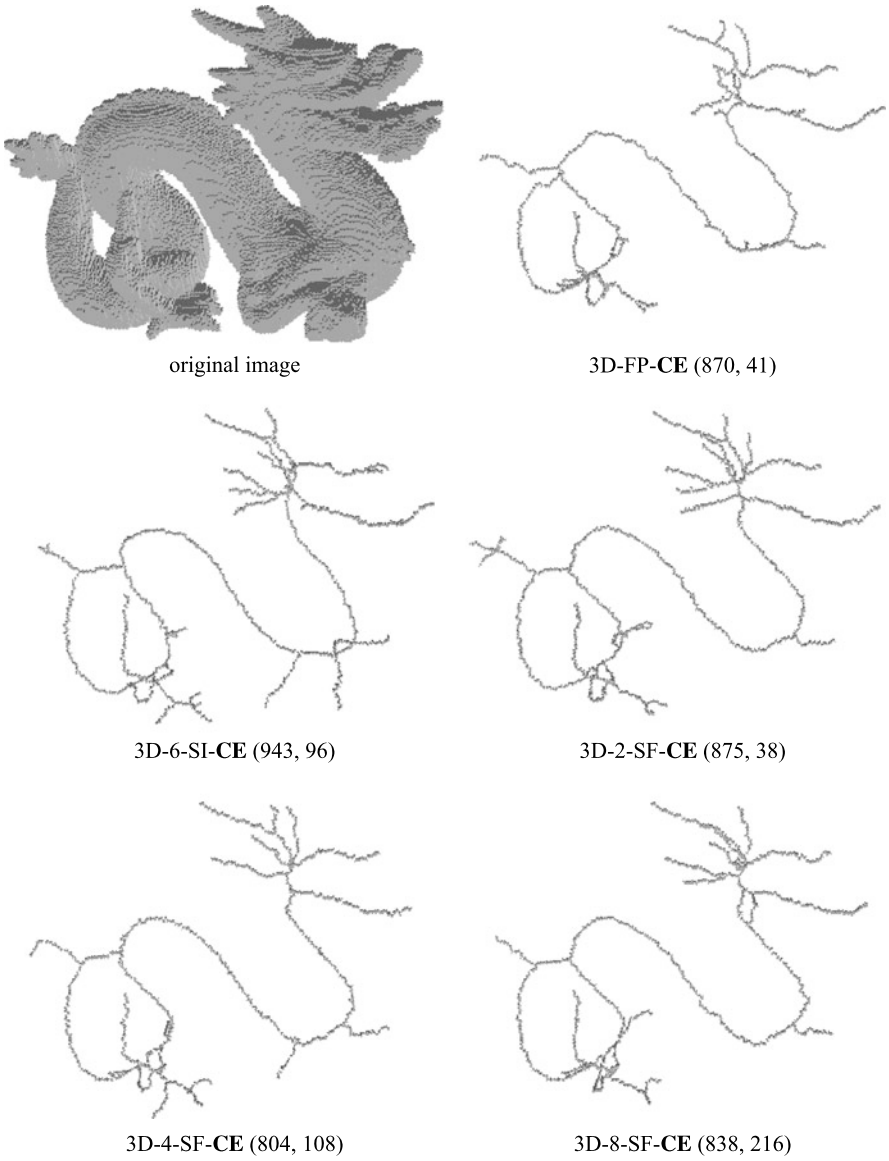


Fig. 6.7 A $135 \times 86 \times 191$ image of a *dragon* and its centerlines produced by the five proposed parallel 3D curve-thinning algorithms. The original image contains 423 059 black points

izations of isthmuses to generate new parallel 3D curve-thinning and surface-thinning algorithms.

- The 3D parallel thinning algorithms presented in this chapter are based on Theorem 2 (i.e., some sufficient conditions for topology preservation). Conditions 2 and 3 of Theorem 2 are "asymmetric", since points that are the smallest elements

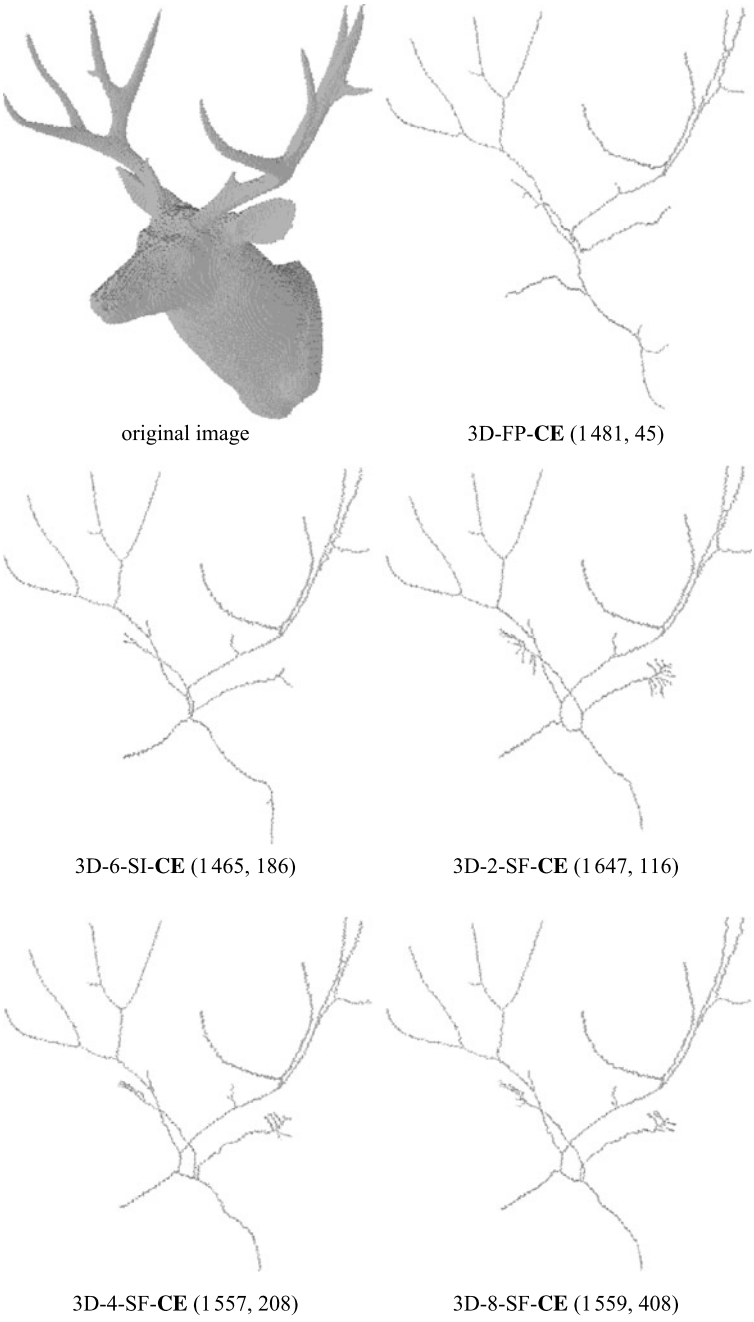


Fig. 6.8 A $380 \times 287 \times 271$ image of a *deer head* and its centerlines produced by the five proposed parallel 3D curve-thinning algorithms. The original image contains 1 658 641 black points

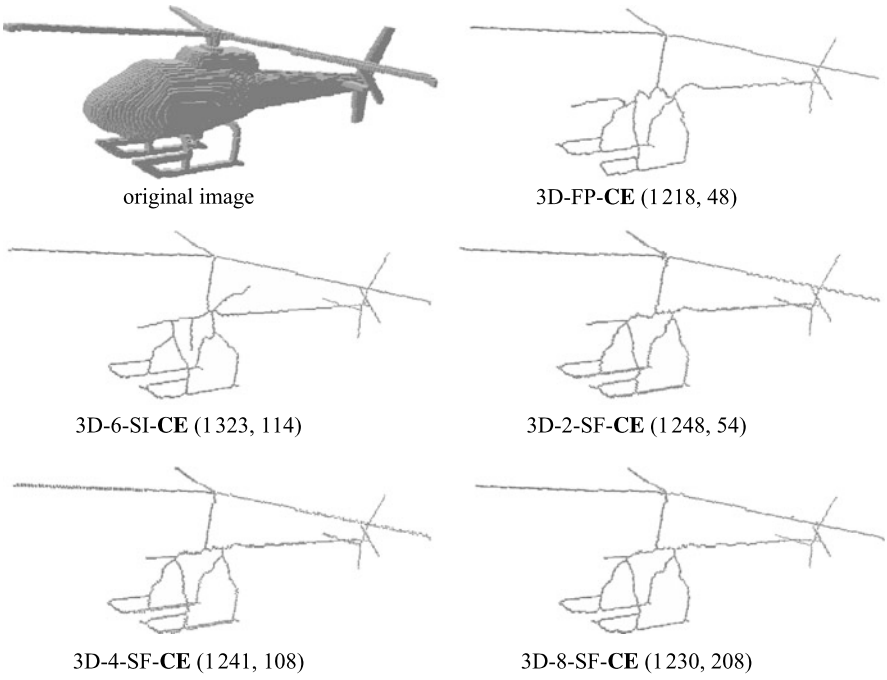


Fig. 6.9 A $103 \times 381 \times 255$ image of a *helicopter* and its centerlines produced by the five proposed parallel 3D curve-thinning algorithms. The original image contains 273 743 black points

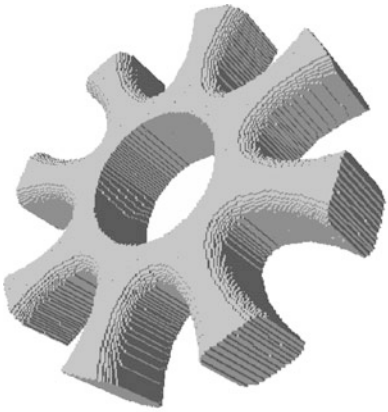
of some sets may not be deleted. It is easy to see that the following theorem provides "symmetric" conditions for topology preservation.

Theorem 6 *The parallel reduction operation $\mathcal{O}$ is topology preserving for* $(26, 6)$ *pictures if all the following conditions hold for any black point p in any picture* $(\mathbb{Z}^3, 26, 6, B)$ *such that p is deleted by $\mathcal{O}$.*

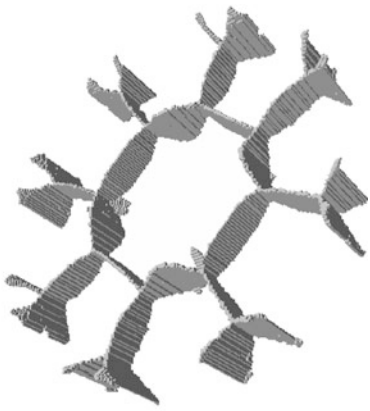
1. *Point p is simple in* $(\mathbb{Z}^3, 26, 6, B)$.
2. *Let $Q \subseteq B$ be any set of simple points in* $(\mathbb{Z}^3, 26, 6, B)$ *such that $p \in Q$, and Q is contained in a unit lattice square.*
   Then point p is simple in picture $(\mathbb{Z}^3, 26, 6, B \backslash (Q \backslash \{p\}))$.
3. *Point p is not an element of any object $C \subseteq B$ in* $(\mathbb{Z}^3, 26, 6, B)$ *such that C is contained in a unit lattice cube.*

In a future work, we plan to combine alternative sufficient conditions for topology preservation with parallel thinning strategies to generate further classes of 3D parallel thinning algorithms.

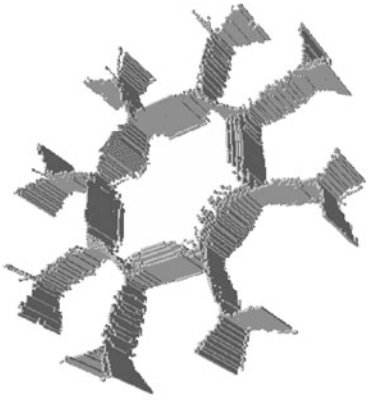
- Unfortunately, skeletonization methods (including thinning) are rather sensitive to coarse object boundaries, hence the produced skeletons generally contain some



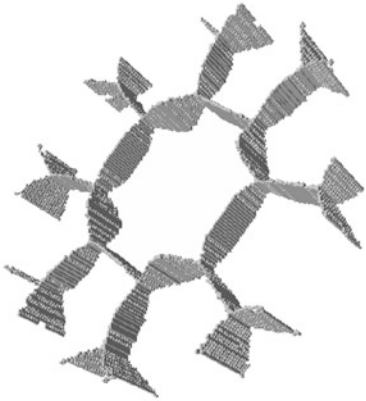
original image



3D-FP-**SE** (25 022, 18)

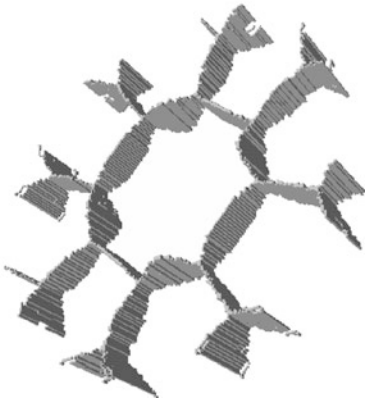


3D-6-SI-**SE** (24 873, 78)



3D-2-SF-**SE** (17 825, 34)

3D-4-SF-**SE** (17 235, 72)



3D-8-SF-**SE** (17 527, 136)

Fig. 6.10 A $45 \times 191 \times 191$ image of a *gear* and its medial surfaces produced by the five proposed parallel 3D surface-thinning algorithms. The original image contains 596 360 black points

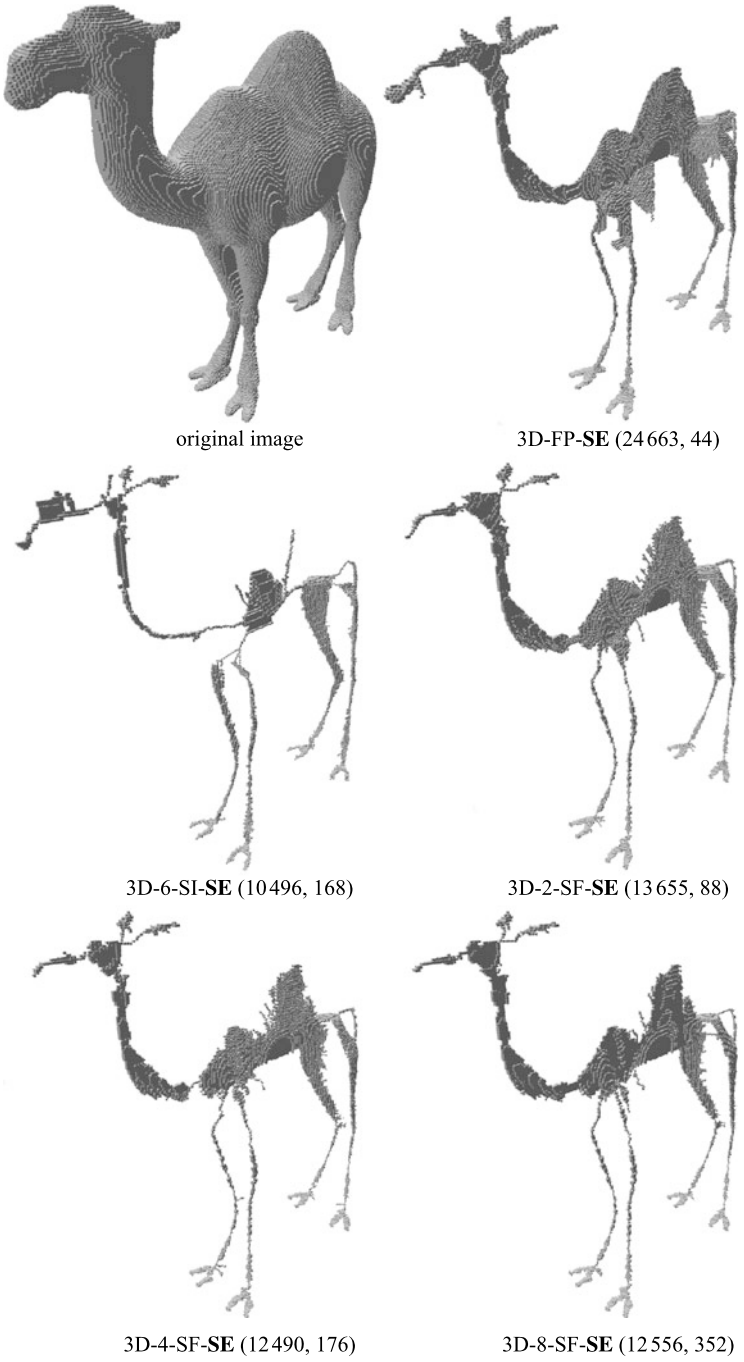


Fig. 6.11 A $285 \times 285 \times 88$ image of a *camel* and its medial surfaces produced by the five proposed parallel 3D surface-thinning algorithms. The original image contains 1 088 458 black points

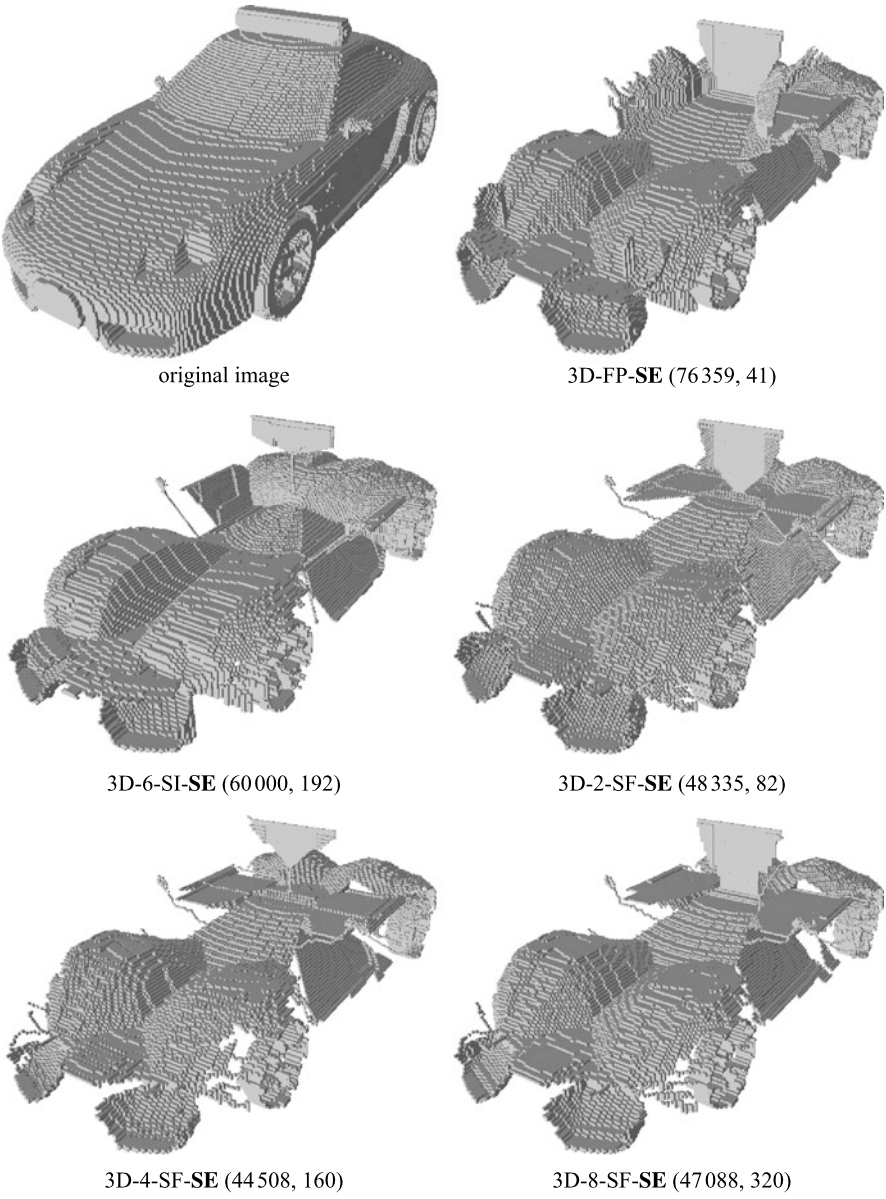


Fig. 6.12 A $122 \times 93 \times 284$ image of a *car* and its medial surfaces produced by the five proposed parallel 3D surface-thinning algorithms. The original image contains 1 321 764 black points

false segments. In order to overcome this problem, unwanted skeletal parts are usually removed by a *pruning* process as a post-processing step [40]. In [29], we presented a new thinning scheme for reducing the noise sensitivity of 3D thinning

algorithms. It uses iteration-by-iteration smoothing which removes some border points being considered as extremities.

We are going to design new topology preserving parallel contour smoothing operations, and combine our 3D parallel thinning algorithms (based on sufficient conditions for topology preservation) with iteration-by-iteration smoothing.

- It is easy to see that subiteration-based and subfield-based parallel thinning schemes are not invariant under the order of deletion directions and subfield activations, respectively. It means that choosing different orders of directions may yield various results in subiteration-based algorithms, and varieties of skeleton-like shape features can be produced by a subfield-based algorithm with diverse orders of the active subfields.

  Neither *order-independent* subiteration-based nor subfield-based parallel thinning algorithms have been proposed. We are going to deal with this unsolved problem (i.e., we plan to construct subiteration-based and subfield-based algorithms that produce the same result for any order of deletion directions and subfield activation).

6.7 Concluding Remarks

Fast and reliable extraction of skeleton-like shape features (i.e., medial surface, centerline, and topological kernel) is extremely important in numerous applications for large 3D shapes. In this chapter we presented a variety of parallel 3D thinning algorithms and their efficient implementation. They are based on some sufficient conditions for topology preserving parallel reduction operations, hence their topological correctness is guaranteed. The algorithms are based on different characterizations of endpoints. Additional types of endpoints coupled with sufficient conditions for topology preservation yield newer thinning algorithms.

Acknowledgements This research was supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency, the European Union and the European Regional Development Fund under the grant agreement TÁMOP-4.2.1/B-09/1/KONV-2010-0005, and the grant CNK80370 of the National Office for Research and Technology (NKTH) & the Hungarian Scientific Research Fund (OTKA).

References

1. Arcelli, C., Sanniti di Baja, G., Serino, L.: New removal operators for surface skeletonization. In: Proc. 13th Int. Conf. Discrete Geometry for Computer Imagery, DGCI 2006. Lecture Notes in Computer Science, vol. 4245, pp. 555–566. Springer, Heidelberg (2006)
2. Bertrand, G.: A parallel thinning algorithm for medial surfaces. Pattern Recognit. Lett. **16**, 979–986 (1995)
3. Bertrand, G., Aktouf, Z.: A 3D thinning algorithm using subfields. In: SPIE Proc. of Conf. on Vision Geometry, pp. 113–124 (1994)

4. Bertrand, G., Couprie, M.: Transformations topologiques discrètes. In: Coeurjolly, D., Montanvert, A., Chassery, J. (eds.) Géométrie discrète et images numériques, pp. 187–209. Hermès Science Publications-Lavoisier, Paris (2007)
5. Blum, H.: A transformation for extracting new descriptors of shape. In: Wathen-Dunn, W. (ed.) Models for the Perception of Speech and Visual Form, pp. 362–380. MIT Press, Cambridge (1967)
6. Gomberg, B.R., Saha, P.K., Song, H.K., Hwang, S.N., Wehrli, F.W.: Topological analysis of trabecular bone MR images. IEEE Trans. Med. Imaging **19**, 166–174 (2000)
7. Gong, W.X., Bertrand, G.: A simple parallel 3D thinning algorithm. In: Proc. 10th IEEE Internat. Conf. on Pattern Recognition, ICPR'90, pp. 188–190 (1990)
8. Hall, R.W.: Parallel connectivity-preserving thinning algorithms. In: Kong, T.Y., Rosenfeld, A. (eds.) Topological Algorithms for Digital Image Processing, pp. 145–179. Elsevier, Amsterdam (1996)
9. Hall, R.W., Kong, T.Y., Rosenfeld, A.: Shrinking binary images. In: Kong, T.Y., Rosenfeld, A. (eds.) Topological Algorithms for Digital Image Processing, pp. 31–98. Elsevier, Amsterdam (1996)
10. Itoh, T., Yamaguchi, Y., Koyamada, K.: Fast isosurface generation using the volume thinning algorithm. IEEE Trans. Vis. Comput. Graph. **7**, 32–46 (2001)
11. Kong, T.Y.: On topology preservation in 2-d and 3-d thinning. Int. J. Pattern Recognit. Artif. Intell. **9**, 813–844 (1995)
12. Kong, T.Y., Rosenfeld, A.: Digital topology: introduction and survey. Comput. Vis. Graph. Image Process. **48**, 357–393 (1989)
13. Lee, T., Kashyap, R.L., Chu, C.: Building skeleton models via 3-D medial surface/axis thinning algorithms. CVGIP, Graph. Models Image Process. **56**, 462–478 (1994)
14. Lohou, C.: Detection of the non-topology preservation of Ma's 3D surface-thinning algorithm, by the use of P-simple points. Pattern Recognit. Lett. **29**, 822–827 (2008)
15. Lohou, C., Dehos, J.: An automatic correction of Ma's thinning algorithm based on P-simple points. J. Math. Imaging Vis. **36**, 54–62 (2010)
16. Lohou, C., Dehos, J.: Automatic correction of Ma and Sonka's thinning algorithm using P-simple points. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 1148–1152 (2010)
17. Ma, C.M.: On topology preservation in 3D thinning. CVGIP, Image Underst. **59**, 328–339 (1994)
18. Ma, C.M.: A 3D fully parallel thinning algorithm for generating medial faces. Pattern Recognit. Lett. **16**, 83–87 (1995)
19. Ma, C.M., Sonka, M.: A fully parallel 3D thinning algorithm and its applications. Comput. Vis. Image Underst. **64**, 420–433 (1996)
20. Ma, C.M., Wan, S.-Y.: Parallel thinning algorithms on 3D (18, 6) binary images. Comput. Vis. Image Underst. **80**, 364–378 (2000)
21. Ma, C.M., Wan, S.Y.: A medial-surface oriented 3-d two-subfield thinning algorithm. Pattern Recognit. Lett. **22**, 1439–1446 (2001)
22. Ma, C.M., Wan, S.Y., Chang, H.K.: Extracting medial curves on 3D images. Pattern Recognit. Lett. **23**, 895–904 (2002)
23. Ma, C.M., Wan, S.Y., Lee, J.D.: Three-dimensional topology preserving reduction on the 4-subfields. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1594–1605 (2002)
24. Manzanera, A., Bernard, T.M., Pretêux, F., Longuet, B.: Medial faces from a concise 3D thinning algorithm. In: Proc. 7th IEEE Int. Conf. Computer Vision, ICCV'99, pp. 337–343 (1999)
25. Mukherjee, J., Das, P.P., Chatterjee, B.N.: On connectivity issues of ESPTA. Pattern Recognit. Lett. **11**, 643–648 (1990)
26. Németh, G., Kardos, P., Palágyi, K.: Topology preserving 2-subfield 3D thinning algorithms. In: Proc. 7th IASTED Int. Conf. Signal Processing, Pattern Recognition and Applications, pp. 310–316 (2010)
27. Németh, G., Kardos, P., Palágyi, K.: Topology preserving 3D thinning algorithms using four and eight subfields. In: Proc. International Conference on Image Analysis and Recognition,

ICIAR 2010. Lecture Notes in Computer Science, vol. 6111, pp. 316–325. Springer, Heidelberg (2010)
28. Németh, G., Kardos, P., Palágyi, K.: A family of topology-preserving 3D parallel 6-subiteration thinning algorithms. In: Proc. 14th International Workshop on Combinatorial Image Analysis, IWCIA'2011, Madrid, Spain. Lecture Notes in Computer Science, vol. 6636, pp. 17–30. Springer, Heidelberg (2011)
29. Németh, G., Kardos, P., Palágyi, K.: Thinning combined with iteration-by-iteration smoothing for 3D binary images. Graph. Models **73**, 335–345 (2011)
30. Palágyi, K.: A 3-subiteration 3D thinning algorithm for extracting medial surfaces. Pattern Recognit. Lett. **23**, 663–675 (2002)
31. Palágyi, K.: A 3-subiteration surface-thinning algorithm. In: Proc. 12th Int. Conf. Computer Analysis of Images and Patterns, CAIP 2007, Vienna, Austria. Lecture Notes in Computer Science, vol. 4673, pp. 628–635. Springer, Heidelberg (2007)
32. Palágyi, K.: A subiteration-based surface-thinning algorithm with a period of three. In: Hamprecht, F., Jähne, B., Schnörr, C. (eds.) Lecture Notes in Computer Science, vol. 4713, pp. 294–303. Springer, Heidelberg (2007)
33. Palágyi, K.: A 3D fully parallel surface-thinning algorithm. Theor. Comput. Sci. **406**, 119–135 (2008)
34. Palágyi, K., Kuba, A.: A 3D 6-subiteration thinning algorithm for extracting medial lines. Pattern Recognit. Lett. **19**, 613–627 (1998)
35. Palágyi, K., Kuba, A.: Directional 3D thinning using 8 subiterations. In: Proc. 8th Int. Conf. on Discrete Geometry for Computer Imagery, DGCI'99, Marne-la-Valle, France. Lecture Notes in Computer Science, vol. 1568, pp. 325–336. Springer, Heidelberg (1999)
36. Palágyi, K., Kuba, A.: A parallel 3D 12-subiteration thinning algorithm. Graph. Models Image Process. **61**, 199–221 (1999)
37. Palágyi, K., Németh, G.: Fully parallel 3D thinning algorithms based on sufficient conditions for topology preservation. In: Proc. 15th IAPR International Conference on Discrete Geometry for Computer Imagery, DGCI 2009. Lecture Notes in Computer Science, vol. 5810, pp. 481–492. Springer, Heidelberg (2009)
38. Palágyi, K., Tschirren, J., Hoffman, E.A., Sonka, M.: Quantitative analysis of pulmonary airway tree structures. Comput. Biol. Med. **36**, 974–996 (2006)
39. Raynal, B., Couprie, M.: Directional 3D thinning using 8 subiterations. In: Proc. 16th Int. Conf. on Discrete Geometry for Computer Imagery, DGCI 2011, Nancy, France. Lecture Notes in Computer Science, vol. 6607, pp. 175–186. Springer, Heidelberg (2011)
40. Shaked, D., Bruckstein, A.: Pruning medial axes. Comput. Vis. Image Underst. **69**, 156–169 (1998)
41. Siddiqi, K., Pizer, S. (eds.): Medial Representations—Mathematics, Algorithms and Applications. Computational Imaging and Vision, vol. 37. Springer, New York (2008)
42. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: Proc. Int. Conf. Shape Modeling and Applications, pp. 130–139. IEEE Press, New York (2003)
43. Tsao, Y.F., Fu, K.S.: A parallel thinning algorithm for 3-D pictures. Comput. Graph. Image Process. **17**, 315–331 (1981)
44. Wan, M., Liang, Z., Ke, Q., Hong, L., Bitter, I., Kaufman, A.: Automatic centerline extraction for virtual colonoscopy. IEEE Trans. Med. Imaging **21**, 1450–1460 (2002)
45. Wang, T., Basu, A.: A note on 'A fully parallel 3D thinning algorithm and its applications'. Pattern Recognit. Lett. **28**, 501–506 (2007)
46. Xie, W., Thompson, R.P., Perucchio, R.: A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton. Pattern Recognit. **36**, 1529–1544 (2003)