



ELSEVIER

Pattern Recognition Letters 23 (2002) 663–675

Pattern Recognition
Letters

www.elsevier.com/locate/patrec

A 3-subiteration 3D thinning algorithm for extracting medial surfaces

Kálmán Palágyi *

Department of Applied Informatics, University of Szeged, P.O. Box 652, H-6701 Szeged, Hungary

Abstract

Thinning is an iterative layer by layer erosion for extracting skeletons. This paper presents an efficient 3D parallel thinning algorithm which produces medial surfaces. A 3-subiteration strategy is proposed: the thinning operation is changed from iteration to iteration with a period of 3 according to the three deletion directions. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: 3D parallel thinning algorithm; Subiteration-based thinning; Digital topology; Topology preservation

1. Introduction

Skeletonization provides shape features that are extracted from binary image data. 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 (Blum, 1967). In discrete spaces, the thinning process is a frequently used method for producing an approximation to the skeleton in a topology-preserving way (Kong and Rosenfeld, 1989). It is 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. Therefore, a thinning algorithm can be regarded as

a reduction operation that changes some 1's (object elements or black points) to 0's (white points) but does not alter 0's.

A 3D reduction operation does *not* preserve topology (Kong, 1995) if

- any object in the input picture is split (into two or more parts) or completely deleted,
- any cavity in the input picture is merged with the background or with another cavity,
- a cavity is created where there was no cavity in the input picture,
- any hole in the input picture is merged with the background or with another hole, or
- a hole is created where there was no hole in the input picture.

A simple point is an object point whose deletion does not alter the topology of the picture (Morphenthaler, 1981). Sequential thinning algorithms delete simple points which are not end points, since preserving end-points provides important information relative to the shape of the objects. Curve thinning (i.e., a thinning process for extracting

* Tel.: +36-62-420-184; fax: +36-62-420-292.

E-mail address: palagyi@inf.u-szeged.hu (K. Palágyi).

medial line) preserves line end-points while surface thinning (i.e., a thinning process for extracting medial surface) does not delete surface end-points.

Parallel thinning algorithms delete a set of simple points. A possible approach to preserve topology is to use subiteration-based approach (Hall, 1996); the thinning operation is changed from iteration to iteration with a period of n ($n \geq 2$); each iteration of a period is then called a subiteration where only border points of certain kind can be deleted. There are six kinds of border points in 3D pictures on cubic grid, therefore, 6-subiteration thinning algorithms were generally proposed.

In this paper, a 3-subiteration algorithm is proposed for surface thinning. Some experiments are made on synthetic objects and the topology preservation for $(26, 6)$ binary pictures (Kong and Rosenfeld, 1989) is proved. Our approach demonstrates a possible way for constructing non-conventional subiteration-based thinning algorithms.

2. Basic notions and results

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 j -adjacent to point p (see Fig. 1). The sequence of distinct points $\langle x_0, x_1, \dots, x_n \rangle$ is a j -path of length $n \geq 0$ from point x_0 to point x_n in a nonempty set of points X if each point of the sequence is in X

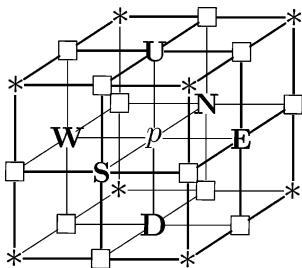


Fig. 1. The frequently used adjacencies in \mathbb{Z}^3 . The set $N_6(p)$ contains the central point p and the six points marked U, D, N, E, S and W. The set $N_{18}(p)$ contains the set $N_6(p)$ and the 12 points marked “□”. The set $N_{26}(p)$ contains the set $N_{18}(p)$ and the eight points marked “*”.

and x_i is j -adjacent to x_{i-1} for each $1 \leq i \leq n$. (Note that a single point is a j -path of length 0.) Two points are 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 P is a quadruple $P = (\mathbb{Z}^3, m, n, B)$ (Kong and Rosenfeld, 1989). Each element of \mathbb{Z}^3 is called a point of 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 (or object) 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 border point if it is 6-adjacent to at least one white point. (Note that this definition is correct only for the special cases $m = 26$ and $m = 18$.) A border point p is called U-border point if the point marked by U in Fig. 1 is white. We can define N-, E-, S-, W- and D-border points in the same way.

A black point is called simple point if its deletion does not alter the topology of the picture (Morphenthaler, 1981). We make use of the following result for $(26, 6)$ pictures.

Theorem 1 (Malandain and Bertrand, 1992; Saha and Chaudhuri, 1994). *Black point p is simple in picture $(\mathbb{Z}^3, 26, 6, B)$ if and only if all of the following 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 it is 6-connected in the set $(\mathbb{Z}^3 \setminus B) \cap N_{18}(p)$.

Theorem 1 shows that the simplicity in $(26, 6)$ pictures is a local property; it can be decided in view of the $3 \times 3 \times 3$ neighbourhood of a given point.

We need to consider what is meant by topology preservation when a number of black points are deleted simultaneously. Ma (1994) and Kong

(1995) gave sufficient conditions for parallel reduction operations of 3D $(26, 6)$ pictures. We use the following, much simpler sufficient conditions:

Theorem 2 (Palágyi and Kuba, 1999a,b). *Let F be a parallel reduction operation on $(26, 6)$ pictures. Then F is topology preserving, if for all pictures $P = (\mathbb{Z}^3, 26, 6, B)$, all of the following conditions hold:*

1. *for all points $p \in B$ that are deleted by F and for all sets $Q \subseteq (N_{18}(p) \setminus \{p\}) \cap B$ that are deleted by F , p is simple in the picture $(\mathbb{Z}^3, 26, 6, B \setminus Q)$;*
2. *no black component contained entirely in a unit lattice cube (i.e., a $2 \times 2 \times 2$ configuration in \mathbb{Z}^3) can be deleted completely by F .*

If an operation satisfies Theorem 2, then it obeys the sufficient conditions stated by Ma (1994) and Kong (1995).

We propose a surface thinning algorithm. The deletable points of the algorithm are border points of certain types and not *surface end-points* (i.e., which are not extremities of surfaces). The proposed algorithm uses the following characterization of the the surface end-points.

Definition 3 (Palágyi and Kuba, 1999a,b). The set $N_6(p)$ is subdivided into three kinds of opposite pairs of points (U, D), (N, S) and (E, W) (see Fig. 1).

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.

3. Existing parallel thinning algorithms

Most of the existing thinning algorithms are parallel as the fire front propagation is by nature parallel. These algorithms delete a set of simple points simultaneously that can alter the topology. There are three major strategies to overcome this problem (Hall, 1996):

Fully parallel algorithms: Algorithms from this group do not divide the thinning operation into different kinds of subiterations. In order to preserve topology, the known three fully parallel 3D

thinning algorithms investigate larger neighbourhood than the $3 \times 3 \times 3$ one: Ma (1995) proposed an algorithm, in which the new value of a black point depends on 30 points (and a parallel re-checking pass is required), the fully parallel algorithm of Ma and Sonka (1996) uses a special neighbourhood containing 50 points and Manzanera et al. (1999) developed an algorithm using a symmetric neighbourhood consisting of 81 points.

Subiteration-based algorithms: In the subiteration-based approach (often referred to as directional or border sequential strategy) the thinning operation is changed from iteration to iteration with a period of n ($n \geq 2$); each iteration of a period is then called a subiteration where only border points of certain kind can be deleted simultaneously. Since there are six kinds of major directions in 3D pictures, 6-subiteration thinning algorithms were generally proposed (Bertrand, 1995; Gong and Bertrand, 1990; Lee et al., 1994; Mukherjee et al., 1990; Palágyi and Kuba, 1998a; Tsao and Fu, 1981). Note that 8-subiteration (Palágyi and Kuba, 1999a) and 12-subiteration (Palágyi and Kuba, 1999b) algorithms have been developed, too. Each existing subiteration-based algorithm examines the $3 \times 3 \times 3$ neighbourhood of each border point.

Subfield-based algorithms: The 3D digital space \mathbb{Z}^3 is partitioned into several (disjoint) subfields that are alternatively activated (Golay, 1969; Preston, 1971). At a given iteration step, only border points in the active subfield are designated to be deleted. Each subiteration is executed in parallel (i.e., all border points in the actual subfield satisfying the deletion condition are simultaneously deleted). Two subfield sequential 3D thinning algorithms working in cubic grid have been proposed so far (Bertrand and Aktouf, 1994; Saha et al., 1997). Both algorithms investigate the $3 \times 3 \times 3$ neighbourhood and use eight subfields, therefore, each iteration step contains eight successive subiterations. Palágyi and Kuba (1998b) proposed a hybrid thinning algorithm. It uses both subfield and subiteration approaches (with two subfields and eight deletion directions). Note that Hall and Küçük (1992) developed a 3D shrinking algorithm that also uses a mix of subfield and subiteration methods.

The algorithm proposed in this paper follows the subiteration strategy. It requires only three subiterations (corresponding to the three kinds of opposite pair of points) in each iteration step, but an additional point not in the $3 \times 3 \times 3$ neighbourhood is examined in each subiteration.

4. The new thinning algorithm

In this section, a new algorithm is presented for extracting medial surfaces from 3D (26,6) pictures.

Each conventional 6-subiteration 3D thinning algorithm uses the six deletion directions that can delete certain U-, D-, N-, E-, S- and W-border points, respectively (Bertrand, 1995; Gong and Bertrand, 1990; Lee et al., 1994; Mukherjee et al., 1990; Palágyi and Kuba, 1998a; Tsao and Fu, 1981). In our 3-subiteration approach, two kinds of border points can be deleted in each subiteration. The three deletion directions correspond to the three kinds of opposite pairs of points, and are denoted by UD, NS and EW.

Suppose that the 3D (26,6) picture to be thinned contains finitely many back points. Reduction operations associated with the three subiterations are called `deletion_from_UD`, `deletion_from_NS` and `deletion_from_EW`.

We are now ready to present the 3-subiteration approach formally:

Input: picture $P = (\mathbb{Z}^3, 26, 6, B)$

Output: picture $P' = (\mathbb{Z}^3, 26, 6, B')$

`3-subiteration_thinning(B, B')`

```

begin
  B' = B;
  repeat
    B' = deletion_from_UD(B');
    B' = deletion_from_NS(B');
    B' = deletion_from_EW(B');
  until no points are deleted;
end.

```

The new value of a black point depends on 28 points in each subiteration. The three special neighbourhoods assigned to the different subiterations are presented in Fig. 2.

Deletable points in a subiteration are given by a set of 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 (see Fig. 3).

The first subiteration assigned to the deletion direction UD can delete certain U- or D-border points; the second subiteration associated with the deletion direction NS attempt to delete N- or S-border points, and some E- or W-border points can be deleted by the third subiteration corresponding to the deletion direction EW. The set of templates T_{UD} is given by Fig. 3. Note that Fig. 3 shows only the eight base templates T1–T8. Additionally, all their rotations around the vertical axis belong to T_{UD} , where the rotation angles are 90° , 180° and 270° . It is easy to see that the complete T_{UD} contains $2 \cdot (1 + 4 + 4 + 4) = 26$ tem-

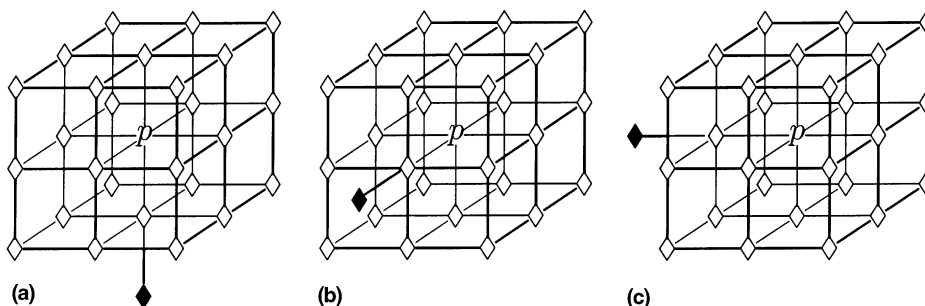


Fig. 2. The special local neighbourhoods assigned to the deletion directions UD (a), NS (b) and EW (c), respectively. The new value of a black point p depends on the $3 \times 3 \times 3$ neighbourhood of p (marked “◆”) and an additional point (marked “◆”) that is not in $N_{26}(p)$.

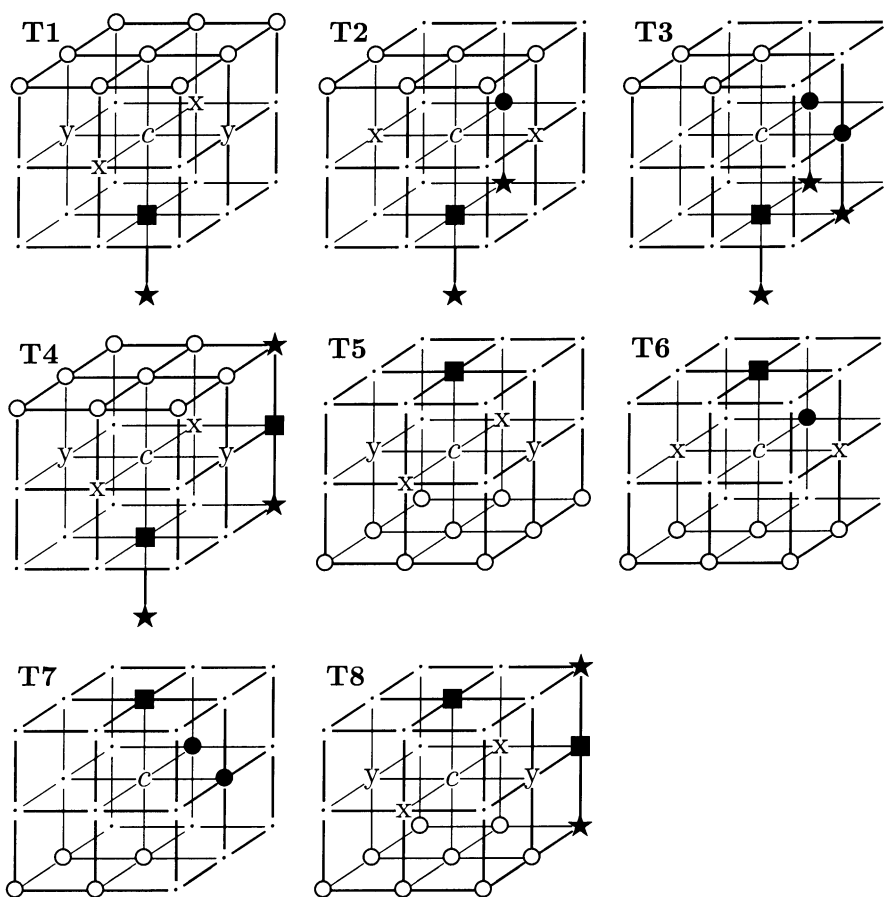


Fig. 3. Base templates T1–T8 and their rotations around the vertical axis form the set of templates T_{UD} assigned to the deletion direction UD. This set of templates belongs to the first subiteration. Notations: each position marked “c”, “●”, “■” and “★” matches a black point; each position marked “o” matches a white point; each “.” (“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. Emphasis is to be put that “x” and “y” positions provide that surface end-points cannot be deleted. (Note that using different symbols for black template positions helps us to prove the topological correctness of the algorithm.)

plates. This set of templates was constructed for deleting some simple points which are neither surface end-points (see Definition 1) nor extremities of surfaces. The deletable points of the other two subiterations (corresponding to deletion directions NS and EW) can be obtained by proper rotations of the templates in T_{UD} . Each template of our algorithm can be given by a Boolean condition that makes easy implementation possible.

Note that choosing another order of the deletion directions yields another algorithm. The proposed algorithm terminates when there are no

more black points to be deleted. Since all considered input pictures are finite, it will terminate.

5. Discussion

There are some 3D thinning algorithms that give their deletable points by sets of templates (or formulae) (Gong and Bertrand, 1990; Ma, 1995; Ma and Sonka, 1996; Manzanera et al., 1999; Mukherjee et al., 1990; Palágyi and Kuba, 1998a; Palágyi and Kuba, 1998b; Palágyi and Kuba,

1999a; Palágyi and Kuba, 1999b; Saha et al., 1997; Tsao and Fu, 1981). A number of configurations around a point are classified as deletable, but they form only a proper subset of simple points of certain kinds. 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);
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 6. It is based on the properties of the templates in T_{UD} (see Proposition 4).

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 (Bertrand and Aktouf, 1994) of an object: the thinning differs from shrinking. This is the reason why end-point criteria are used in thinning. We propose the end-point characterization given by Definition 3. According to this requirement, the end points are removed by none of our templates.

Geometrical correctness of the extracted skeleton is mostly achieved by the subiteration (multi-directional) thinning approach. An object is to be shrunk uniformly from each direction.

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 a lot of objects containing various segments. For example, Gerig et al. (1993) demonstrated that the curve thinning algorithm proposed by Tsao and Fu (1981) does not provide maximally thinned “skeleton”. However, base templates T4 and T8 in T_{UD} can match relatively few configurations; they are fairly important in thinning some kinds of segments. Our algorithm has been tested on objects of different shapes. Here we present five examples (see Figs. 4–8).

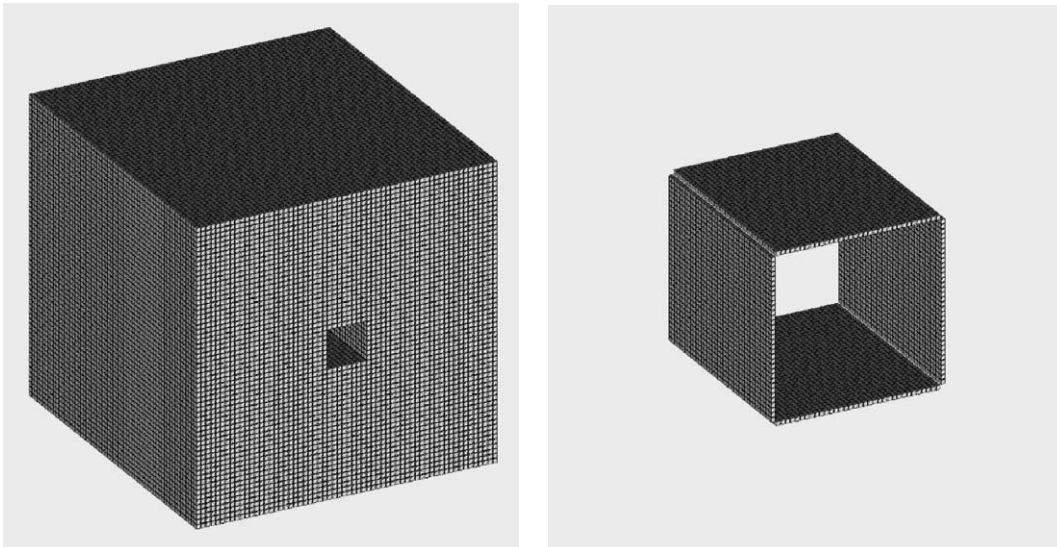


Fig. 4. The synthetic picture #1 containing a $64 \times 64 \times 64$ cube with a hole (left) and its medial surface produced by the proposed algorithm (right).

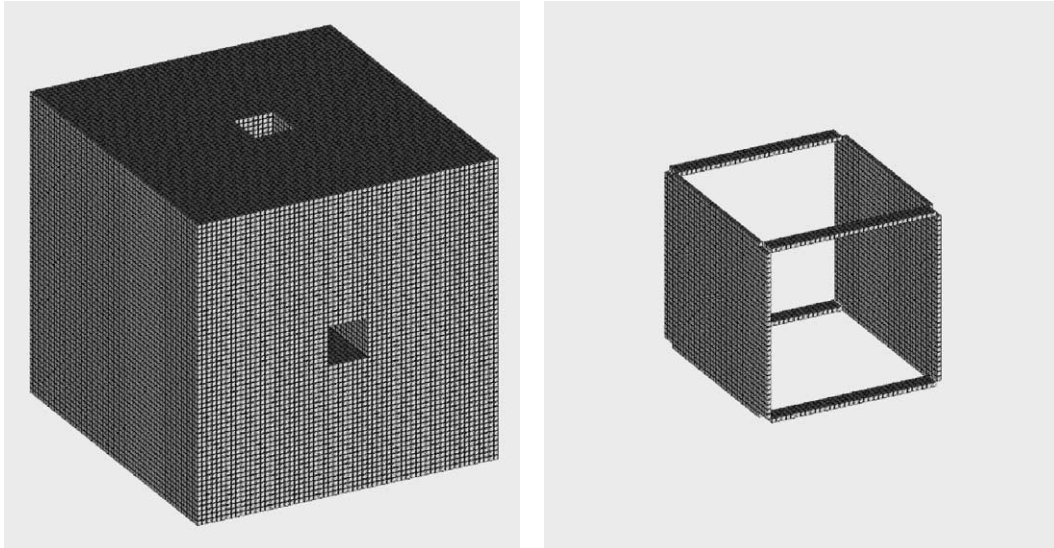


Fig. 5. The synthetic picture #2 containing a $64 \times 64 \times 64$ cube with two holes (left) and its medial surface produced by the proposed algorithm (right).

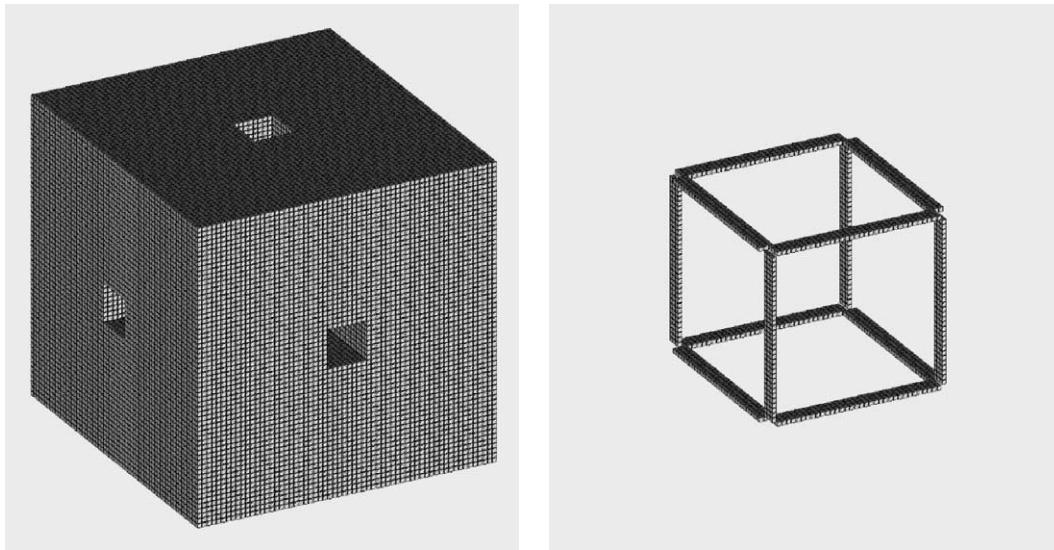


Fig. 6. The synthetic picture #3 containing a $64 \times 64 \times 64$ cube with three holes (left) and its medial surface produced by the proposed algorithm (right).

Thinning-based skeletonization concentrates on the topological correctness. Unfortunately, the distance induced by the investigated local neighbourhood cannot be regarded as a good approximation to the Euclidean distance. Therefore,

thinning algorithms are neither invariant under rotation (assuming arbitrary rotation angles) nor scale change. It can be stated that there is no thinning algorithm capable of preserving the sharpness of any corners (if a local neighbourhood

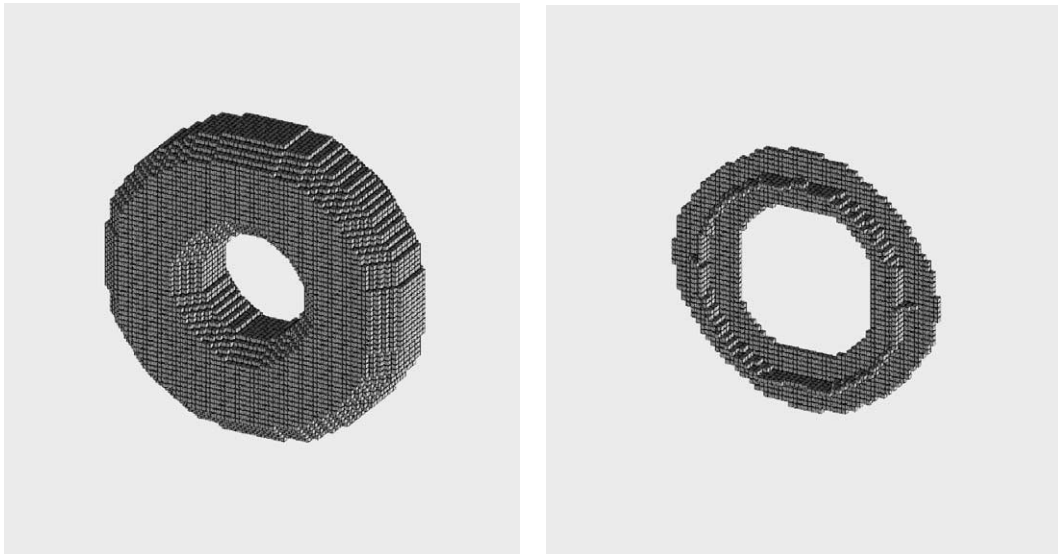


Fig. 7. The synthetic picture #4 (of size $64 \times 64 \times 20$) containing a doughnut (left) and its medial surface produced by the proposed algorithm (right).

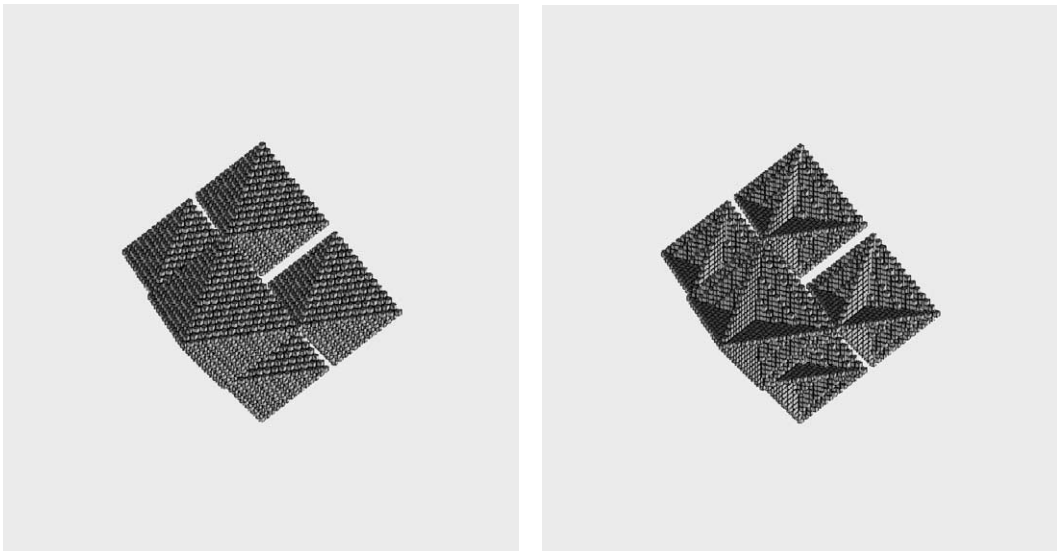


Fig. 8. The synthetic picture #5 (of size $64 \times 64 \times 32$) containing five connected pyramids (left) and its medial surface produced by the proposed algorithm (right).

of border points is investigated). Skeletonization methodology based on Euclidean distance mapping (Borgefors, 1984) 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 “skeletons”. There are some promising 3D skeletonization algorithms using both the approaches (Borgefors et al., 1999;

Pudney, 1998; Saito and Toriwaki, 1994; Svensson et al., 1999; Borgfors et al., 1999).

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 (that are not end points) may create spurious skeletal parts. 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.

The proposed algorithm makes easy implementation possible. The dependence for the first subiteration can be given by a Boolean function of 28 variables $F : \{0, 1\}^{28} \rightarrow \{0, 1\}$. We refer to the individual variables by assigning indices to them as illustrated in Fig. 9(a). Boolean function F can be expressed in the following form:

$$x'_1 = F(x_1, x_2, \dots, x_{28}) = x_1 \wedge \bigvee_{i=1}^{26} g_i(x_2, x_3, \dots, x_{27}),$$

where Boolean functions of 27 variables g_i ($i = 1, \dots, 26$) correspond to the 26 templates in T_{UD} . For example, the conjunctive formula of g_1 assigned to the base template T1 is presented in Fig. 9. Since the templates belonging to other subiterations can be derived from rotations of templates in T_{UD} . Boolean functions assigned to the derived templates can be given by the function

$g_i(\Pi(x_2, \dots, x_{28}))$, where Π is the proper permutation of 27 variables.

It is easy to see that evaluation of the new value of a point takes less than 500 elementary Boolean operations including logical or (“ \vee ”), logical and (“ \wedge ”), and logical not (“ \neg ”). Optimization of the Boolean function evaluation could be achieved, for instance, with binary decision diagrams (Robert and Malandain, 1998). Mention is to be made that the optimization is not important if we use a sequential computer. In this case, Look-up tables (LUTs) can be generated to store deletable point functions.

The computation time of a thinning process depends on the complexity of an iteration step and the required number of iteration steps. Note that it is rather difficult to give any kind of estimation of the complexity, since the number of required iteration steps is “data-dependent”.

We have compared the proposed algorithm with other subiteration-based 3D surface thinning algorithms. The following five algorithms have been investigated:

- D3-P: the 3-subiteration curve thinning algorithm (presented in this paper).
- D6-GB: 6-subiteration surface thinning algorithm that was proposed by Gong and Bertrand (1990).
- D6-PK: modified version of the 6-subiteration curve thinning algorithm proposed by Palágyi and Kuba (1998a) for extracting medial faces.

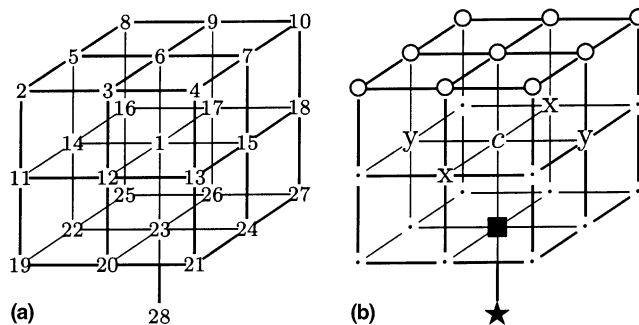


Fig. 9. The set of indices $\{1, 2, \dots, 28\}$ is assigned to the special local neighbourhood of the investigated point (a) and the base template T1 contained in T_{UD} (b). (Note that at least one position marked “x” matches a black point and at least one position marked “y” matches a black point.) The sample template can be described by the following formula: $g_1(x_2, \dots, x_{28}) = \bar{x}_2 \wedge \bar{x}_3 \wedge \bar{x}_4 \wedge \bar{x}_5 \wedge \bar{x}_6 \wedge \bar{x}_7 \wedge \bar{x}_8 \wedge \bar{x}_9 \wedge \bar{x}_{10} \wedge x_{23} \wedge x_{28} \wedge (x_{12} \vee x_{17}) \wedge (x_{14} \vee x_{15})$.

Table 1

The number of required iteration steps for the five test pictures and the investigated algorithms

Test pictures	Algorithms			
	D3-P	D6-GB, D6-PK	D8-PK	D12-PK
#1, #2, #3	42	84	48	84
#4	21	42	16	24
#5	12	24	24	24

- D8-PK: 8-subiteration surface thinning algorithm that was proposed by Palágyi and Kuba (1999a).
- D12-PK: 12-subiteration surface thinning algorithm that was proposed by Palágyi and Kuba (1999b).

The number of required iteration steps is given in Table 1 for the five test pictures (see Figs. 4–8) and the investigated algorithms. We can state that the proposed 3-subiteration algorithm generally requires the least number of iterations. If we assume unit time access LUTs, then the presented algorithm (D3-P) may be the fastest subiteration-based one.

Eight iterations of the 8-subiteration algorithm and 12 iterations of the 12-subiteration algorithm produce an $(n-8) \times (n-8) \times (n-8)$ cube from an $n \times n \times n$ one) then three iterations of the 3-subiteration algorithm and six iterations of the 6-subiteration algorithm can remove only the outmost layer from an object (one period of subiterations of these algorithm produces an $(n-2) \times (n-2) \times (n-2)$ cube from an $n \times n \times n$ one).

6. Verification

The proposed 3-subiteration thinning algorithm is topology preserving for $(26, 6)$ pictures. It is sufficient to prove that reduction operation given by the set of templates T_{UD} is topology preserving. If the first subiteration of the algorithm is topology preserving, then the other two are topology preserving, too, since the applied rotations of the deletion templates do not alter the topological properties. Therefore, the entire algorithm is topology preserving, since it is composed of topology-preserving reductions.

In order to prove both conditions of Theorem 2, we classify the elements of templates and state some properties of the set of templates T_{UD} . The element in the very centre of a template is called *central* (marked by “c” in Fig. 3). A noncentral template element is called *black* if it is always black (marked by “●”, “■” and “★” in Fig. 3). A noncentral template element is called *white* if it is always white (marked by “○” in Fig. 3). Any other noncentral template element which is neither white nor black is called *potentially black* (marked by “x”, “y” and “.” in Fig. 3). 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 T_{UD} ; p is *nondeletable* otherwise.

Proposition 4. *Let us consider the first subiteration and examine the configurations illustrated in Fig. 10.*

1. *Black point p in configuration (a) is nondeletable.*
2. *Black point p in configuration (b) is deletable if*
 - $q = 0, r = 1$ and $s = 1, 0$,
 - $q = 1$ and $r = 0$.
3. *Black point p in configuration (c) is deletable if $q = 1$ and $r = 1$.*
4. *Black point p in configuration (d) is deletable if $q = 1$.*

The topological correctness of the first subiteration of the proposed algorithm is stated by the following theorem.

Theorem 5. *Reduction operation given by the set of templates T_{UD} is topology preserving for $(26, 6)$ pictures.*

Proof. It is easy to see that each template in T_{UD} deletes only simple points of $(26, 6)$ pictures.

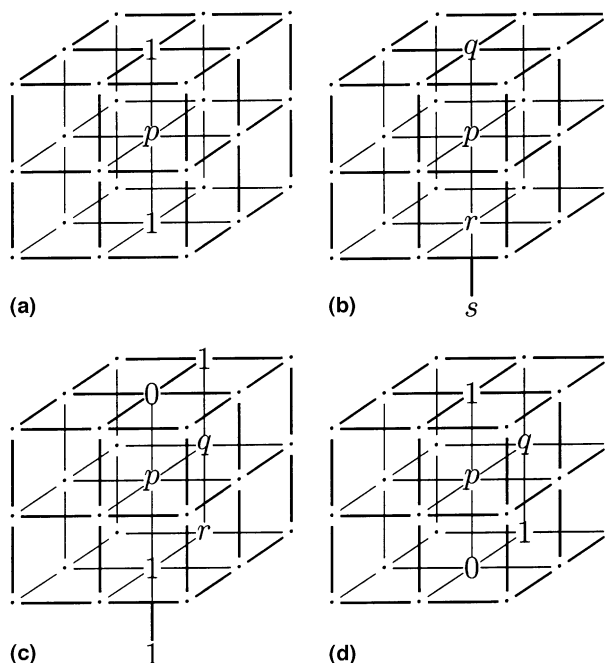


Fig. 10. Configurations assigned to Proposition 4. Note that Proposition 4 holds for their rotations around the vertical axis, too. (The rotation angles are 90° , 180° and 270° .)

The first point is to verify that there exists a 26-path between any two nonwhite positions (condition 1 of Theorem 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. It is obvious by careful examination of the templates in T_{UD} .

To prove that condition 2 of Theorem 1 holds, it is sufficient to show for each template in T_{UD} that:

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

The three points are obvious by a careful examination of the templates in T_{UD} .

We know that each deletable point p is simple. It can be stated that the value of any point coin-

cidng with a potentially black template position does not alter the simplicity of p . We can state that the simplicity of a point p does not depend on the points that coincide with a template position marked “★”, “x” or “y” (see Fig. 3). In addition, black points that coincide with template positions marked “■” are nondeletable (by Proposition 4(1)). Therefore, it is sufficient to deal with deletable points that coincide with template positions marked “●”. Note that base templates T1, T4, T5 and T8 (and their rotated versions) do not contain any positions marked “●”. Therefore, only base templates T2, T3, T6 and T7 (and their rotated versions) are to be investigated. It is easy to see with the help of Proposition 4 that deletion of points coinciding with template positions marked “●” does not alter the simplicity of point p . Therefore, Condition 1 of Theorem 2 is satisfied.

Condition 2 of Theorem 2 can be seen with the help of Proposition 4(2) too. Let us consider a unit lattice cube containing an upper set of four points $U = \{u_1, u_2, u_3, u_4\}$ and a lower set of four points $L = \{l_1, l_2, l_3, l_4\}$. Let $C \subseteq U \cup L$ be a black

component contained in the unit lattice cube. If $C \cap L$ contains a deletable point then $C \cap U \neq \emptyset$ by Proposition 4(2). It is easy to see that any point in $C \cap U$ is nondeletable by Proposition 4(2). Therefore, black component C cannot be deleted completely. \square

Acknowledgements

The author is grateful to Richard W. Hall, Attila Kuba, and László G. Nyúl for their valuable suggestions and remarks. This work was supported by the FKFP 0908/1997 Grant.

References

- Bertrand, G., Aktouf, Z., 1994. A 3D thinning algorithms using subfields. In: Proc. SPIE Conf. on Vision Geometry III, Vol. 2356, pp. 113–124.
- Bertrand, G., 1995. A parallel thinning algorithm for medial surfaces. *Pattern Recognition Lett.* 16, 979–986.
- Blum, H., 1967. A transformation for extracting new descriptors of shape. In: *Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, MA, pp. 362–380.
- Borgefors, G., 1984. Distance transformations in arbitrary dimensions. *Comput. Vision, Graphics, Image Process.* 27, 321–345.
- Borgefors, G., Nyström, I., Sanniti di Baja, G., 1999. Computing skeletons in three dimensions. *Pattern Recognition* 32, 1225–1236.
- Gerig, G., Koller, Th., Székely, G., Brechbühler, Ch., Kübler, O., 1993. Symbolic description of 3-D structures applied to cerebral vessel tree obtained from MR angiography volume data. In: Proc. 13th Internat. Conf. Information Processing in Medical Imaging, IPMI'93. Lecture Notes in Computer Science, Vol. 687. Springer, Berlin, pp. 94–111.
- Golay, M.J.E., 1969. Hexagonal parallel pattern transformations. *IEEE Trans. Comput.* 18, 733–740.
- Gong, W.X., Bertrand, G., 1990. A simple parallel 3D thinning algorithm. In: Proc. 10th Internat. Conf. on Pattern Recognition, pp. 188–190.
- Hall, R.W., Küçük, Ş., 1992. Parallel 3D shrinking algorithms using subfields notions. In: Proc. 11th IEEE Int. Conf. Pattern Recognition, pp. 395–398.
- Hall, R.W., 1996. Parallel connectivity-preserving thinning algorithms. In: Kong, T.Y., Rosenfeld, A. (Eds.), *Topological Algorithms for Digital Image Processing*. Elsevier, Amsterdam, pp. 145–179.
- Kong, T.Y., 1995. On topology preservation in 2-D and 3-D thinning. *Int. J. Pattern Recognition Artificial Intell.* 9, 813–844.
- Kong, T.Y., Rosenfeld, A., 1989. Digital topology: introduction and survey. *Comput. Vision, Graphics, Image Process.* 48, 357–393.
- Lee, T., Kashyap, R.L., Chu, C., 1994. Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graphical Models Image Process.* 56, 462–478.
- Ma, C.M., 1994. On topology preservation in 3D thinning. *CVGIP: Image Understanding* 59, 328–339.
- Ma, C.M., 1995. A 3D fully parallel thinning algorithm for generating medial faces. *Pattern Recognition Lett.* 16, 83–87.
- Ma, C.M., Sonka, M., 1996. A fully parallel 3D thinning algorithm and its applications. *Comput. Vision Image Understanding* 64, 420–433.
- Manzanera, A., Bernard, T.M., Pretêux, F., Longuet, B., 1999. Medial faces from a concise 3D thinning algorithm. In: Proc. 7th IEEE Internat. Conf. Computer Vision, ICCV'99, pp. 337–343.
- Malandain, G., Bertrand, G., 1992. Fast characterization of 3D simple points. In: Proc. 11th IEEE Internat. Conf. on Pattern Recognition, pp. 232–235.
- Morgenthaler, D.G., 1981. Three-dimensional simple points: serial erosion, parallel thinning and skeletonization. Technical Report TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD.
- Mukherjee, J., Das, P.P., Chatterjee, B.N., 1990. On connectivity issues of ESPTA. *Pattern Recognition Lett.* 11, 643–648.
- Palágyi, K., Kuba, A., 1998a. A 3D 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recognition Lett.* 19, 613–627.
- Palágyi, K., Kuba, A., 1998b. A hybrid thinning algorithm for 3D medical images. *J. Comput. Inform. Technol. – CIT* 6, 149–164.
- Palágyi, K., Kuba, A., 1999a. Directional 3D thinning using 8 subiterations. In: Proc. 8th Internat. Conf. on Discrete Geometry for Computer Imagery, DGCI'99. Lecture Notes in Computer Science, Vol. 1568. Springer, Berlin, pp. 325–336.
- Palágyi, K., Kuba, A., 1999b. A parallel 3D 12-subiteration thinning algorithm. *Graphical Models Image Process.* 61, 199–221.
- Preston, K., 1971. Feature extraction by Golay hexagonal pattern transforms. *IEEE Trans. Comput.* 20, 1007–1014.
- Pudney, C., 1998. Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Comput. Vision Image Understanding* 72, 404–413.
- Robert, L., Malandain, G., 1998. Fast binary image processing using binary decision diagrams. *Comput. Vision Image Understanding* 72, 1–9.
- Saha, P.K., Chaudhuri, B.B., 1994. Detection of 3-D simple points for topology preserving transformations with application to thinning. *IEEE Trans. Pattern Anal. Machine Intell.* 16, 1028–1032.
- Saha, P.K., Chaudhuri, B.B., Majumder, D.D., 1997. A new shape-preserving parallel thinning algorithm for 3D digital images. *Pattern Recognition* 30, 1939–1955.

- Saito, T., Toriwaki, J., 1994. New algorithm for Euclidean distance transformation of an n -dimensional digitized picture with applications. *Pattern Recognition* 27, 1551–1565.
- Svensson, S., Nyström, I., Borgefors, G., 1999. Fully reversible skeletonization for volume images based on anchor-points from the D^{26} distance transform. In: Proc. 11th Scandinavian Conf. Image Analysis, SCIA'99, pp. 601–608.
- Tsao, Y.F., Fu, K.S., 1981. A parallel thinning algorithm for 3-D pictures. *Comput. Graphics Image Process.* 17, 315–331.