An improved fully parallel 3D thinning algorithm[1]

Tao Wang and Anup Basu

Department of Computing Science
University of Alberta
Edmonton, AB T6G 2E8, Canada
Email: {taowang, anup}@cs.ualberta.ca

Abstract

A 3D thinning algorithm erodes a 3D image layer by layer to extract the skeletons. This paper presents an improved fully parallel 3D thinning algorithm which extracts medial lines from a 3D image. This algorithm is based on Ma and Sonka's thinning algorithm [16], which fails to preserve connectivity of 3D objects. We start with Ma and Sonka's algorithm and examine its verification of connectivity preservation. Our analysis leads to a group of different deleting templates, which can preserve connectivity of 3D objects.

1. Introduction

Thinning is a useful technique having potential applications in a wide variety of problems. It creates a compact representation (*skeleton*) of the models that may be used for further processing. 3D Skeletons can be used in many applications [7-12] such as 3D pattern matching, 3D recognition and 3D database retrieval.

In 3D Euclidean space, the skeletons can be defined as the *medial axes* of the models. The medial axes are the loci of the centers of all inscribed maximal spheres of the model where these spheres share at least two points with the boundary of the model [35]. However, exact computation of the medial axes is non trivial and time-consuming [2-5].

[1] The support of an ICORE scholarship in making this work possible is gratefully acknowledged.

Researchers presented different methods, such as Voronoi diagram [4, 5] and template based thinning method [14-19], to get the approximate medial axes.

There are two types of medial axis in literature, *medial line* [14, 16, 19] and *medial surface* [15, 17, 18]. In many applications [7, 9], we prefer the medial line to medial surface because the medial line is more compact and easier to handle. Thus, this paper focuses on the medial line extraction.

Two kinds of 3D models are investigated in literature, *3D mesh* [2-6] and *3D image* [13-25]. A *3D mesh* is a piecewise linear surface. It consists of polygonal faces pasted together along the edges. Some algorithms [2-6] have been proposed to extract skeletons from 3D meshes. Algorithms in this category extract medial surfaces from the 3D mesh. However, the algorithms have some drawbacks. First, the medial surfaces are unstable. Small deformations in the boundary of the model can lead to large changes in the medial surfaces. Second, it is difficult to compute the medial surfaces because of the algebraic complexity, which also leads to complicated implementations. Third, sample points should be very dense to get the precise medial surfaces. Because of the aforementioned difficulties 3D mesh will not be used in this paper.

A *3D image* is a mapping that assigns the value of 0 or 1 to each point in the 3D space. Points having the value of 1 are called *black (object)* points, while 0's are called *white (background)* ones. Black points form objects of the image. The thinning operation iteratively *deletes* or *removes* some object points (that is, changes some black points to white) until only some restrictions prevent further operation. Note that the white points will never be changed to black ones in any circumstances. Algorithms [13-25] in this category can extract medial surfaces and/or medial lines. Only the medial line extraction method will be discussed in this paper. Most of the existing thinning algorithms are parallel, since the *medial axis transform* (MAT) can be defined as *fire front propagation*, which is by nature parallel [35]. There are three categories of parallel thinning algorithms in literature, *sub-iteration parallel* thinning algorithm [14, 15, 18], *sub-field parallel* thinning algorithm [24, 25] and *fully parallel* thinning algorithm [16]. Brief surveys of algorithms in each category can be found in the literature [16, 18].

The rest of this paper is organized as follows. In Section 2, some basic concepts will be presented. Section 3 will briefly discuss Ma and Sonka's algorithm [16]. The problematic part in this algorithm is analyzed and the modification is presented in Section 4, before the work is concluded in Section 5.

2. Basic concepts

We first describe some terms and notation:

Let p and q be two different points with coordinates (px , py , pz) and (qx , qy, qz), respectively, in a 3D image P. The Euclidean distance between p and q is defined as:

$$dis = \sqrt{(px - qx)^2 + (py - qy)^2 + (pz - qz)^2}$$

Then p and q are:

6-adjacent if $dis \leq 1$. *18-adjacent* if $dis \leq \sqrt{2}$. *26-adjacent* if $dis \leq \sqrt{3}$. Let us denote the set of points *k-adjacent* to point p by $N_k(p)$, where k = 6, 18, 26, (see Figure 1). $N_k(p)$ is also called p's *k-neighborhood*. Let p be a point in a 3D image. Then, $e(p)$, $w(p)$, $n(p)$, $s(p)$, $u(p)$, *and* $d(p)$ are *6-neighbors* of p, which represent 6 directions of *east*, *west*, *north*, *south*, *up*, and *down*, respectively. The *18-neighbors* of p (but not in p's *6-neighborhood*) are $nu(p)$, $nd(p)$, $ne(p)$, $nw(p)$, $su(p)$, $sd(p)$, $se(p)$, $sw(p)$, $wu(p)$, $wd(p)$, $eu(p)$, and $ed(p)$, which represent 12 directions of *north-up, north-down, north-east, north-west, south-up, south-down, south-east, south-west, west-up, west-down, east-up,* and *east-down*, respectively.

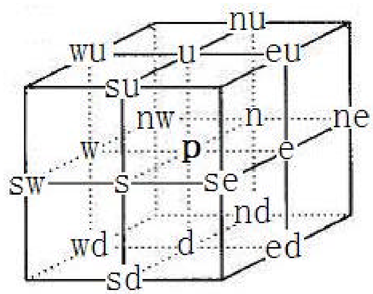


Figure 1: The adjacencies in a 3D image. Points in $N_6(p)$ are marked *u, n, e, s, w,* and *d.* Points in $N_{18}(p)$ but not in $N_6(p)$ are marked *nu, nd, ne, nw, su, sd, se, sw, wu, wd, eu,* and *ed.* The unmarked points are in $N_{26}(p)$ but not in $N_{18}(p)$.

It is very important for thinning algorithms to *preserve connectivity* for 3D objects [14, 16]. If a thinning algorithm fails to preserve connectivity, the skeletons extracted from the object will be disconnected, which is unacceptable in many applications. A sequential thinning algorithm can preserve connectivity easily if it is only allowed to delete *simple points* [14]. However, a parallel thinning algorithm may delete many black points in every iteration, even if it is only allowed to delete simple points, the algorithm may not preserve connectivity [16, 18]. This problem was investigated and the general results were proposed by Ma [30, 16].

3. Ma and sonka's algorithm

In 1996, Ma and Sonka proposed a fully 3D thinning algorithm [16], which was applied to many applications such as medical image processing [36] and 3D reconstruction [37].

The algorithm is based on some pre-defined templates (Class A, B, C and D). If the neighborhood of an object point matches one of the templates, it will be removed. Figure 2 shows the four basic template cores. In this figure, a "•" is used to denote an object point, a "◦" is used to denote a background point. An unmarked point is a "don't care" point, which can represent either an object point or a background point.

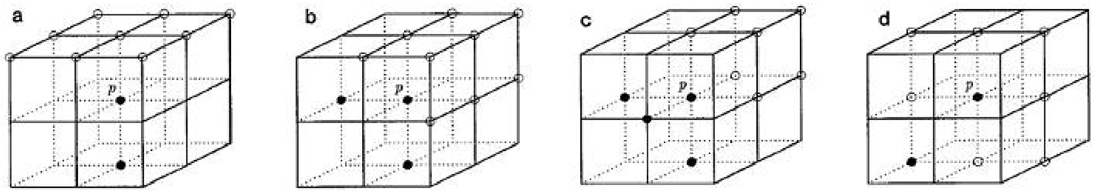


Figure 2: Four template cores (Class A, B, C and D) of the fully parallel thinning algorithm. A "•" is an object point. A "◦" is a background point. An unmarked point can be either an object point or a background point. For (d), *p* must be simple. Image adapted from Ma [16].

The template cores themselves are not the deleting templates. Some translations [16] must be applied to the template cores to generate the deleting templates. We can get 6 templates in Class A, 12 templates in Class B and 8 templates in Class C and 12 templates in Class D according to the translations. Templates in Class A-D are shown in Figure 3-Figure 6.

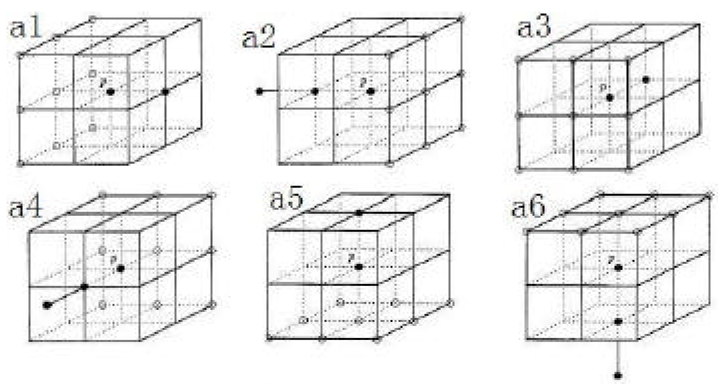


Figure 3: 6 deleting templates in Class A.

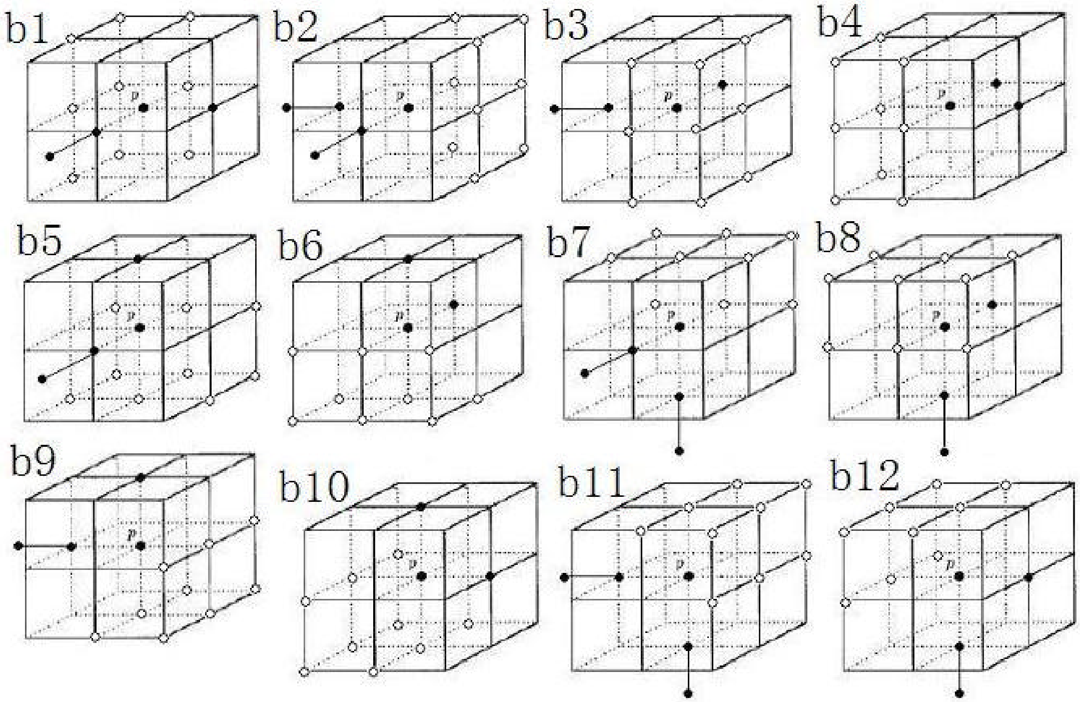


Figure 4: 12 deleting templates in Class B.

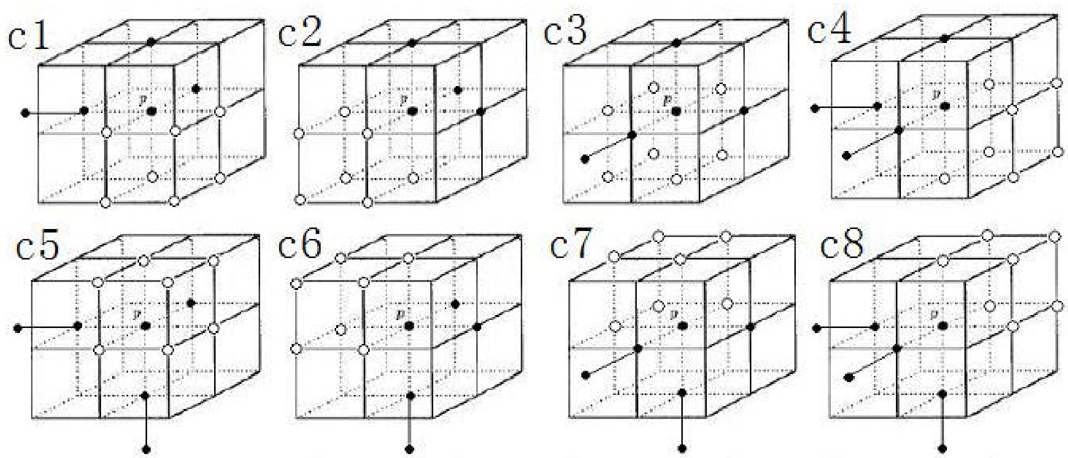


Figure 5: 8 deleting templates in Class C.

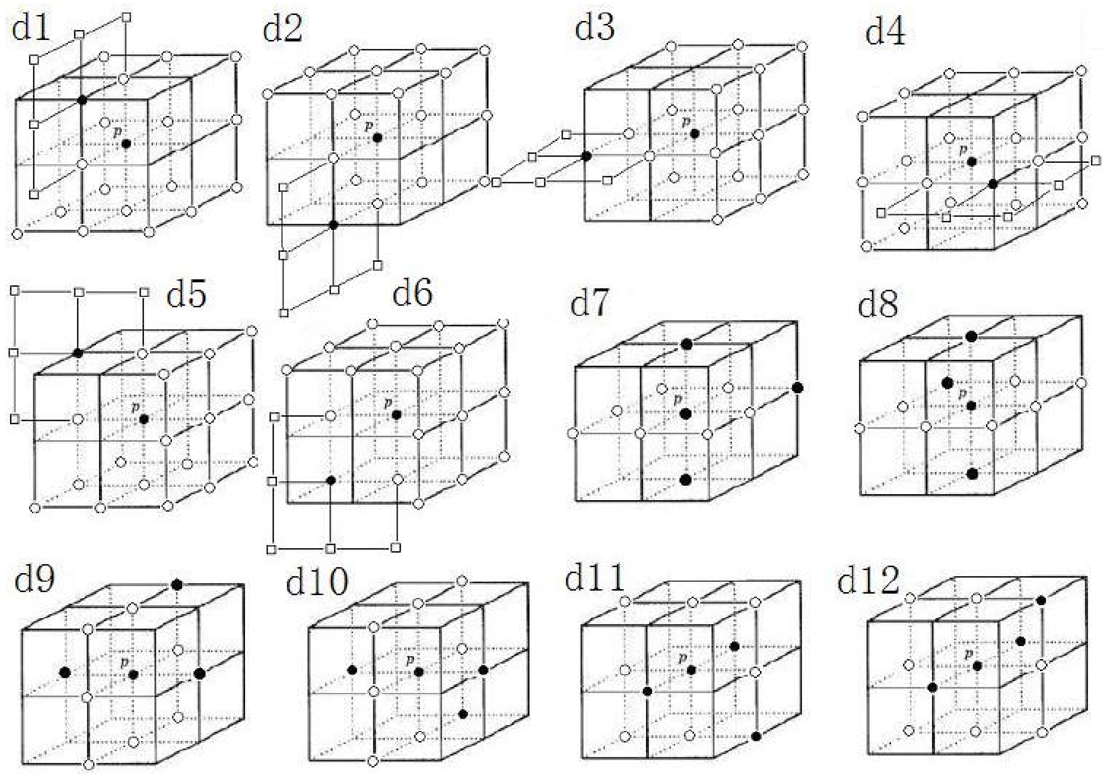


Figure 6: 12 deleting templates in Class D.

In each iteration, all *non tail-points* [16] satisfying at least one of the deleting templates in Class A, B, C or D are deleted in the fully parallel thinning algorithm as follows:

Algorithm

Repeat

1) Mark every object point which is 26-adjacent to a background point;

2) **Repeat**

Simultaneously delete every non tail-point which satisfies at least one deleting template in Class A, B, C, or D;

Until no point can be deleted;

3) Release all marked but not deleted points;

Until no marked point can be deleted;

4. The problem and a solution

A 3D parallel thinning algorithm should preserve connectivity. However, by studying the configuration in Fig 7, we find that the algorithm fails to preserve connectivity. In Fig 7, a "•" is an object point. All other points are background points, and it shows a 26-connected 3D object a-b-c-d-e-f-g. In Ma and Sonka's thinning algorithm, point *c*, *d* and *e* will be deleted because *c* satisfies template *a5* in Class A, *d* satisfies template *d7* in Class D and *e* satisfies template *a6* in Class A. However, the deletion of point *c*, *d* and *e* leads to disconnection of the object.

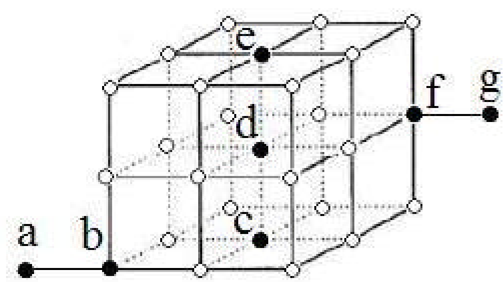


Figure 7: A connected object a-b-c-d-e-f-g in 3D space. A "•" is an object point. A "○" is a background point. All other points in 3D space are background points. In Ma and Sonka's algorithm, point *c* will be deleted by template *a5* in Class A, point *d* will be deleted by template *d7* in Class D and point *e* will be deleted by template *a6* in Class A. Hence, the object will be disconnected.

Ashutosh et al. [36] also found that this algorithm disconnected some small segments. But they did not find out why this algorithm fails to preserve connectivity and propose a solution to the problem. In this section, we will show the reason for the problem and how to modify the templates in Class D to preserve connectivity.

Ma and Sonka proposed a general theorem [16] and used it to prove that the 3D thinning algorithm preserves connectivity in the VERIFICATION section in that paper. According to our observation, we note that LEMMA 3.5 in the VERIFICATION section is problematic.

"**LEMMA 3.5**: *Let p, q be two 6-adjacent object points in a 3D image where both p and q satisfy* Ω. *Then either* $q \notin \Omega(p)$ *or* $p \notin \Omega(q)$."

Ω is used to denote the set of deleting templates in Class A, B, C or D. An object point satisfies Ω if it satisfies any one of the deleting templates in Ω. "$q \in \Omega(p)$" means that q must be an object point for p to satisfy Ω. "$q \notin \Omega(p)$" means p still satisfies Ω after q is deleted.

For the 3D object in Figure 7, c and d are two 6-adjacent points. According to LEMMA 3.5, either $c \notin \Omega(d)$ *or* $d \notin \Omega(c)$. However, if c is deleted, d will not satisfy any of the deleting templates. And if d is deleted, c will not satisfy any of the deleting templates. Therefore, although c and d are 6-adjacent, $c \in \Omega(d)$ and $d \in \Omega(c)$. We can prove that for points d and e, $d \in \Omega(e)$ and $e \in \Omega(d)$, although d and e are 6-adjacent, in the same way.

LEMMA 3.5 requires that for two 6-adjacent points p and q, if both p and q satisfy Ω, then either $q \notin \Omega(p)$ or $p \notin \Omega(q)$. Let $p1$ and $p2$ be the two "don't care" points in p's 6-neighborhood, as showed in Figure 8. According to LEMMA 3.5, if $p1$ is 1, then $p2$ must be 0; if $p2$ is 1 then $p1$ must be 0. So ($p1$, $p2$) can be (0, 0), (0, 1) or (1, 0), but not (1, 1). There is no template in Class A-C that has value of (1, 1) for ($p1$, $p2$), however, the deleting templates in Class D violate this rule. For instance, in template *d7*, ($p1$, $p2$) is (1, 1), which causes LEMMA 3.5 to fail.

Based on this observation, we can change deleting template *d7* to make LEMMA 3.5 satisfy. According to different values of ($p1$, $p2$), we change template *d7* to three new templates as shown in Figure 9.

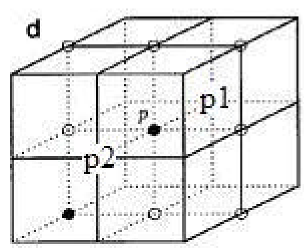


Figure 8: Template core of Class D.

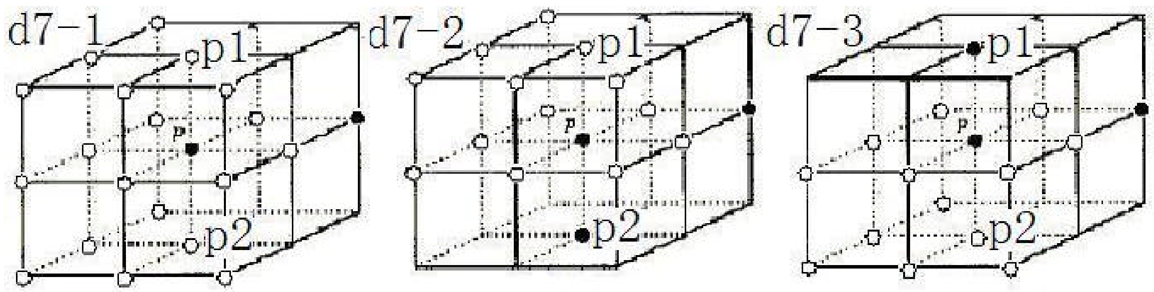
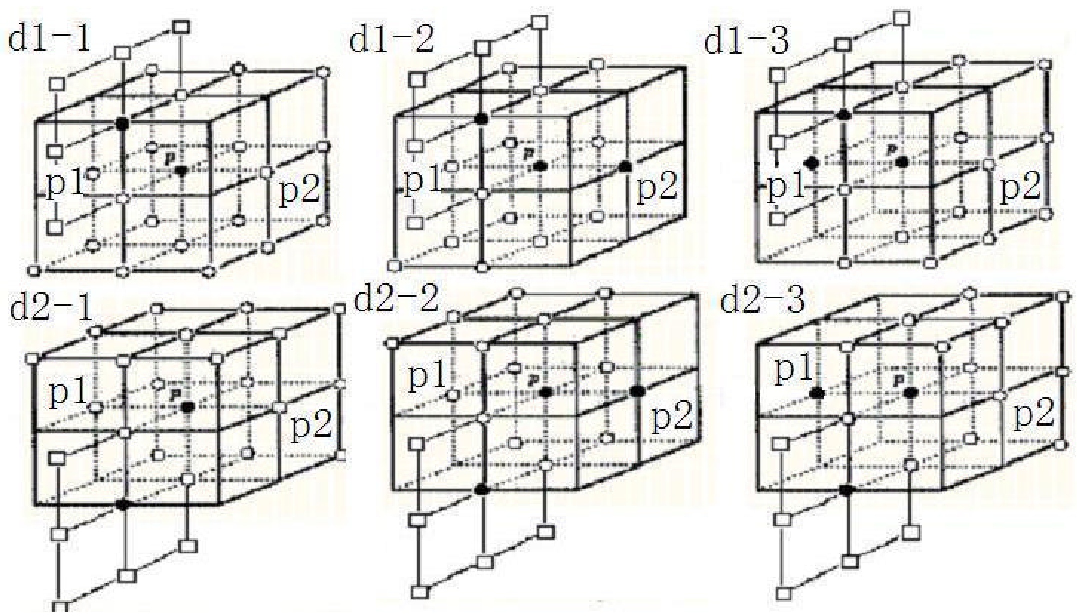
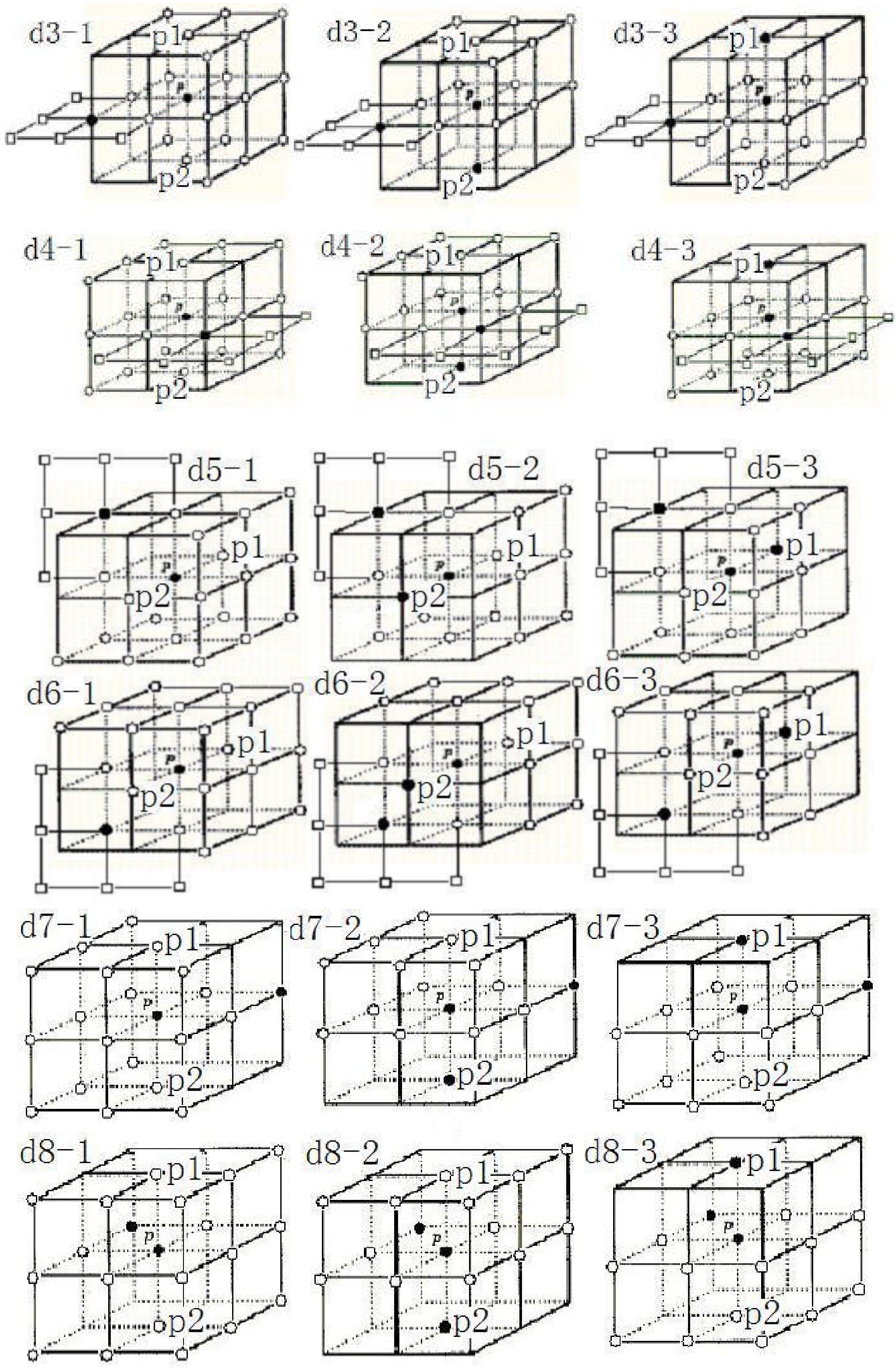


Figure 9: Template d7-1 to d7-3.

In this way, we change the 12 deleting templates of Class D to 36 deleting templates according to different values of (*p1, p2)*. Figure 10 shows the modified templates in Class D. Each template in Class D is changed to three templates, in which (*p1, p2*) are (0, 0), (0, 1) or (1, 0) respectively. Since (*p1, p2*) is not (1, 1) in any template, LEMMA 3.5 is satisfied for the new set of templates.




d3-1
p1
p
p2
d3-2
p1
p
p2
d3-3
p1
p
p2
d4-1
p1
p
p2
d4-2
p1
p
p2
d4-3
p1
p
p2
d5-1
p1
p
p2
d5-2
p1
p
p2
d5-3
p1
p
p2
d6-1
p1
p
p2
d6-2
p1
p
p2
d6-3
p1
p
p2
d7-1
p1
p
p2
d7-2
p1
p
p2
d7-3
p1
p
p2
d8-1
p1
p
p2
d8-2
p1
p
p2
d8-3
p1
p
p2

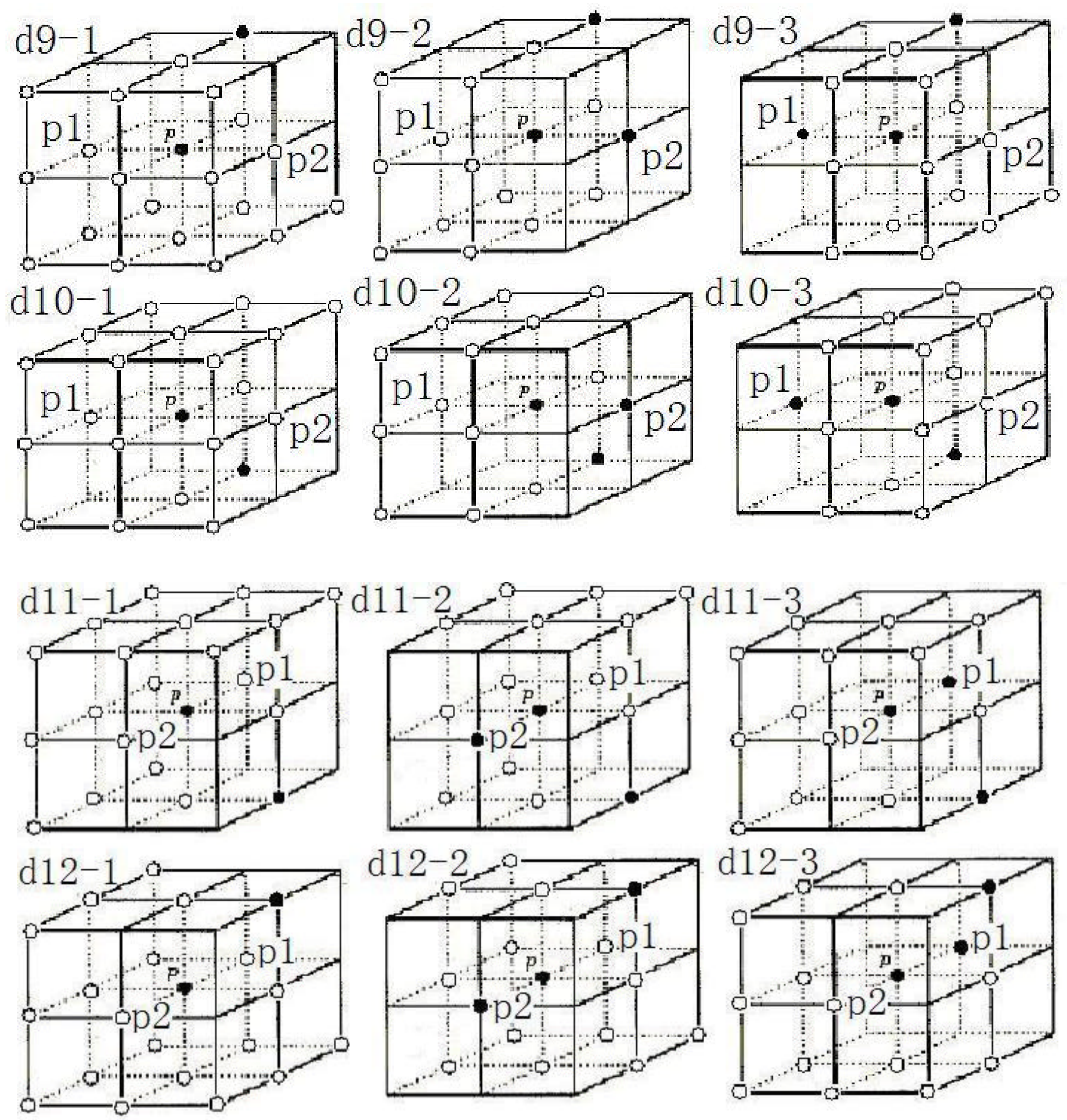
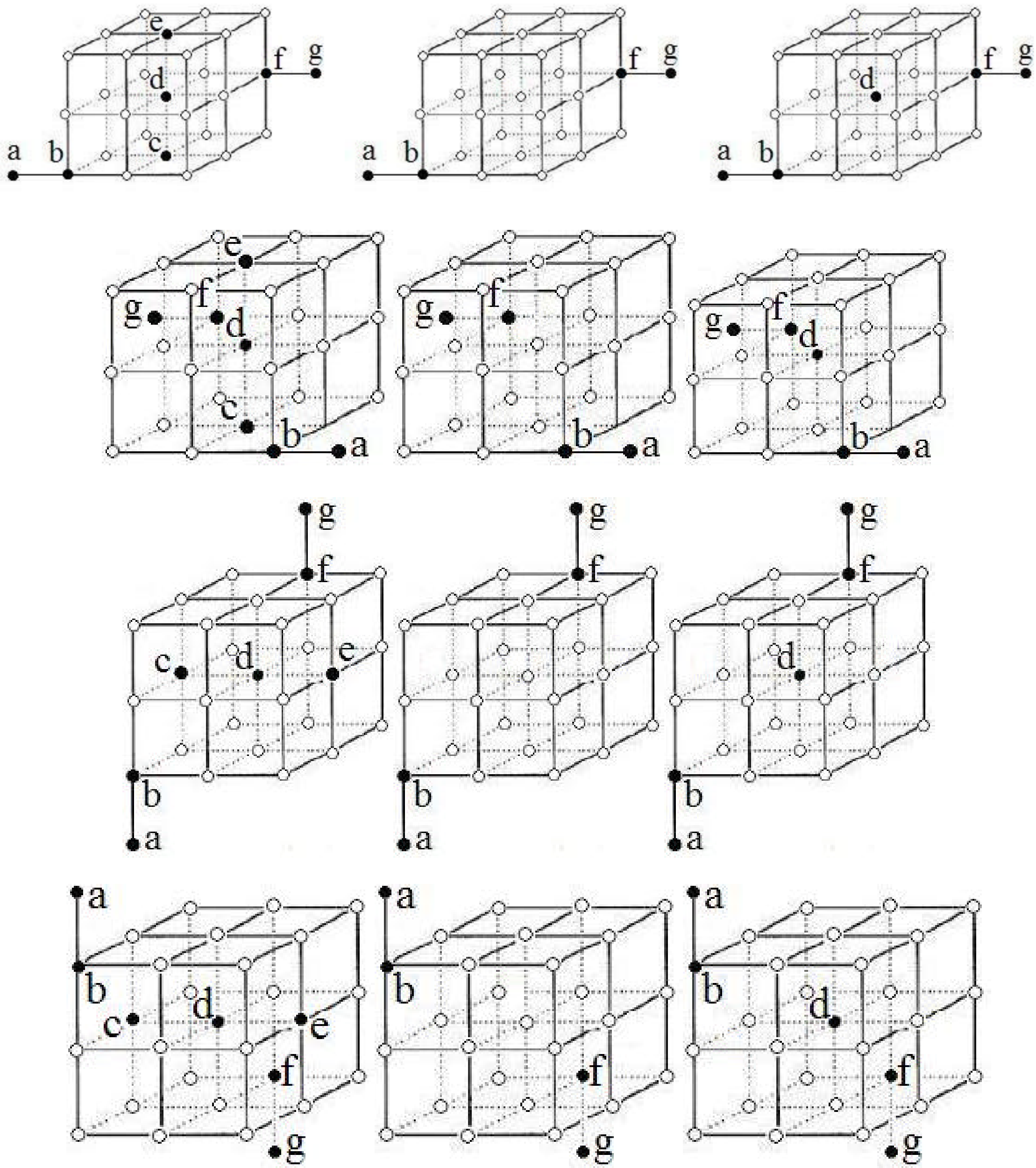
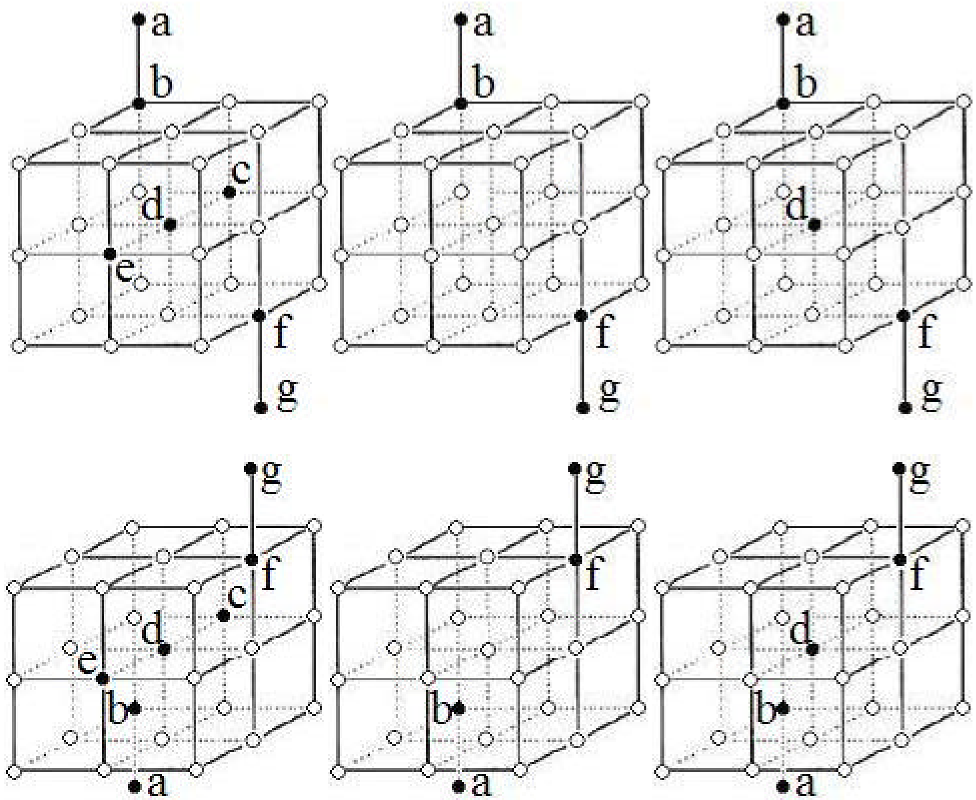


Figure 10: The modified deleting templates in Class D. Each template in Class D is changed to three templates, in which (*p1, p2*) are (0, 0), (0, 1) or (1, 0) respectively.

Figure 11 shows some different results of Ma and Sonka's algorithm and the modified one. Six 26-connected 3D objects are shown in (a). For Ma and Sonka's algorithm, point *c*, *d* and *e* are deleted, thus the connected object is disconnected after thinning, as shown in (b). For the modified algorithm, point *c* and *e* are deleted, but point *d* will not be deleted, thus connectivity is preserved after the thinning operations, as shown in (c).



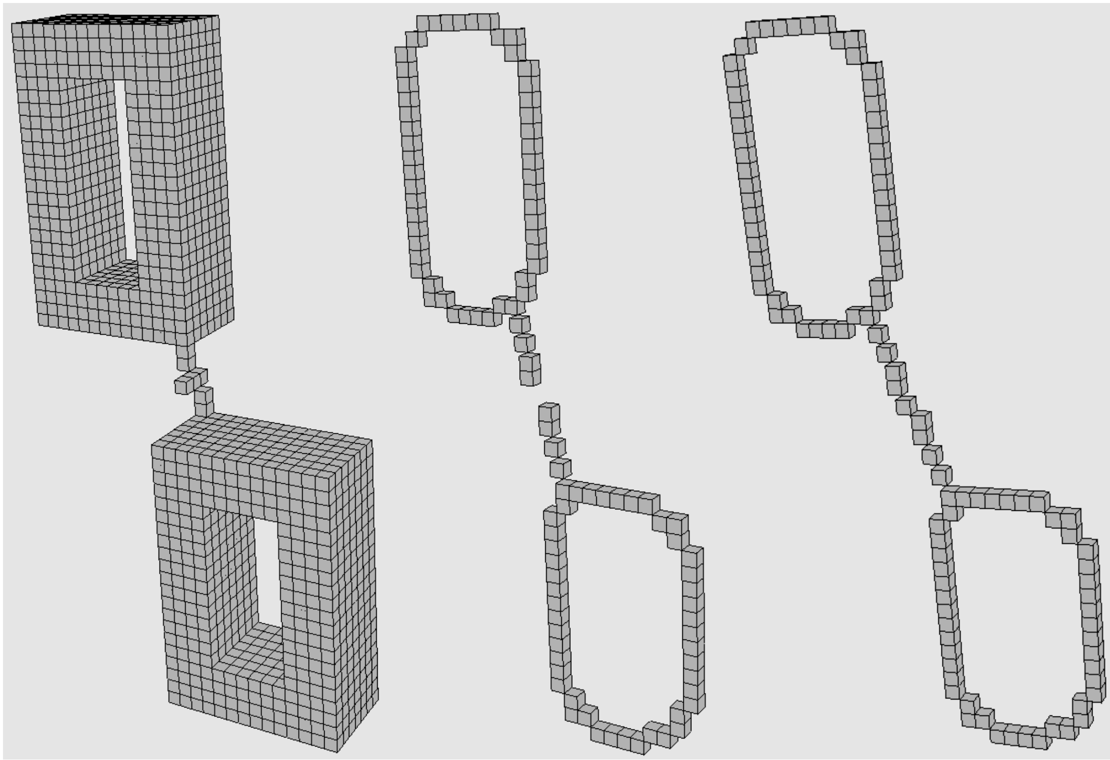
a b c d e f g
a b f g
a b d f g
a b c d e f g
a b f g
a b d f g
a b c d e f g
a b f g
a b d f g
a b c d e f g
a b f g
a b d f g



(a) (b) (c)

Figure 11: A "•" is an object point. A "∘" is a background point. All other points in 3D space are background points. (a) The original 3D object a-b-c-d-e-f-g. (b) The thinning result of Ma and Sonka's algorithm. Point *c*, *d* and *e* are deleted by some templates in Class A and Class D. Thus, the object gets disconnected. (c) The thinning result of the modified algorithm. Points *c* and *e* are deleted by some templates in Class A, but point *d* is not deleted, thus the object is still connected.

Figure 12 shows a different result of Ma and Sonka's algorithm and the modified one. Two "0" connected by a structure (as shown in Figure) are shown in (a). For Ma and Sonka's algorithm, the connected object is disconnected after thinning, as shown in (b). For the modified algorithm, the connected object is still connected after the thinning operations, as shown in (c).



(a) (b) (c)

Figure 12: (a) Original 3D object; (b) Result of Ma and Sonka's algorithm; (c) Result of modified algorithm.

Verification

The verification procedure is same as Ma and Sonka's algorithm. LEMMA 3.5 satisfies, therefore, the improved algorithm is connectivity preserving.

5. Conclusions

In this paper, an improved fully parallel 3D thinning algorithm is proposed. It is derived from Ma and Sonka's algorithm [16] and modifies some templates to preserve connectivity of 3D objects. The motivation behind this paper is that we found Ma and Sonka's algorithm failed to preserve connectivity, which is very important for 3D thinning algorithms. We then studied why MA and Sonka's algorithm failed and proposed a solution to the problem. Experimental results demonstrate the validity of our solution.

In future work, we will apply our algorithm in a variety of CT and MRI data processing applications.

Acknowledgements

The author thanks Kalman Palagyi for discussion on 3D thinning algorithms.

References

[1] R. C. Gonzalez, R. E. Woods, Digital Image Processing (second edition), Prentice-Hall Inc., New Jersey, USA, 2002.

[2] F. F. Leymarie, B. B. Kimia, Computation of the shock scaffold for unorganized point clouds 3d, IEEE Proc. of CVPR'03, Madison, Wisconsin, USA, June 2003, Vol. 1, pp. 821-827.

[3] F. F. Leymarie, Three-dimensional shape representation via shock flows, PhD thesis, Brown University, May 2003.

[4] T. K. Dey, W. Zhao, Approximating the medial axis from the Voronoi diagram with a convergence guarantee, Lecture Notes in Computer Science, Vol. 2461/2002, pp. 387-398.

[5] M. Foskey, M. C. Lin, D. Manocha, Efficient computation of a simplified medial axis, ACM Proc. of Symposium on Solid Modeling and Applications, June 2003, Seattle, Washington, USA, pp. 96-107.

[6] K. Suresh, Automating the CAD/CAE dimensional reduction process. ACM Proc. of Symposium on Solid Modeling and Applications, Seattle, Washington, USA, June 2003, pp. 76-85.

[7] A. Alexe, V. Gaildrat, L. Barthe, Interactive modelling from sketches using spherical implicit functions, ACM AFRIGRAPH'04, Stellenbosch, Cape Town, South Africa, November 2004, pp. 25-34.

[8] P. J. Besl, R. C. Jain, Three-dimensional object recognition, ACM Computing Surveys,

17 (1), 1985, pp. 75-145.

[9] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A Search Engine for 3D Models, ACM Trans. on Graphics, 22(1), 2003, pp. 83-105.

[10] S. Yoshizawa, A. G. Belyaev, H. P. Seidel, Free-form skeleton-driven mesh deformations, ACM Proc. of Symposium on Solid Modeling and Applications, June 2003, Seattle, Washington, USA, pp. 247-253.

[11] R. L. Blanding, C. Brooking, M. A. Ganter, D. W. Storti, A skeletal-based solid editor, ACM Proc. of Symposium on Solid Modeling and Applications, Ann Arbor, Michigan, United States, 1999, pp. 141-150.

[12] D. W. Storti, G M. Turkiyyah, M. A. Ganter, C. T. Lim, D. M. Stal, Skeleton-based modeling operations on solids. ACM Proc. of Symposium on Solid Modeling and Applications, 1997, Atlanta, Georgia, USA, pp. 141-154.

[13] G Borgefors, I. Nystrom, G S. Di Baja, Computing skeletons in three dimensions, Pattern Recognition, 32 (7), July 1999, pp. 1225-1236.

[14] K. Palagyi, A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, Pattern Recognition Letters, 19 (7), May 1998, pp. 613-627.

[15] C. Lohou, G. Bertrand, A 3D 12-subiteration thinning algorithm based on P-simple points, Discrete Applied Mathematics, 139 (1-3), April 2004, pp. 171-195.

[16] C M. Ma, M. Sonka, A fully parallel 3D thinning algorithm and its applications, Computer Vision And Image Understanding, 64 (3), November 1996, pp. 420-433.

[17] P. K. Saha, B. B. Chaudhuri, D. D. Majumder, A new shape preserving parallel thinning algorithm for 3D digital images, Pattern Recognition, 30 (12), December 1997, pp. 1939-1955.

[18] K. Palagyi, A. Kuba, A parallel 3D 12-subiteration thinning algorithm, Graphical Models and Image Processing, 61 (4), July 1999, pp. 199-221.

[19] W. J. Xie, R. P. Thompson, R. Perucchio, A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton, Pattern Recognition, 36 (7), July 2003, pp.

1529-1544.

[20] C. M Holt, A. Stewart, M. Clint, An improved parallel thinning algorithm, Communications of the ACM, 30 (2), February 1987, pp. 156-160.

[21] L. J. Latecki, U. Eckhardt, A. Rosenfeld, Well-composed sets, Computer Vision and Image Understanding, 61 (1), January 1995, pp. 70-83.

[22] L. J. Latecki, 3D well-composed pictures, Graphical Models and Image Processing 59 (3), May 1997, pp. 164-172.

[23] L. J. Latecki, C. M. Ma, An algorithm for a 3D simplicity test, Computer Vision And Image Understanding, 63 (2), March 1996, pp. 388-393.

[24] G. Bertrand, Z. Aktouf, A 3D thinning algorithm using subfields, SPIE Proc. of Conference on Vision Geometry, San Diego, CA, USA, 1994, pp. 113–124.

[25] P. K. Saha, B. B. Chaudhury, D. D. Majumder, A new shape-preserving parallel thinning algorithm for 3D digital images, Pattern Recognition, 30 (12), December 1997, pp. 1939–1955.

[26] G. Bertrand, A parallel thinning algorithm for medial surfaces, Pattern Recognition Letter, 16 (9), September 1995, pp. 979–986.

[27] W. X. Gong, G. Bertrand, A simple parallel 3D thinning algorithm, IEEE Proc. of International Conference on Pattern Recognition, Atlantic City, NJ, USA, 1990, pp. 188–190.

[28] T. Lee, R. L. Kashyap, C. Chu, Building skeleton models via 3-D medial surface/axis thinning algorithms, CVGIP: Graphical Models and Image Processing. 56 (6), November 1994, pp. 462–478.

[29] Y. F. Tsao, K. S. Fu, A parallel thinning algorithm for 3-D pictures, Computer Graphics Image Process, 17, 1981, pp. 315–331.

[30] C. M. Ma, On topology preservation in 3D thinning, CVGIP: Image Understanding, 59(3), 1994, pp. 328–339.

[31] A. Rosenfeld, A characterization of parallel thinning algorithms, Information and

Control, 29, 1975, pp. 286–291.

[32] R. W. Hall, Tests for connectivity preservation for parallel reduction operators, Topology and its Applications, 46, 1992, pp. 199–217.

[33] C. Ronse, Minimal test patterns for connectivity preservation in parallel thinning algorithm for binary images, Discrete Applied Mathematics, 21, 1988, pp. 67–79.

[34] R. W. Hall, Connectivity preserving parallel operators in 2D and 3D images, SPIE Proc. of Conference on Vision Geometry, Boston, MA, USA, 1992, pp. 172–183.

[35] H. Blum, A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, MA, USA, 1967, pp. 362–380.

[36] A. Chaturvedi, Z. Lee, Three-dimensional segmentation and skeletonization to build an airway tree data structure for small animals, 50 (7), April 2005, Physics in Medicine and Biology, pp. 1405-1419.

[37] M.S. Talukdar, O. Torsaeter, M.A. Ioannidis, J.J. Howard, Stochastic reconstruction, 3D characterization and network modeling of chalk, Journal of Petroleum Science And Engineering, 35 (1-2), July 2002, pp. 1-21.